Direct Adaptive Control: An Echo State Network and Genetic Algorithm Approach

Dongming Xu, Jing Lan and José C. Principe
Computational NeuroEngineering Laboratory
Department of ECE, University of Florida
Gainesville, FL 32611
E-mail: {dmxu, jinglan, principe}@cnel.ufl.edu

Abstract—This paper presents a direct adaptive approach to design controllers for nonlinear dynamical systems, where system identification of the unknown dynamical system is not required. The solution is powered by both echo state network (ESN) and genetic algorithm (GA). ESN enables a simple modeling of the controller, with which only a linear readout needs to be trained. GA is used to optimize ESN's linear readout directly so that system identification is not required. Simulation results reveal that the algorithm is capable of achieving very good control performance with computational efficiency.

I. INTRODUCTION

Identification and control of unknown nonlinear systems are of great interest because systems in real-world applications are mostly unknown and nonlinear. Control system design usually falls into two general categories: indirect control and direct control. Most methods focus on indirect control because system architectures have been fully studied and can be analytically solved. Conventional indirect adaptive control includes two steps. In the first step, adaptive linear or nonlinear networks can be used to model a target plant that has unknown parameters. Great effort has been made on the design of an optimal network (reference model) that works the best to mimic the dynamics of the unknown plant. Examples such as local linear model, neural network, recurrent network, and fuzzy logics have been studied before [1], [2], [3], [4]. After system ID is completed, the resulting reference model is used to facilitate controller design. If the reference model has an inverse function, we can simply use it to generate the desired control signal based on the current system output and error. Or, we can use another adaptive model to construct the controller. The reference model then serves as a pathway that allows error back-propagation, which is often used in the design of feed-forward or recurrent neural networks. Structure of conventional indirect control system is shown in Fig. 1.

Direct adaptive control does not require the procedure of system identification. Thus, it tremendously reduces the computational complexity. However, it is oftentimes complicated to implement. In direct control, the unknown plant is between the controller and the system output. Hence, the output error can not be back propagated to adjust the control signal. One solution is to use a global optimization algorithm such as the genetic algorithm (GA) to search the whole parameter space of the controller [5]. Fitness function is used to evaluate

the potential of the parameters. Adjustment of controller parameters does not depend on the output error explicitly.

Fig. 1. Diagram of a conventional indirect control design system. For direct control, the block of system identification is not required.

Among the possible models to design controllers in both direct and indirect control, neural networks and recurrent networks have been extensively studied [1]. Although a recurrent network has advantages over feed-forward neural network in terms of modeling dynamical systems with its dynamical memory and time embedding, it is hard to implement one because its training demands high computational complexity. Recently, the echo state network (ESN) [6], [7] has been introduced as a novel approach for designing recurrent neural networks (RNN). ESN comprises two components: a fixed recurrent network carrying information about input history, and an adaptive feed-forward read-out that retrieves the desired information related to the input. ESN compensates for not training the recurrent connection weights by increasing the dimensionality of the hidden layer. When training ESNs, we only need to optimize the weights of a feed-forward linear network. ESN has been successfully applied for system identification [6], [8]. A natural approach would be to also use ESN as the controller. It has the power of recurrent network but requires simpler training. A problem arises on how to train the readout in a control system. Normally, it would be trained through recursive algorithms which minimize mean square error (MSE) or higher order error momentum, or simply by Wiener-Hopf solution. However, in such cases, we still

Fig. 2. Direct adaptive control using an echo state network. Genetic algorithm is used to optimize the linear mapper as the ESN readout.

need the system ID and a reference model. The simplicity of its training would seem to be insignificant to reduce the complexity of a controller design. In this paper, we investigate the feasibility of combining ESN and genetic algorithm under the framework of direct adaptive control to achieve both robust performance and computational efficiency.

II. CONTROLLER DESIGN USING ECHO STATE NETWORK

The system structure of direct adaptive control using an ESN is shown in Fig. 2. The ESN is used to construct function $u_k = U(e_k, d_{k+1})$ so that $y_{k+1} = g(y_k, u_k) = d_{k+1}$, where g describes the unknown plant. The linear readout of ESN is optimized by GA. We will now discuss each part in detail.

A. Echo State Network as the Controller

ESN's advantage lies in the ability to overcome the difficulties in RNN training by fixing a priori the recurrent network connections and adapting only a feed-forward read-out network [6], [7]. This approach perceives an RNN as a combination of two components: a reservoir of highly interconnected dynamical components, and a memoryless readout network that reads the reservoir and produces the network output. At each time step, the echo states carry information in their dynamics a short-term history of the input pattern, allowing the readout to instantaneously extract the desired information related to the history of input patterns. An interesting property of ESN is that the reservoir is obtained from a fixed network of recurrent connections. Therefore the training of the RNN reduces to that of a feed-forward network since only the readout network weights are trained. Internal state $\mathbf{x}$ of the simplest ESN has the following transfer function

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W}^{in}\mathbf{I}(n+1) + \mathbf{W}\mathbf{x}(n)),$$

where $\mathbf{f} = (f_1, f_2, \dots, f_N)$ is the internal unit's output function, which normally is a sigmoid function. $\mathbf{I}$ is the input vector and $\mathbf{W}^{in}$ is the input matrix that is randomly generated. The output is computed according to

$$\mathbf{O}(n+1) = \mathbf{f}^{out}(\mathbf{W}^{out}\mathbf{x}(n+1)).$$

When $\mathbf{f}^{out}$ is a linear function, we simply have $\mathbf{y}(n+1) = \mathbf{W}^{out}\mathbf{x}(n+1)$. $\mathbf{W}^{out}$ defines the memoryless linear network that needs to be trained. For our purpose, $\mathbf{I}(n) = [d(n+1) - y(n)]^T$ and $\mathbf{O}(n) = u(n)$, where d is the desired output state, y is measured output and u is the control signal.

Several parameters determine the performance of an ESN. The number of units is important because we need a large variety of dynamics reservoir which can only be achieved when the number of processing elements is large. Spectral radius (SR) of $\mathbf{W}^{out}$ must be less than 1 to satisfy the echo state conditions [6], [7]. It has also been shown that different values of (SR) determine the actual dynamics and memory length of ESN [9]. ESN with an SR close to 1 usually has slower memory decay and better performance when dealing with time-varying signals. In contrast to the problem of system identification, in our case, the control system tends to emphasize on most recent histories of system states and desired outputs. Thus a relatively small SR is sufficient.

Typically $\mathbf{W}^{out}$ can be learned through adaptive algorithms or simply using Wiener-Hopf solution, where $\mathbf{W}^{out} = \mathbf{R}^{-1}\mathbf{P}$, $\mathbf{R} = E\{\mathbf{x}\mathbf{x}^T\}$ and $\mathbf{P} = E\{d\mathbf{x}\}$ [6], [7], [8]. However, in the case of direct control, there is no solvable connection between the output error and the control signal. Thus, error information cannot be incorporated directly with adaptive algorithms. In this case, a global search algorithm seems appealing because it does not need error information to calculate the required update for parameters, but simply searches for new ones.

B. Optimization of ESN Readout by Genetic Algorithm

The genetic algorithm is one of the stochastic optimization methods that are suitable for difficult search problems [10]. The algorithm encodes a population of possible solutions into vectors or strings (*chromosomes*). An objective function (fitness function) is designed to evaluate the performance of every candidate solution. At each step, the fitness of every chromosome is evaluated. Chromosomes are ranked according to their relative fitness in the population. Then each of them is assigned a probability of reproduction based on the ranking. Probability of reproduction determines the chance whether the strings of a chromosome will be copied into the next generation or not. Winning chromosomes generate offsprings through some criteria such as crossover and mutation. Crossover exchanges portions of two parents from the previous generation and produce two new offsprings. Mutation is simply to change part of the chromosomes by a small chance to produce new ones in the new generation. There are various methods to design crossover and mutation schemes.

Fitness function design is straightforward for our work. Output error of the unknown plant needs to be minimized. Mean square error is an obvious choice for the objective function. At every time step t , we calculate $1/N \cdot \sum (d_t - y_t)^2$ throughout the whole training sequence. This criterion will guarantee the convergence of output error. However, we may also need to constrain the control signal. Intuitively, if the outputs of the system are smooth, we want the control signal to be smooth as well. One scenario that may occur often is that the control signal may oscillate but still produce the desired output. To solve this problem, we add the smoothness of the control signal into the fitness function. A simple method is to calculate the average changing rate (which can be approximated by the first-order derivative of the control sequence). Thus, we define the

fitness function for GA to optimize $\mathbf{W}^{out}$ as

$$F(\mathbf{W}^{out}) = \frac{1}{N} \left[\sum_{t=1}^N (d_t - y_t)^2 + \sum_{t=2}^N (u_t - u_{t-1})^2 \right]$$

III. SIMULATIONS AND DISCUSSIONS

We study a simplified SISO nonlinear missile dynamical model where only yaw dynamics are considered. The two-state system [11] is described as

$$\begin{aligned} \dot{x}_1 &= x_2 - 0.1 \cos(x_1)(5x_1 - 4x_1^3 + x_1^5) \\ &\quad - 0.5 \cos(x_1)u \\ \dot{x}_2 &= -65x_1 + 50x_1^3 - 15x_1^5 - x_2 - 100u \\ y &= x_1 \end{aligned}$$

where x_1 is the state to be controlled. The input is the rudder deflection and is limited by ± 0.5 radians in practice. Samples are collected with $T_s=0.05s$ (which corresponds to 500s of flight time). We use an ESN of 100 units where the internal connection weights are set to 0.4, -0.4 and 0 with probabilities of 0.02, 0.02 and 0.96, respectively. As discussed previously, spectral radius of $\mathbf{W}^{out}$ is set to a relatively small value 0.6. In the genetic algorithm, 30 populations of weight matrix $\mathbf{W}^{out}$ are produced for each generation. Crossover rate is set to 0.7 which means 70% of the new offsprings are generated through crossover. The other 30% will change the encoded string through mutation. A zero-mean Gaussian distributed random vector with variance of 0.05 will be added to each chromosome. The variance will linearly decay to zero when the required number of generation is reached. For every generation, 3 best candidates are preserved for the new generation. Input-output pairs for training are generated by using an arbitrary piece-wise linear trajectory as the control signal, and passing it through the missile system. The resulting states are the desired outputs. The first 700 samples are used for training and another 700 samples are used for testing.

GA is set to update for up to 300 generations but normally approaches the best solutions within 50 generations. Figure 3-4 show the training and testing performance of the ESN controller. The signal to error ratio (SER) for training is around 22dB and 18dB for testing with the same kind of signal. Control signals in training and testing both mimic a piecewise linear function although the desired control signal is unknown to the system in both cases. We also tested set-point (0 0.6) tracking performance of the controller. As can be seen in Fig. 5, rising time, overshoot and settling time are rather small. However, there is a residual error for the on-state. To examine whether this is a training problem or a crucial problem caused by the architecture, set-point output is used as the desired signal to train the controller. Figure 6 is the training performance when the step signal is used as desired output. The controller works very well for the set-point tracking.

Actually, One of the advantages of direct control is that the desired output for training can be customized to any signal that a real system may want to achieve. Thus, ESN readout

Fig. 3. Training results for ESN controller. 700 samples are used for training. Input-output pairs for training are generated from an arbitrary piecewise linear control signal.

Fig. 4. Testing results for ESN controller. 700 samples are used for testing. Desired outputs are generated in the same way as the training data.

can always be trained for specific tasks for more accuracy and at the same time still learn the general dynamics of the system. For example, we can also combine both the set-point tracking response and some slow changing signals together as the training sequence, thus to make the system learn to control both slow and fast dynamics. To compare the ESN controller with other methods, we also used a multiple PID constructed by the Gaussian mixture model [4] to check the set-point tracking performance (Fig. 7). The PID controller has a rising time of 0.33s, a settling time of 0.39s, and an overshoot of 4.5%. The ESN controller has a rising time of 0.21s, a settling time of 0.45s, and an overshoot of 6%. Compared to the PID controller, ESN controller has a shorter rising time, a slightly longer but similar settling time and a higher overshoot.

Fig. 5. Set-point (0-0.6) tracking performance. Input-output pairs for training are generated from an arbitrary piecewise linear control signal.

Fig. 6. Performance when set-point (0-0.6) is used for training.

The simulation results indicate the potential of an ESN controller when its readout is optimized by the genetic algorithm. Taking into account the rather small ESN (100 units) and the moderate running time of GA we used, performance could be improved by increasing the network complexity and the optimization time. Although the internal weights only need to satisfy the echo state conditions, GA may also be used to optimize their values. However, in such cases, the size of the parameter space is proportional to N^2 , which will significantly slow down the optimization process.

IV. CONCLUSIONS

We demonstrated a direct adaptive controller design by using echo state network and genetic algorithm. ESN has the power of recurrent network but lower training complexity. Combined with genetic algorithm, ESN performs well to control unknown nonlinear systems without system identification.

Fig. 7. Comparison of set-point tracking performance with PID controller

GA is efficient and not really time consuming with today's fast computer resources. Although we only used SISO system as an example, the method applies to MIMO systems. Moreover, the increase of output dimension does not change any of the design structure. Only when multiple control signals are needed, the searching time of GA may be increased due to the increase of the dimensions of ESN's readout.

ACKNOWLEDGMENT

This work is supported by NASA grant NAG-1-02068.

REFERENCES

- [1] K. S. Narendra, and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Journal on Neural Networks*, vol. 1, no. 1, pp. 698–706, March 1990.
- [2] K. S. Narendra, C. Xiang, "Adaptive Control of Discrete-Time Systems Using Multiple Models," *IEEE Journal on Automatic Control*, vol. 45, no. 9, pp. 1669–1686, 2000.
- [3] J.-S.R. Jang, C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378–406, 1995.
- [4] J. Lan, J. Cho, D. Erdogmus, J.C. Principe, M.A. Motter, J. Xu Local Linear PID Controllers for Nonlinear Control, *International Journal of Control and Intelligent Systems*.
- [5] Y. Ichikawa, T. Sawa, "Neural Network Application for Direct Feedback Controllers," *IEEE Transactions on Neural Networks*, vol. 3, no. 2, pp. 224–231, March 1992.
- [6] H. Jaeger The "echo state" approach to analysing and training recurrent neural networks. Technical report GMD report 148, German National Research Center for Information Technology, 2001. <http://www.faculty.iu-bremen.de/hjaeger/pubs.html>
- [7] H. Jaeger Short term memory in echo state networks. Technical report GMD report 152, German National Research Center for Information Technology, 2001. <http://www.faculty.iu-bremen.de/hjaeger/pubs.html>
- [8] Y.N. Rao, S.P. Kim, J.C. Sanchez, D. Erdogmus, J.C. Principe, J.M. Carmena, M.A. Lebedev, M.A. Nicolelis, Learning Mapping in Brain Machine Interfaces with Echo State Networks. ICASSP, 2005. Accepted.
- [9] M.C. Ozturk, D. Xu, J.C. Principe, Analysis and Design of Echo State Networks for Function Approximation Neural Computation, 2005. Submitted.
- [10] D.B. Fogel, "An introduction to simulated evolutionary optimization," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 3–14, January 1994.
- [11] X. Ni, M. Verhaegen, A.J. Krijgsman, H.B. Verbruggen, New Method for Identification and Control of Nonlinear Dynamic Systems, *Engineering Applications of Artificial Intelligence*, 9(3), 231–243, 1996.