System Identification Using Genetic Programming for Electro-Hydraulic Actuator

Hamid Bamshad, Minsu Jegal, and Hyun Seok Yang
Mechanical Department, Yonsei University, Seoul, South Korea
Email: {hamidbamshad, jglupin,hsyang}@yonsei.ac.kr

***Abstract*—Hydraulic actuators have wide ranges of application in modern industry. Designing a robust controller for this kind of actuator is always a significant concern among engineers. Recently the new kind of compact electro-Hydraulic actuator developed which although it has many benefits. It associated with some problems as unknown highly nonlinear parameters. First step to design controller is a system identification. There are many algorithms and techniques to find an accurate model for the system. In this paper, the tree structure genetic programming algorithm has been used to find the nonlinear model for the electro-hydraulic actuator. For this kind of electro-hydraulic actuator, some behaviors of the system are known but still the relation of these sub-functions–whole system- is not known due to high nonlinearities and compact model. So, based on the abilities of genetic programming, this theory was selected for system identification. This system is considered as a multi-input, single-output system. There are some significant considerations for genetic operations such as selection, crossover, and mutation. The fitness function is defined based on the error between the model output and experimental data. After many runs and generations, the model for the system has been found which is well-matched with real data.**

***Index Terms*—genetic programming, system identification, hydraulic actuator, nonlinear system**

I. Introduction

One of the most popular actuators in industry is hydraulic actuator. This kind of actuator is able to provide high force with consuming less energy and also low cost in compare to the electric drive systems. Recently, a new kind of hydraulic actuator (Fig.1) has been developed which has some advantages over the common hydraulic actuators such as completely isolated system without any oil leakage, small size, self-contained, easy plug and play installation, light weight, and so on. These features make this actuator unique and there is wide range of application for it, especially in case of positioning systems and robotics.

Achieving the desirable control for the systems has been a significant concern among control engineers. A number of robust and adaptive control strategies have been proposed to deal with such problems [1] but the performance of these controllers, in the first step, is highly related to the reliability and accuracy of the system model which is obtained by system identification. Moreover, system identification for complex systemalways associates with some difficulties [2] Hydraulic actuator is considered as a system with high level of nonlinearities, uncertainties, and time varying characteristics which cause some difficulties for system identification.[1]

System model can be achieved by two approaches, physical law modeling and system identification. Physical law modeling which requires expert and thorough knowledge about the system is hard to apply. System identification in compare, needsonly input and output data to obtain system model, is popular due to ease to apply. [1] Modeling the system by physical laws gives us a particular insight into the system's properties, which allows us to seek the parameterized models that are flexible enough to capture most of dynamic behaviors of the system. However, because of complexity of the system, it is not always possible to achieve a perfect mathematical model which presents the real system precisely.

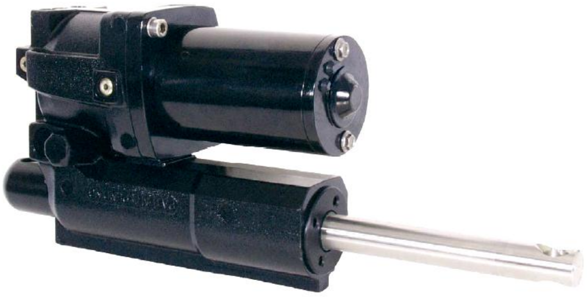


Figure 1. Electro-hydraulic actuator

One of the well-known solutions for this problem is Genetic Programming (GP). GP is an optimization method which can be used to optimize the complex systems with have nonlinear structure. [3] In 1954 GP began with the evolutionary algorithms first used by Nils Aall Barricelli applied to evolutionary simulations. Later, in early 1970s, GP was used to solve more complex engineering problem through ecelution strategies by Ingo Rechenberg. John Holland played significant role in development on GP in 1970s. [4] The first statement of modern "tree-based" genetic programming was given by Nichael L. Cramer in1985. [5] This work was later greatly expanded by John R. Koza, a main proponent of GP who has pioneered the application of genetic

Manuscript received September 8, 2014; revised December 19, 2014.

doi: 10.12720/joace.3.6.457-462

programming in various complex optimization and search problems. [6]

GP that is an extension of a genetic algorithms paradigm, presents some candidate solutions (individuals) as a tree structure (Fig. 1.) which are created by some random selection of both terminal and function sets from a database and function libraries that are available. The initial population will be selected randomly and they vary in length. [7] GP works by emulating natural evolution to generate a better model structure which minimize or maximize some objective functions. The fitness of solution candidates is obtained by evaluating the accuracy of the candidate solutions output over training set and the real experimental data. One of the significant advantages of GP is that if already some behavior or pre-knowledge of the system are known, there is possibility to use these knowledge in order to create better initial population and as a result getting better result from GP. [3]

In the following, the second section of this paper will give a brief introduction about genetic programming algorithms and the rest of this paper is organized as follows. Section 3 describes how to apply genetic programming into the system identification for electro-hydraulic actuator. Section 4 presents the theoretic modeling of an electrohydraulic control system. Finally, Section 5 presents the experimental tests aswell as the identification results and finally section 6 concludesthe paper.

II. Introduction to Genetic Programming

As mentioned in previous part, genetic programming is considered as an extension of genetic algorithms and in general definition, GP is an automated method for generating computer programs that solve carefully specific problems, but highly depended on principles of natural selection. In other words, GP makes some initial computer programs randomly, where only the relatively more successful individuals pass on genetic operations to the next generation. Fig. 2. shows a general GP procedure flowchart for the function approximation. Each generation can improve good candidate solutions and finally after many generations, the best result and fitted function can be obtained. The most issues about GP will be explained briefly as follow;

A. Encoding and Initial Population

All the individuals are encoded as a tree structure. Each tree structure presents a mathematical model as shown in Fig. 3. Each structure contains terminal nodes (such as basic algebra operations, sub functions) and root nodes (such as constant number, input). These terminals and roots could be selected from available function library. Despite the genetic algorithms chromosomes, length of GP's individuals is not fixed and they may change from one generation to the other generation. Moreover, the length of each tree structure can be varied by other tree structure. This feature of GP makes a good chance for it to have a wide range of candidate solutions. [8]

In order to create initial tree structure, nodes are chosen randomly from the terminal and root sets. Therefore, this technique can produce various type of tree with various lengths.

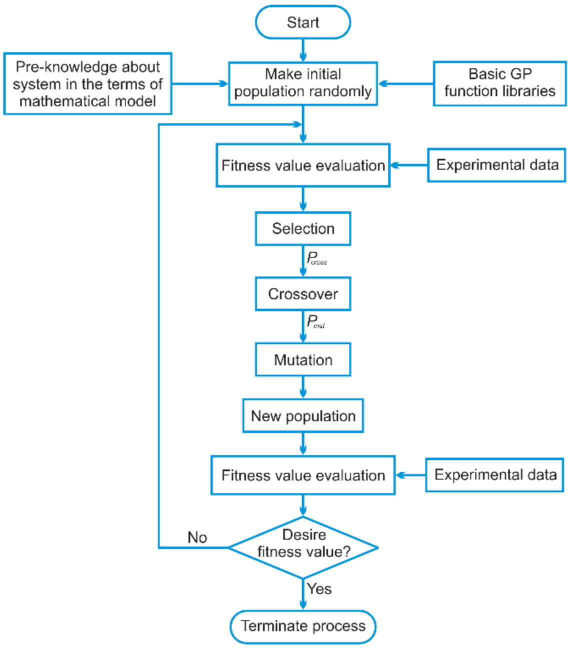


Figure 2. Example of a figure caption.

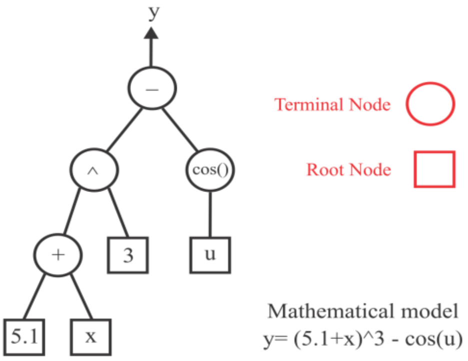


Figure 3. Tree structure example

B. Function Library

Function library is a kind of pool filled with function and terminal sets in order to make initial population or apply mutation branch. In definition, terminal set is composed of statements, operation, basic algebra and sub-function available to the GP system. Besides, root set is comprised of the inputs to the GP program and the constants number. Moreover, any prior knowledge of the physical system should be considered for making initial population.

C. Fitness Function

A fitness function is a particular type of objective function that is used to describe how close a solution candidate is to the real value. There are many fitness functions for GA and GP which based on the conditions, each problem requires a proper fitness function. Generally speaking, fitness function for system identification is the error between the output of solution candidate and experimental data based on the same input.

D. *Operations*

In GP there are three major operations which creates the main parts of GP; Selection, Crossover and Mutation. Here is a brief explanation for these operations.

1) *Selection*

The basic part of the selection process is to select condidates stochastically from one generation to create the basis of the next generation. The requirement is that the fittest individuals have a greater chance of survival than weaker ones. There are some well-known selection methods for GA and GP such as Roulette-wheel, Tournament, Ranking, and Elitist selection methods.

2) *Crossover*

Crossover is a genetic operator that combines two individuals (parents) in order to make new individuals (offspring).Crossover occurs during evolution according to a user-definable crossover probability. The crossover probability ratio depends on the problem conditions but, commonly, it is defined between 60% to 90%. Crossover algorithm has been shown by Fig. 4.

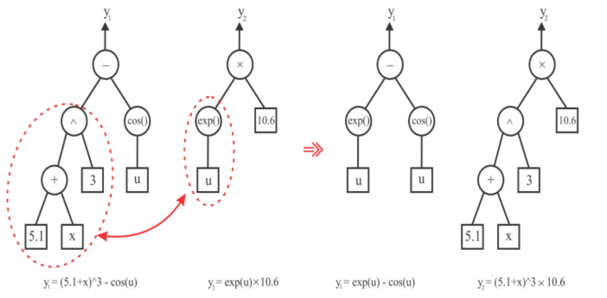


Figure 4. Crossover operation

3) *Mutation*

Mutation is a genetic operator that changes one sub-treein an individualwith a randomly generated sub-tree (Fig. 5). This can result in entirely new individual has been added to the population. Mutation is an important part of the genetic search since it helps to prevent the population from stagnating at any local optima. The mutation probability should usually be set fairly low, less than 5%.

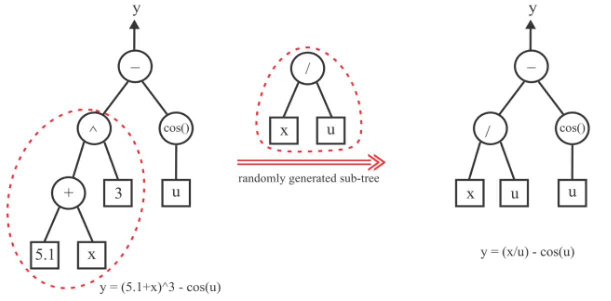


Figure 5. Mutation operation

E. *Termination*

Termination is the criterion by which the genetic algorithm decides whether to continue searching or stop it. There are many ways to define criterion for termination based on problem conditions. The most well-known criterions are generation number, fitness threshold and fitness convergence. Generation number method stops the evolution when the user-specified max number of evolutions has been run.As fitness threshold, generation will be stopped whenever the best fitness in the current population becomes less than the user-specified fitness threshold when the objective is to minimize the fitness. As fitness convergence, generation will be stopped whenever the fitness is deemed as converged to some number and there is no anymore improvement in the fitness value.

III. Apply GP into the Hydraulic Actuator Problem

As mentioned before in introduction section, the main issue in this research is to find a perfect model for electro-hydraulic actuator in order to design a good controller. This new actuator is totally compacted and isolated that does not provide any measurements for the inside parameters such as pressure and oil flow rate which are two important parameters to find the mathematical model for the system. GP has been used to solve this problem. In this research Matlab has been used to encode GP in computer. The main issues about proposed GP are explained in the following parts.

A. *Encode Tree Structure*

One of the important parts for GP is how to encode the algorithm into the computer language. As mentioned above, Matlab has been used for coding which provide many tools for coding. This hydraulic actuator is considered as multi input and single output (MI SI). Based on the available functions and conditions, assumed that each node is able to have maximum 2 branches and minimum no branch. A fully branched template defined in the form of matrix as Fig. 6.(a) shows. These nodes will be filled by roots or terminals; if node is filled by terminal, it means that there is no branch for that node and the following sub-tree template for that node will be empty. In order to apply this structure a specific number is assigned for each root or terminal function. These numbers which present a root or terminal are recorded in the template matrix 1 that is called as a Function Matrix shown in Fig. 6(b). There is template matrix 2 that is called as a Features Matrix shown in Fig. 6(c). Each node of this matrix contains the number of branch for the same node with the node in the function matrix. Therefore, the value for arrays in feature matrix could be zero, one, or two. Fig. 6(d) shows a Value Matrix which is going to filled by the value of constant number, input variable (based on experimental data), and calculated value for each terminal node consequently. Hence, the first node in value matrix present the final output for this individual based on the ith experimental data.

B. *Initial Population*

Initial population plays a significant role in finding perfect model. Initial population should contains not only all of the basic algebra functions, but also all the suspected behavior of the system. In order to make initial population, there are several assumptions those should be considered. These assumptions can be listed as follow;

- Population size should be enough to have various type of tree structure. In this research 50 initial populations have been used.
- The maximum length for the tree structure in initial population is 10 nodes level
- The minimum length for the tree structure is 3 nodes level. It means that always the first and second node levels are filled with terminal functions which have 2 sub-branches.
- Each individual should contain all of the input variables inside its tree structure at least one time.

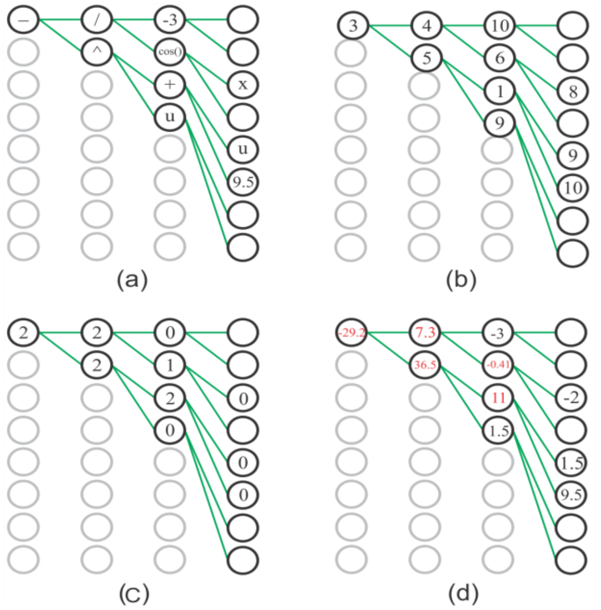


Figure 6. Tree structure templates for encoding into the computer. (a): General form of template, (b): Function matrix, (c): Feature matrix, (d): Value matrix

C. Fitness Function and Function Library

The function library is very important and should be flexible enough to be able to present a wide range of the real system. In this research fitness function is defined as a summation of squared errors between model output and experimental data for each of N data points as Eq. 1 shows. In order to achieve a perfect model, all the suspected function has been included in function library.

D. Operations

In this research the, two method of elitist and Roulette-wheel selection has been used. First, based on elitist selection theory, 20% of the population with best fitness values is chosen to pass to the next generation, avoiding the crossover and mutation operators. Elitism prevents the random destruction by crossover or mutation operators of individuals with good genetics. Then crossover and mutation applied to the rest of population. The rate of crossover and mutation for rest of population assumed to be 90% and 2%, respectively. Roulette-wheel selection method based on the fitness value of individuals selects two parents for crossover or one parent for mutation operators. The nodes for crossover and mutation selected randomly and it can make tree structure longer or shorter in length. In order to avoid long tree structure, a limitation of 20 nodes level defined for individuals. If operation makes offspring with longer length than limitation, that operation is not acceptable and the selected node should be change again until the new tree structure length been less than limitation.

IV. Experimental Test

In order to gain some experimental data for hydraulic actuator, an arm manipulator has been designed and manufactured. As Fig. 7 shows, the arm manipulator has one degree of freedom and it is controlled by hydraulic actuator. A linear potentiometer sensor has been implemented to measure and record arm rotation.

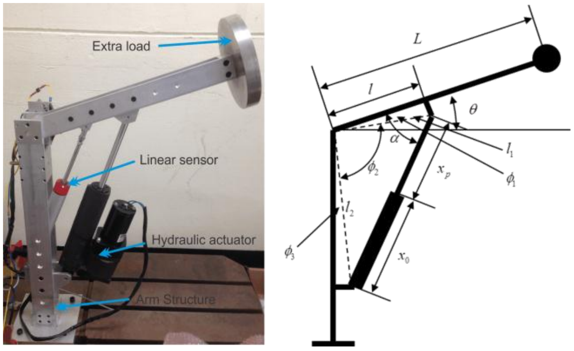
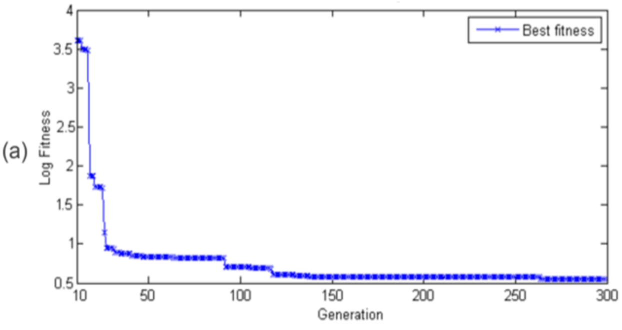


Figure 7. 1 DOF arm manipulator actuated by Electro-Hydraulic actuator

V. Result

A set of the experimental data that obtained from the system is provided for the GP training parts. Also, In order to validate the final GP solutions, two other sets of data are provided which are vary not only from training set but also from each other. In this system, three variables of command input, current position, and current speed have been used as the inputs and future position as an output for the GP algorithm. Also, Hyperbolic function has been added to the function library since it has behavior close to the friction behavior. The GP runs for several times and the best two runs are selected as the best answers. Each runs contains 300 generations. Fig. 8 shows the improvement of fitness function through the best two runs.



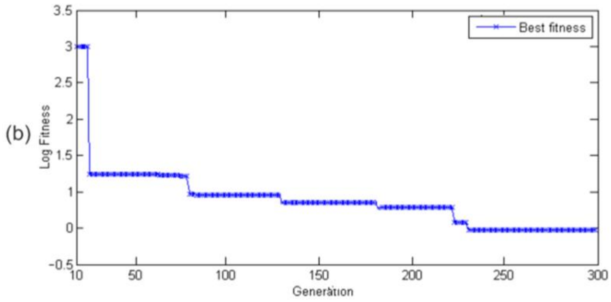


Figure 8. Fitness improvement during generations; (a),(b) best runs

Table I shows two final equations which found by these two best runs as a model for the hydraulic actuator. After finding best solutions, these solutions are validated by the validation data sets. In these models x_1, x_2, x_3, and y refer to the current position, current speed, current command input and future position of arm manipulator respectively.

TABLE I. TWO BEST SOLUTION FOUND BY GP

Best Solution	GP Model
Solution 1	$y = 0.007078\times x_2 - 0.0012\times x_1 + 0.000404\times tanh(x_3) -$ $0.000202\times tanh(x_2)\times(x_3+5.072)+1.001\times x_1\times tanh(x_1+10.88) -$ $0.007078\times tanh(2\times tanh(x_3))\times tanh(tanh(x_2{}^2)) - 1.0453\times 10^{(-5)}$
Solution 2	$y = 0.9998\times x_1 + 0.001449\times x_2 + 0.0001587\times x_3$- $0.0007536\times \tanh(\tanh(x_2\times(\cos(x_3)\text{-}10.26)))$- $0.001629\times\cos(x_1)+0.0001587\times\tanh(x_2)+0.00163$

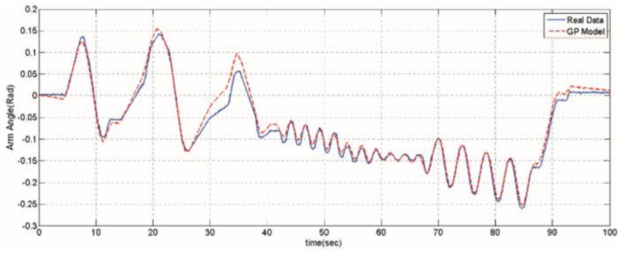
Fig. 9 and Fig. 10, show the result of how obtained solutions are fitted with real data for both training and validation sample data sets. As the table1 shows, both equations contain hyperbolic function which states that the friction plays important role in this system. In order to find the difference between the real data and GP model, the difference between output values of each of them for the three sample data sets is calculated by root mean square error (RMS) as following equation.

$$RMS \quad Error = \sqrt{\sum_{i=1}^{N} e_i^{\,2} \Big/ N} \tag{3}$$

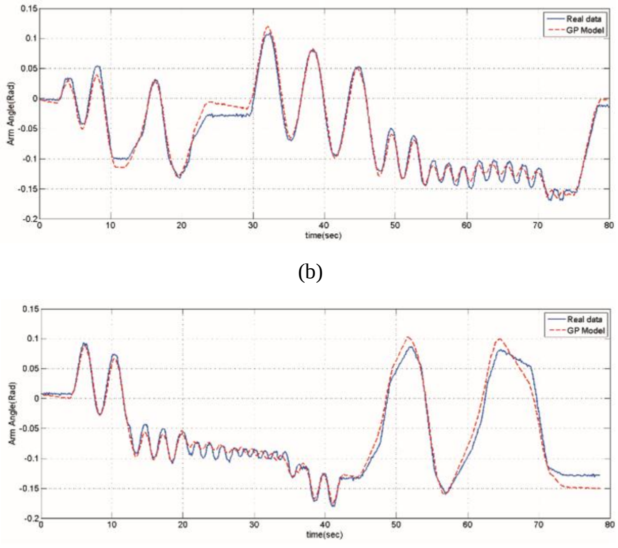
where

$$e = y_{real} - y_{\text{mod}el} \tag{4}$$

Term N indicates number of data sample in each set.



(a)



(b)

(c)

Figure 9. System output for the first solution for (a) training set , (b) First validation set, and (c)Second validation set

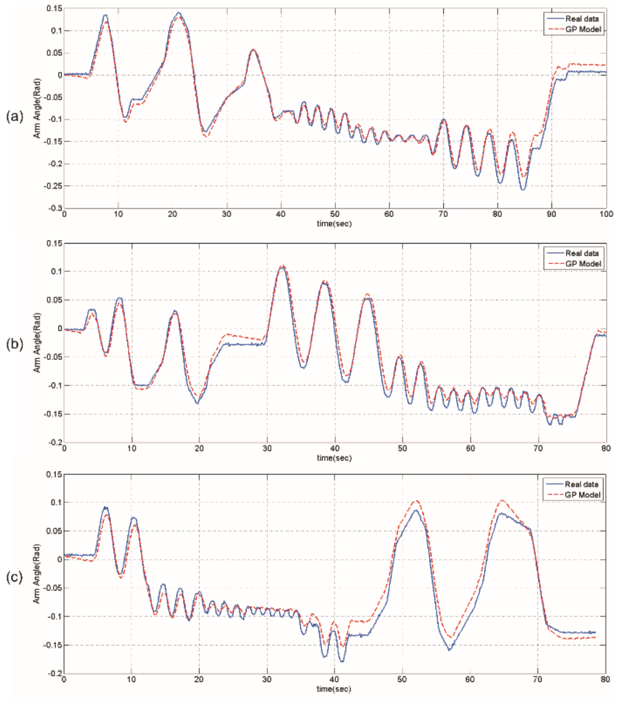
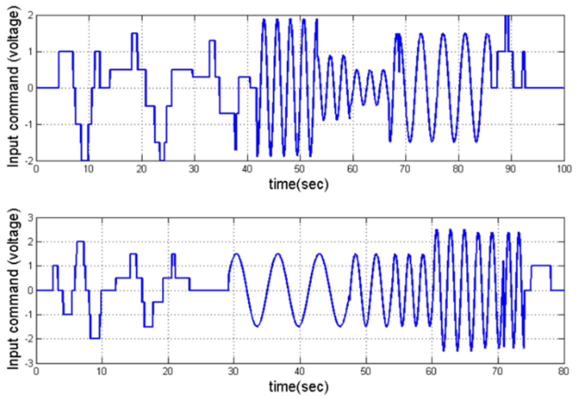


Figure 10. System output for the second solution for (a) training set , (b) First validation set, and (c)Second validation set



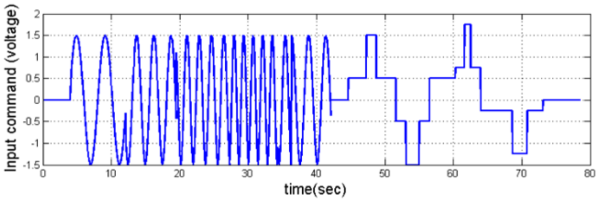


Figure 11. Random input command to the system for (a) Training set, (b) First validation set, and (c)Second validation set

Fig. 11 shows the three different command inputs which have been used for all training and validation sets in both real system and GP model as an open loop system without out any feedback. These inputs have both step and sinusoid with various frequencies and amplitudes in order to excite all moods of the system.

VI. Conclusion

The results of this investigation confirm that Genetic Programming is indeed a powerful tool in order to find the nonlinear model for the Electro-Hydraulic actuator. It can allow nonlinear model structure to be developed in the way that best fit experimental data satisfied. The GP automates the trial and error process of combination of suspected sub-functions and as a result it can be used as a function approximation for other modeling cases. Moreover, there some certain concerns which should be considered while using GP algorithm. How to choose function library, how to create initial population and how to find the fitness value are some important issue in GP algorithm. Also, the experimental data plays important role in GP since they should be wide enough to cover all the behaviors of the system and also it's should be taken in to the account that validation set data must be vary from the training set.

Acknowledgment

This research was supported by the Nuclear R & D Program through the Ministry of Science,ICT & Future Planning.

References

[1] T. G. Ling, M. F. Rahmat, and A. R. Husain, "System identification and control of an electro-hydraulic actuator system," in *Proc. 2012 IEEE 8th International Colloquium on Signal Processing and its Applications*.

[2] M. F. Rahmat, S. M. Rozali, N. A. Wahab, Zulfatman, and K. Jusoff, "Modeling and controller design of an electro-hydraulic actuator system," *American Journal of Applied Sciences*, vol. 7, no. 8, pp. 1100-1108, 2010.

[3] G. J. Gray, D. J. Murray-Smith, Y. Li, K. C. Sharman, and T. Weinbrenner, "Nonlinear model structure identiÞcation using genetic programming," *Control Engineering Practice*, vol. 6, 1341Ð1352, August 1997.

[4] H. J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[5] N. L. Cramer, "A representation for the adaptive generation of simple sequential programs," in *Proc. 1st International Conference on Genetic Algorithms*, 1985.

[6] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA, USA: MIT Press, 1992.

[7] Y. S. Yeun, J. C. Suh, and Y. S. Yang, "Function approximations by superimposing genetic programming trees: With applications to engineering problems," *Information Sciences*, vol. 122, 259±280, March 1999.

[8] S. Settea and L. Boullartb, "Genetic programming: Principles and applications," *Engineering Applications of Artificial Intelligence*, vol. 14, pp. 727–736, 2002.

[9] T. X. Indrawanto and B. K. Jusoff, "Sliding mode control of a single rigid hydraulically actuated manipulator," *International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS*, vol. 11, no. 05, September 2011.

[10] L. Marton, S. Foder, and N. Sepehri, "A practical method for friction identification in hydraulic actuators," *Mechatronics Journal*, vol. 21, no. 1, pp. 350–356, September 2010.

[11] X. B. Trana, N. Hafizahb, and H. Yanadac, "Modeling of dynamic friction behaviors of hydraulic cylinders," *Mechatronics Journal*, vol. 22, pp. 65–75, December 2011.

Hamid Bamshad Born in 1987. He Received his B.Sc. degree in mechanical engineering at Ferdowsi Universiy of Mashhad, Iran, in 2011. He is currently master candidate in the department of mechanical Engineeringat Yonsei University, Seoul, South Korea.His main research interest is robotics and control and industrial automation.

Minsu Jegal received the B.S. from the Department of Mechanical Engineering, Yonsei University, Seoul, in 2009. He is currently a Ph.D. candidate in the Department of Mechanical Engineering, Graduate School of Yonsei University and coordinates research on the posture decision control of the redundant manipulator on mobile platform.

Hyun Seok Yang received the B.S. degree from the Department of Mechanical Engineering,Yonsei University, Seoul, in 1984 and the M.S. and Ph.D. degrees from the Department of Mechanical Engineering, Massachusetts Institute of Technology (MIT) in 1988 and 1993. He is currently a professor in the Department of Mechanical Engineering, Yonsei University and coordinates research on the motion control androbotics.