2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

An Evolutionary Hybrid Algorithm for Control System Analysis

Sulistiyo, Nakao Zensho, Chen-Yen Wei
Electrical and Electronic Engineering, University of the Ryukyu
email: {tiyo, nakao}@augusta.eee.u-ryukyu.ac.jp; chen@tec.u-ryukyu.ac.jp

***Abstract*—We employ Genetic Programming (GP) which is optimized with Simulated Annealing (SA) to recognize characteristic of a plan. Its result is described in Laplace function. The algorithm proceeds with automatic PID designs for the plant.**

***Index Terms*—Genetic Programming, Simulated Annealing, PID, compensator.**

I. INTRODUCTION

OUR previous works [6]-[8] give detail explanation on how to identify characteristic of dynamical system by means of artificial intelligence (AI). This process is conducted in time domain. By applying certain rules, the resulting time domain function can be transformed to Laplace function. This identification algorithm remove manual plant analysis step which many times are very difficult to do. As results many control systems analyses can be automated. Ref. [8] gives examples on how to automate Lead-Lag compensators design.

It is interesting to note that more than half of the industrial controllers in use today utilize PID or modified PID control schemes ([4] pp 681). Many techniques to tune PID are already proposed. Earlier techniques are manual techniques. They start with searching for transfer function of the observed plan and continue with compensator design process. Such methods are called analytical methods. These methods usually precise, but nevertheless to find transfer function of many plants is not an easy task. Ziegler-Nichols is a method to design PID if transfer function of the system is unknown. But this method is not precise also less flexible to make desired PID with certain behaviors. Others methods use artificial intelligence (such as fuzzy or neural network) in designing PID [5]. These methods are quiet powerful, but they do not give us any information about the system such as transfer function of the observed system. One of its disadvantages is: when the compensated system is failed, debugging process will be very difficult.

Sulistiyo is a master student of University of the Ryukyus. 1 Senbaru Nishihara-cho 903-0213 Okinawa Japan

Nakao Zensho is a professor of Electronic and Electrical engineering of University of The Ryukyus.

Yen-Wei Chen is a professor of Electronic and Electrical Engineering of University of the Ryukyus

Our proposed algorithm combines between advantages in process automation by using artificial intelligences and precession in designing PID, which is achieved by analytical method. First, It searches for system transfer function automatically which is accomplished by GP and SA for system identification. Result of the first step is system's transfer function in Laplace domain. This equation is used in the second step where PID design takes place. This design process is also automated using SA. It is impossible to developed ultimate methods, which can solve all PID design problems. So our aim in this paper is to give guidance on how to automate PID design without loosing information of the system itself.

II. SYTEM IDENTIFICATION TECHNIQUE

Since it is impossible to describe our system identification by means of GP and SA in detail here, we encourage readers to read [1] and [6]-[8]. Here we only provide brief explanation

Dynamic system can be described by differential equation. If this equation can be linearized then Laplace transform can be used to solve the differential equation. Transformation from time domain function to Laplace domain function is almost always results in rational function, which can be divided into several elementary functions by partial fraction expansion method. Relation between elementary Laplace functions and their related time domain functions are listed in Important Laplace Transform (ILT) table. As a result solution of differential system many times is a combination of several functions of ILT table.

GP basically will search a combination of its terminal nodes and function nodes to solve given problem. Since system identification in this case can be regarded as finding combination of elementary functions as listed in ILT table, GP may be employed. To boost its performance, functions from ILT table can be included as terminal nodes. Numerical constants exist in each individual are optimized using SA. Operators to combine those elementary functions are only '+' and '-', so transformation to Laplace domain can be automated in easy way. In our previous work we called this algorithm GPWNSTNSA.

It is required that the system whose data are taken is a unity feedback system. When the data are taken, the feedback line should be removed and the input should be impulse. This data will be used as training data in GPWNSTNSA algorithm. So after transforming the result of GPWNSTNSA (time domain function), we get transfer function in Laplace ([4] pp. 416-423).

III. AUTOMATIC PID DESIGN

A. Single Component

A PID-Controller consists of three different elements, so it is sometimes called a three terms controller. PID stands for P (proportional), I (integral), and D (derivative) control, its basic implementation is shown at Fig.1. PID can be implemented to meet various design specifications for the system. These can include the rise and settling time as well as the overshoot and accuracy of the system step response. Further explanation about PID can be found at [4] and [9].

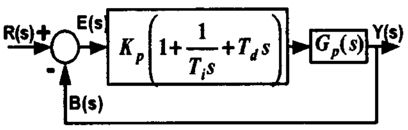

Fig. 1. PID control of a plant

The task is basically to determine values of K_p, T_i and T_d, so the over all system meet the necessary conditions. This is a problem of numerical variables optimization. So artificial intelligence technique for numeric optimization such as SA can be employed to search optimum combination of those coefficients. Fig.2 is an example on using SA to design PID,(*TS*: settling time, *OV*: over shoot)

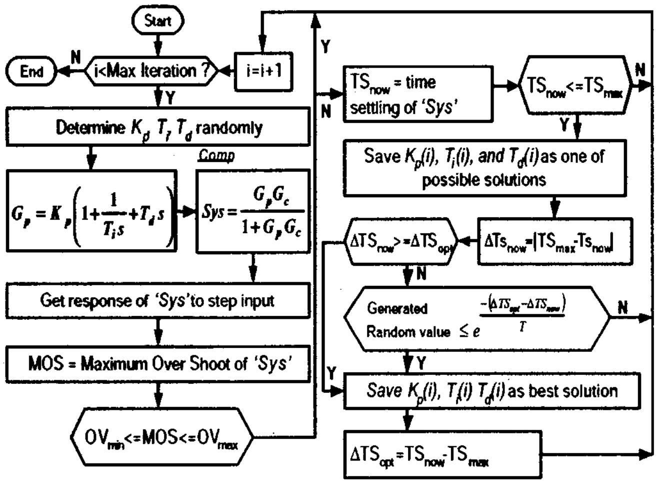

Fig.2. Algorithm for single PID design

In the basic PID control system such as the one shown in Fig. 1, if the reference input is a step function, then, because of the presence of the derivative term in the control action, the manipulated variable u(t) will involve an impulse function /delta function (set-kick phenomenon). To avoid the set-kick phenomenon, we may wish to operate the derivative action only in the feedback path so that the differentiation occurs only on the feedback signal and not on the reference signal. The control scheme arranged in this way is called the PI-D (Fig. 3)

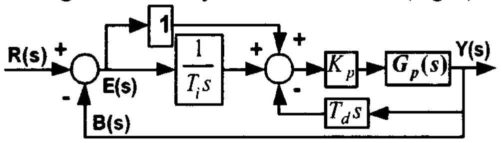

Fig.3. PI-D-controlled system

Even though the control scheme is different, still the task is to determine values of K_p, T_i, and T_d. Algorithm in Fig.2 can be used by modifying block <u>Comp</u>. Transfer function for PI-D:

$$Sys=\frac{Y(s)}{R(s)}=\frac{K_pG_p(s)}{1+\left(1+\frac{1}{T_is}+T_ds\right)K_pG_p(s)} \quad (1)$$

Another modification is what so called I-PD. The basic idea of the I-PD control is to avoid large control signal (which will cause saturation phenomenon) within the system. By bringing the proportional and derivative control actions to the feedback path, it is possible to choose larger values for K_p and T_d than those possible by the PID control case, the I-PD-controlled system will attenuate the effect of disturbance faster then PID-controlled case. The scheme of I-PD is shown Fig.4

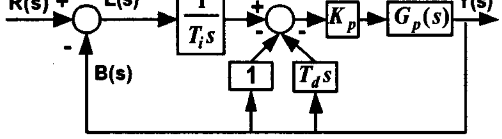

Fig.4. I-PD-controlled system

Algorithm as shown in Fig.2 can be used also to solve I-PD control scheme by modifying block <u>Comp</u>. For I-PD the equation should be:

$$Sys=\frac{Y(s)}{R(s)}=\left(\frac{1}{T_is}\right)\frac{K_pG_p(s)}{1+\left(1+\frac{1}{T_is}+T_ds\right)K_pG_p(s)} \quad (2)$$

Basically, algorithm as shown in Fig.2 is a common way to automate PID design in case only one compensator is involved. In all design: *Gp*, TS_{max}, *OV*min, OV_{max} are inputs to the algorithm. As usual in SA, *T* (temperature of annealing process) is decreasing as process continues.

B. Double Components

This project is made with MATHLAB as its programming environment. MATLAB has Symbolic Math Toolboxes, this toolbox incorporates symbolic computation into the numeric environment of MATLAB. Detail explanation of this toolbox can be found at [1].

Ability to process equation in symbolic manner give many benefits in making algorithm for many others PID scheme. Here we give an example on solving two-degrees-control-freedom, where the system is subjected to the disturbance input D (s) in addition to reference input R(s). The scheme is shown in Fig.5.

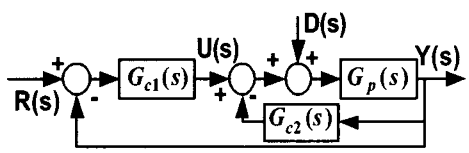

Fig.5. Two-degrees-of freedom control system

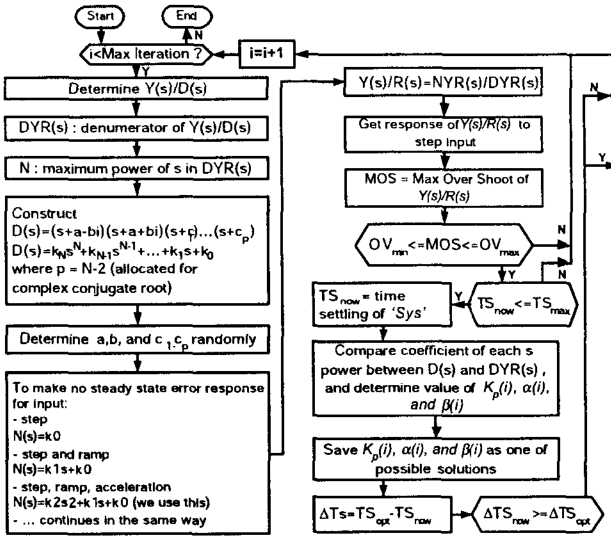

Fig.6 Algorithm for double PID design

Relation of input and output of the compensated system can be described by these bellow three equations:

$$\frac{Y(s)}{R(s)}=\frac{G_{c1}G_p}{1+G_cG_p} \quad (3)$$

$$\frac{Y(s)}{D(s)}=\frac{G_p}{1+G_cG_p} \quad (4)$$

$$G_c=G_{c1}+G_{c2} \quad (5)$$

Algorithm as shown in Fig.6, demonstrate how to design control systems that will exhibit no steady errors in following ramp and acceleration inputs and at the same time force the response to the step disturbance input to approach zero quickly. This technique is usually called zero-placement approach.

IV. EXPERIMENTATION

A. Single Component

The pitch angle of an airplane is controlled by adjusting the angle (and therefore the lift force) of the rear elevator. The aerodynamic forces (lift and drag) as well as the airplane's inertia are taken into account.

The equations governing the motion of an aircraft are a very complicated set of six non-linear coupled differential equations. However, under certain assumptions, they can be decoupled and linearized into the longitudinal and lateral equations. Pitch control is a longitudinal problem, and in this example, we will design an autopilot that controls the pitch of an aircraft. The basic coordinate axes and forces acting on an aircraft are shown in the Fig.7 ([2] for detail).

We do not explain in detail on how to derive equations, which govern aircraft behavior here. We used a model of a commercial Boeing aircraft proposed by [2] as our plant. The input is elevator deflection angle) and the output is the pith angle. Using MATLAB we simulate behavior of the aircraft. Fig.8 (dashed-dotted line) represents behavior of the plant when unity feedback line is removed and impulse is applied to the plant as input.

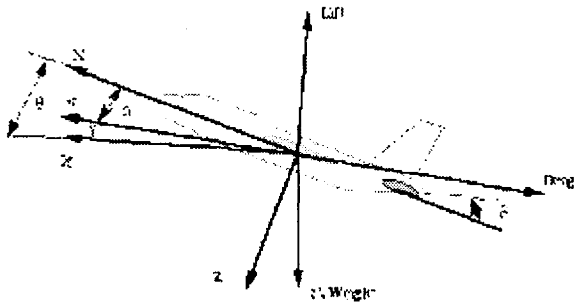

Fig.7. Basic axes and forces coordinate of an aircraft

The data is used as training data for GPWNSTNSA.

Table 1. General settings for GPWNSTNSA

Population size = 60
Maximum layers for initial generation = 3
Maximum layers = 5
Mutation Probability = 0.001
Maximum Generation = 60
Reproduction method = rank method
SA iteration for each individual = 30

Setting for GP's operators:

- Using 'ramped-half-and-half' to create initial population.
- Using rank selection for reproduction (best two individuals will be copied to the next generation).
- Using tournament selection to choose parents.

The experiments were conducted 5 times, individual with minimum error compare to the training data was used to design PID. Graphical representation of the best individual is shown in Fig.8 (solid line). As we can see from Fig.8, that system generated by GPWNSTNSA is almost the same with the target system, so the generated system is reliable enough to be used in PID design step.

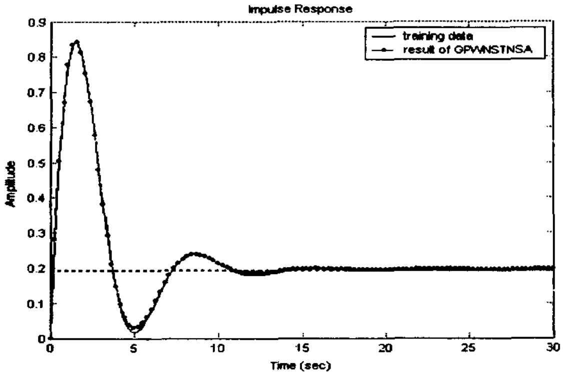

Fig.8. Comparison between model and result of GPWNSTNSA

The resulting equation after Laplace transformation is:

$$G_p = \frac{1.16s + 0.18}{s^3 + 0.76s^2 + 0.921s} \quad (6)$$

In reality it is difficult to realize certain design since there are limitations in equipments so our strategy here is instead of provide 1single solution, we provide some possible solutions and according to given criteria, the best solution is chosen. So in case the best solution can not be realized, then user still alternative solutions. Our goal in designing the for this aircraft problem was if the aircraft experienced a step with magnitude 0.2 rad (11 degree) the compensated should have: Overshoot (OV_{max}), less than 10%; Settling time(TS_{max}), less than 2 seconds Steady state error, less than 2% (data response is traced in backward direction, position of the point with magnitude out of SSE error is TS_{now}) Using algorithm in Fig.2, if variable range are: $6 <= \{T_d, K_p, T_i\} <= 2$. Some of possible answers are (the best marked with bold):

Table2. Result of single PID design
(solutions with the lowest TS is regarded as the best)

T_d	**5.5568**	4.2705	5.7700	5.5014	5.6281	5.0688
K_p	**5.4641**	5.9219	5.0806	5.3410	5.0343	5.1195
T_i	**3.0170**	5.1673	4.9496	3.3324	3.5229	3.9364
Ov	**0.2039**	0.2039	0.2048	0.2038	0.2039	0.2059
TS	**0.2200**	0.2200	0.2200	0.2300	0.2300	0.2300

As we can see the goals of creating PID are reached.

B. Double Components

Our goal in this experiment is to show that algorithm as shown in Fig.6 is working better than analytical methods, so we do not do any system identification we just take a specific transfer function and do automatic PID design. We will give comparison between our proposed method with analytical method on solving double PID design problem as mentioned in ([4] pp.705-718). The target plant is:

$$G_p(s) = \frac{10}{s(s+1)} \quad (7)$$

The task is to design controllers $G_{c1}(s)$ and $G_{c2}(s)$ such that maximum overshoot in the response to the unit-step reference input be less than 19% (OV_{max}), but more than 2%(OV_{min}), and the settling time be less than 1 sec (TS_{max}). It desired that the steady-state errors in following the ramp and acceleration reference input be zero. The response to the unit-step disturbance input should have small amplitude and settle to zero quickly. Variable ranges were a little wider then given by source book: $1.9 <= \{a, b\} <= 6$; $6 <= c <= 12$. Using Algorithm shown by Fig.6, the results are:

Table3. Result of double PID design for GC1

T_d	1.767	1.764	2.064	2.058	2.04	1.985
K_p	8.234	7.818	12.48	12.45	12.24	11.47
T_i	14.68	13.88	26.61	26.58	25.97	23.47
Ov	1.1773	1.1748	1.1890	1.1894	1.1896	1.1884
TS	0.6100	0.6300	0.8400	0.8400	0.8500	0.8700

Table4. Result of single PID design for GC 2

T_d	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1
K_p	0	~0	~0	~0	~0	0
T_i	0	~0	~0	~0	~0	0

The best result which given by analytical method (with 13671 attempts) are: $G_{c1}(s)$:T_d=2.04;K=12.244;T_i=25.968 and $G_{c2}(s)$: T_d=-0.1; K~0;T_i~0. It gives Ts:0.85. So our algorithm is proved to run faster and more accurate (compare column 4 and 5)

V. CONCLUSION

Many analytical algorithms for control systems already built, those algorithms usually quiet precise. But still they need human ability to derive characteristics of the related plants, which make those algorithms can not be automated. Our proposed algorithm provides solution on automating control system analyses, since it has automatic system identification, which is accomplished by GPWNSTNSA. Further benefits can be achieved by integrating artificial intelligence in solving control system designs.

REFERENCES

[1] www. Mathworks.com

[2] www. Engine.umich.edu/group/ctm/examples/pitch/PIDpitch.html.

[3] J. Koza: *Genetic Programming I: On the programming of computers by means of natural selection*, The MIT Press, 1992.

[4] Katsuhiko Ogata: *Modern Control Engineering*, Prentice Hall, 2002.

[5] Petrov M., T.Proychev, A.Topalov. An adaptive Hybrid Fuzzy – Neural Controller, Prepr 2nd IFAC Workshop on New Trends in Design of Control Systems,1997, pp 326-331.

[6] Sulistiyo, Zensho Nakao, Yen-Wei Chen, Accurate Symbolic Regression by Genetic Programming and Simulated Annealing Applied For Dynamical Systems Analysis, *Proc. of the Third Indonesian Scientific Meeting Kyushu-Okinawa*, Saga, 2003, pp. A1(014 - 019).

[7] Sulistiyo, Zensho Nakao, Yen-Wei Chen, & Matsuo Kazunori, A Data Fitting System for High Frequency Dynamical Systems by Genetic Programming, Simulated Annealing, and Adaptive Neuro Fuzzy Inference System, IEICE Technical Report on Nonlinear Problems, 2003, pp. 19-24.

[8] Sulistiyo, Zensho Nakao, Yen-Wei Chen, Kazunori Matsuo, Automatic System Identification By A Hybrid Algorithm, to be presented at *Proceedings of the 5th Metaheuristic International Conference*, Kyoto, 2003.