Active Control of Thermoacoustic Instability in a Model Combustor with Neuromorphic Evolvable Hardware

John C. Gallagher and Saranyan Vigraham

Department of Computer Science and Engineering
Wright State University, Dayton, OH, 45435-0001
{jgallagh,svigraha}@cs.wright.edu

Abstract. Continuous Time Recurrent Neural Networks (CTRNNs) have previously been proposed as an enabling paradigm for evolving analog electrical circuits to serve as controllers for physical devices [6]. Currently underway is the design of a CTRNN-EH VLSI chips that combines an evolutionary algorithm and a reconfigurable analog CTRNN into a single hardware device capable of learning control laws of physical devices. One potential application of this proposed device is the control and suppression of potentially damaging thermoacoustic instability in gas turbine engines. In this paper, we will present experimental evidence demonstrating the feasibility of CTRNN-EH chips for this application. We will compare our controller efficacy with that of a more traditional Linear Quadratic Regulator (LQR), showing that our evolved controllers consistently perform better and possess better generalization abilities. We will conclude with a discussion of the implications of our findings and plans for future work.

1 Introduction

An area of particular interest in modern combustion research is the study of lean premixed (LP) fuel combustors that operate at low fuel-to-air ratios. LP fuels have the advantage of allowing for more complete combustion of fuel products, which decreases harmful combustor emissions that contribute to the formation of acid rain and smog. Use of LP fuels however, contributes to flame instability, which causes potentially damaging acoustic oscillations that can shorten the operational life of the engine. In severe cases, flame-outs or major engine component failure are also possible. One potential solution to the thermoacoustic instability problem is to introduce active control devices capable of sensing and suppressing dangerous oscillations by introducing appropriate control efforts.

Because combustion systems can be so difficult to model and analyze, self-configuring evolvable hardware (EH) control devices are likely to be of enormous value in controlling real engines that might defy more traditional techniques. Further, an EH controller would be able to adapt and change online, continuously optimizing its control over the service life of a particular combustor. This paper

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 431–441, 2003.

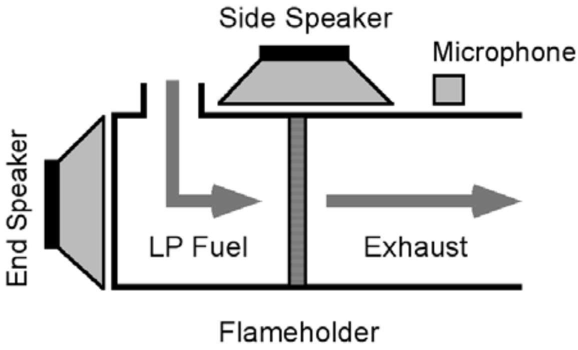

Fig. 1. Schematic of a Test Combustor

will discuss our efforts to control the model combustor presented in [10] [11] with a simulated evolvable hardware device. We will begin with brief summaries of the simulated combustor and our CTRNN-EH device. Following, we will discuss our evolved CTRNN-EH control devices and how their performance compares to a traditional LQR controller. Finally, we will discuss the implications of our results and discuss future work in which we will apply CTRNN-EH to the control of real engines.

2 The Model Combustor

Figure 1 shows a schematic of a simple combustor. Premixed fuel and air is introduced at the closed end and the flame is anchored on a perforated disk mounted inside the chamber a short distance from the closed end (the flameholder). Combustion products are forced out the open end. Thermoacoustic instability can occur due to positive feedback between combustion dynamics of the flame and acoustic properties of the combustion chamber. Qualitatively speaking, flame dynamics are affected by mechanical vibration of the combustion chamber and mechanical vibration of the combustion chamber is affected by heat release/flame dynamics. When these two phenomena reinforce one another, it is possible for the vibrations of the combustion chamber to grow to unsafe levels. Figure 2 shows the engine pressure with respect to time for the first 0.04 seconds of uncontrolled operation of an unstable engine. Note that maximum pressure amplitude is growing exponentially and would quickly grow to unsafe levels. In the model engine, a microphone is mounted on the chamber to monitor the frequency and magnitude of pressure oscillations. A loudspeaker effector used to introduce additional vibrations is mounted either at the closed end of the chamber or along its side. Figure 1 shows both speaker mounting options, though for any experiment we discuss here, only one would be used at a time.

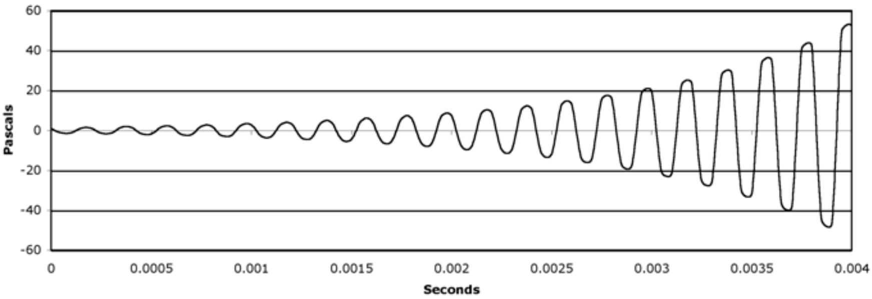

Fig. 2. Time Series Response of the Uncontrolled EM1 Combustor

A full development of the simulation state equations, which have been verified against a real propane burning combustor, is given in [10]. Using these state equations, we implemented C language simulations of four combustor configurations. All four simulations assumed a specific heat ratio of 1.4, an atmospheric pressure of 1 atmosphere, an ambient temperature of 350K, a fuel/air mixture of 0.8, a speed of sound of 350 m/s, and a burn rate of 0.4 m/s. The four engine configurations, designated SM1, SM2, EM1, and EM2, were drawn from [10] and represent speaker side-mount configurations resonant at 542 Hz and 708 Hz and end-mount configurations resonant at 357 Hz and 714 Hz respectively.

3 CTRNN-EH

CTRNN-EH devices combine a reconfigurable analog continuous time recurrent neural network (CTRNN) and Star Compact Genetic Algorithm (*CGA) into a single hardware device.

CTRNNs are networks of Hopfield continuous model neurons [2][5][12] with unconstrained connection weight matrices. Each neuron's activity can be expressed by an equation of the following form:

$$\tau_i \frac{dy_i}{dt} = -y_i + \sum_{j=1}^{N} w_{ji} \sigma \left(y_j + \theta_j \right) + s_i I_i \left(t \right) \tag{1}$$

where y_i is the state of neuron i, τ_i is the time constant of neuron i, w_{ji} is the connection weight from neuron j to neuron i, $\sigma\left(x\right)$ is the standard logistic function, θ_j is the bias of neuron j, s_i is the sensor input weight of neuron i, and $I_i\left(t\right)$ is the sensory input to neuron i at time t.

CTRNNs differ from Hopfield networks in that they have no restrictions on their interneuron weights and are universal dynamics approximators [5]. Due to their status as universal dynamics approximators, we can be reasonably assured that any control law of interest is achievable using collections of CTRNN neurons. Further, a number of analog and mixed analog-digital implementations are known [13] [14] [15] and available for use.

*CGAs are any of a family of tournament-based modified Compact Genetic Algorithms [9] [7] selected for this application because of the ease in which they may be implemented using common VLSI techniques [1] [8]. The *CGAs require far less memory than other EAs because they represent populations as compact probability vectors rather than as sets of actual bit strings. In this work, we employed the mCGA variation similar to that documented in [9]. The algorithm can be stated as shown in figure 3.

Figure 4 shows a schematic representation of our CTRNN-EH device used in intrinsic mode to learn the control law of an attached device. In this case, the user would provide a hardware or software system that produces a scalar measure (performance score) of the controlled devices effectiveness based upon inputs from some associated instrumentation. This is represented in the rightmost block of Figure 4. The CTRNN-EH device, represented by the leftmost block in the figure, would receive fitness scores from the evaluator and sensory inputs from the controlled device. The CGA engine would evolve CTRNN configurations that monitor device sensors and supply effector efforts that maximized the controlled devices performance.

4 CTRNN-EH Control Experiments

In the experiments reported in this paper, we employed a simulated CTRNN-EH device that contained a five neuron, fully-connected CTRNN as the analog neuromorphic component and a mCGA [8] as the EA component. The CTRNN was interfaced to the combustor as shown in Figure 5. Each neuron received the raw microphone value as input. The outputs of two CTRNN neurons controlled the amplitude and frequency of a voltage controlled oscillator that itself drove the loudspeaker (I.E. The CTRNN had control over the amplitude and frequency of the loudspeaker effector). Speaker excitations could range from 0 to 10 mA in amplitude and 0 to 150 Hz in frequency. The error function (performance evaluator) was the sum of amplitudes of all pressure peaks observed in a period of one second. This error function roughly approximates and produces the same relative rankings that would be produced by using simple hardware to integrate the area under the microphone signal in the time domain. mCGA parameters were chosen as follows: simulated population size of 1023, a maximum tournament count of 100,000, and a bitwise mutation rate of 0.05. Forty CTRNN parameters (five time constants, five biases, five sensor weights, and twenty-five intra-network weights) were encoded as eight bit values resulting in a 320 bit genome. All experiments were run on a 16 node SGI Beowulf cluster.

We ran 100 evolutionary trials for each of the four engine configurations. On average, 589, 564, 529, and 501 tournaments were required to evolve effective oscillation suppression for SM1, SM2, EM1, and EM2 respectively. Each of the the resulting four hundred evolved champions was tested for control efficacy across all

```
1. Initialize probability vector
   for i := 1 to L do p[i] := 0.5

2. Generate two individuals from the vector
   a := generate(p);
   b := generate(p);

3. Let them compete
   winner, loser := evaluate(a, b)

4. Update the probability vector toward the winner
   for i := 1 to L do
      if winner[i] <> loser[i] then
         if winner[i] = 1 then
            p[i] := p[i] + (1 / N)
         else
            p[i] := p[i] - (1 / N)

5. Mutate champ and evaluate
   if winner = a then
      c := mutate(a);
      evaluate(c);
      if fitness(c) > fitness(a) then
         a := c;
   else
      c := mutate(b);
      evaluate(c);
      if fitness(c) > fitness(b) then
         b := c;

6. Generate one individual from the vector
   if winner = a then
      b := generate(p);
   else
      a := generate(p);

7. Check if probability vector has converged
   for i := 1 to L do
      if p[i] > 0 and p[i] < 1 then goto step 3

8. P represents the final solution
```

Fig. 3. Pseudo-code for mCGA

four modeled engine configurations (SM1, SM2, EM1, and EM2). All were effective in suppressing vibrations under the conditions for which they were evolved. In addition, all were capable of effectively suppressing vibrations in the engine configurations for which they were not evolved. Typical engine noise suppression

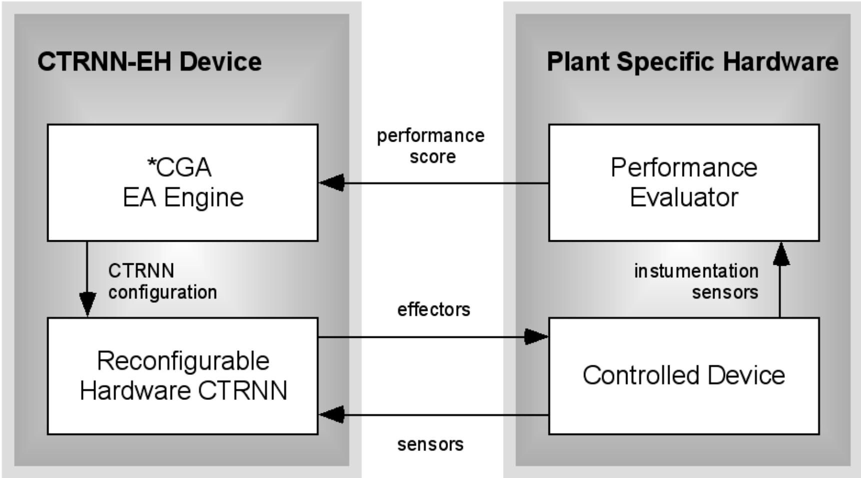

Fig. 4. Schematic of CTRNN-EH Controller

results for both a side mounted CTRNN-EH controller and a Linear Quadratic Regulator (LQR) are shown in Figure 6. Tables 1, 2 3, and 4 summarize the average settling times (the time the controller requires to stabilize the engine) across all experiments. Note that in Figure 6, our evolved controller settles to stability significantly faster than the LQR. The LQR controllers presented in [10] and [11] had settling times of about 40 mS and 20 mS for the end-mounted and side-mounted configurations respectively. Note that our evolved CTRNNs compare very well to LQR devices. On average, they evolved to produce settling times of better than 20 ms. The very best CTRNN controllers settle in as few as 8 ms. Further, the presented LQR controllers failed to function properly when used in a mounting configuration for which they were not designed, while all of our evolved controllers appear capable of controlling oscillations irregardless of where the effector is mounted. Both of these results suggest that our evolved controllers may be both faster (in terms of settling time) and more flexible (in terms of effector placement) than the given LQR devices. Presuming that we implemented only the analog CTRNN portion of the CTRNN-EH device, this improved capability would be achieved without a significant increase in the amount of analog hardware required.

In other, related work, we have observed that mCGA seems better able to evolve CTRNN controllers than the population based Simple Genetic Algorithm (sGA) that it emulates [7]. This effect was observed in experiments reported here as well. We evolved 100 CTRNN controllers for the each engine configuration using a tournament based simple GA with uniform crossover, a bitwise mutation rate of 0.05, and a population size of 1023. On average, the sGA required 5000 tournaments to evolve effective control. The difference between the number of generations required for sGA and mCGA is statistically significant. Table 5 shows

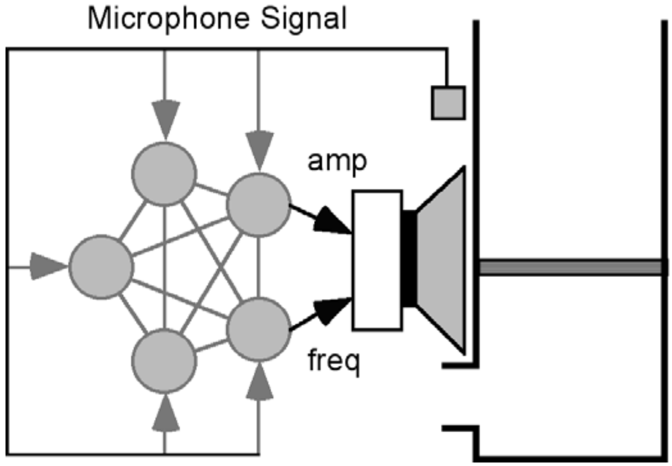

Fig. 5. CTRNN to Combustor Interface

the average settling times of sGA and mCGA controllers evolved in the SM1 configuration. These results are representative of those observed under other evolutionary conditions.

5 Conclusions and Discussion

In this paper, we demonstrated that, against an experimentally verified combustor model, CTRNN-EH evolvable hardware controllers are consistently capable of evolving highly effective active oscillation suppression abilities that generalized to control different engine configurations as well. Further, we demonstrated that we could surpass the performance of a benchmark LQR device reported in the literature as a means of solving the same problem. These results are in themselves significant. More significant, however, are the implications of those results.

First, the LQR devices referenced were developed based upon detailed knowledge of the system to be controlled. A model needed to be constructed and validated before controllers could be constructed. Even in the case of the relatively simple combustion device that was modeled and simulated, this was a significant effort. Though it may be the case that improved control can be had by using other model-based methods, any such improvements would be purchased at the cost of significant additional work. Further, it is not clear that one would be able to construct appropriately detailed mathematical models of more realistic combustor systems with more realistic engine actuation methods. Thus, it is not clear if model-based control methods could be applied to more realistic engines. Our CTRNN-EH controllers were developed without specific knowledge of the plant to be controlled. A *CGA evolved a very general dynamics approximator

Table 1. Controllers Evolved in SM1 Configuration

Statistic	Tested in EM1	Tested in EM2	Tested in SM1	Tested in SM2
Average	12.51 ms	11.80 ms	11.141 ms	11.78 ms
Stdev	5.38 ms	5.22 ms	5.21 ms	1.08 ms

Table 2. Controllers Evolved in EM1 Configuration

Statistic	Tested in EM1	Tested in EM2	Tested in SM1	Tested in SM2
Average	14.68 ms	13.84 ms	13.05 ms	12.20 ms
Stdev	6.37 ms	6.23 ms	5.97 ms	1.14 ms

Table 3. Controllers Evolved in SM2 Configuration

Statistic	Tested in EM1	Tested in EM2	Tested in SM1	Tested in SM2
Average	21.93 ms	21.41 ms	20.06 ms	13.03 ms
Stdev	3.74 ms	3.80 ms	3.92 ms	0.67 ms

Table 4. Controllers Evolved in EM2 Configuration

Statistic	Tested in EM1	Tested in EM2	Tested in SM1	Tested in SM2
Average	13.22 ms	12.53 ms	11.85 ms	11.91 ms
Stdev	5.79 ms	5.58 ms	5.58 ms	1.07 ms

Table 5. Controllers Evolved with sGA in SM1 Configuration

Statistic	Tested in EM1	Tested in EM2	Tested in SM1	Tested in SM2
Average	14.72 ms	17.31 ms	13.65 ms	14.03 ms
Stdev	4.92 ms	5.61 ms	5.16 ms	3.23 ms

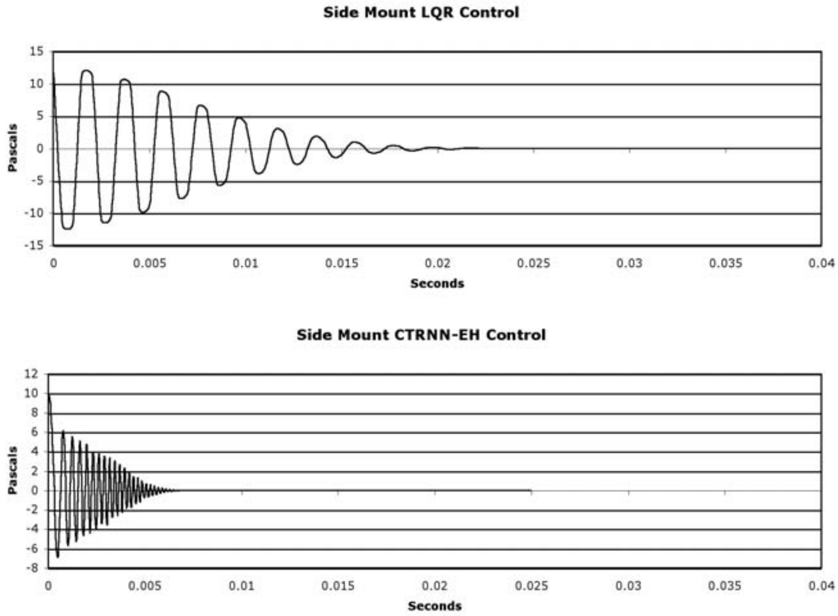

Fig. 6. Typical LQR Response vs. CTRNN-EH Response

to stabilize the engine. Such a technique could be applied without modification to any engine and/or combustor system – with any sort of engine effectors. Naturally, one might argue that the evolved control devices would be too difficult to understand and verify, rendering them less attractive for use in important control applications. However, especially in cases where there are few sensor inputs, we have already developed analysis techniques that should be able to construct detailed explanations of CTRNN operation with respect to specific control problems [3] [4]. The engine controllers we presented in this paper are currently undergoing analysis using these dynamical systems methods and we expect to construct explainations of their operation in the near future.

Second, although our initial studies have been of necessity in simulation, we have made large strides in constructing hardware prototypes on our way to a complete, self-contained VLSI implementation. We have already constructed and verified a reconfigurable analog CTRNN engine using off-the-shelf components [6] and have implemented the mCGA completely in hardware with FPGAs [7]. Our early experiments suggest that our hardware behaves as predicted in simulation. We are currently integrating these prototypes to create the first, fully hardware CTRNN-EH device. This first integrated prototype will be used to evolve oscillation suppression on a physical test combustor patterned after that modeled in [10]. Our positive results in simulation make moving to this next phase possible.

Third, earlier in this paper, we reported that mCGA evolves better solutions than does a similar simple GA. This phenomenon is not unique to the engine control problem, in fact, we have observed it in evolving CTRNN based controllers for other physical processes [7]. Understanding why this is the case will likely lead to important information about the nature of CTRNN search spaces, the mechanics of the *CGAs, or both. This study is also currently underway.

Evolvable hardware has the potential to produce computational and control devices with unprecedented abilities to automatically configure to specific requirements, to automatically heal in the face of damage, and even to exploit methods beyond what is currently considered state of the art. The results in this paper argue strongly for the feasibility of EH methods to address a difficult problem of practical import. They also point the way toward further study and development of general techniques of potential use to the EH community.

Acknowledgements. This work was supported by Wright State University and The Ohio Board of Regents through the Research Challenge Grant Program.

References

1. Aporntewan, C. and Chongstitvatana. (2001). A hardware implementation of the compact genetic algorithm. in *The Proceedings of the 2001 IEEE Congress on Evolutionary Computation*
2. Beer, R.D. (1995). On the dynamics of small continuous-time recurrent neural networks. in *Adaptive Behavior*3(4):469–509.
3. Beer, R.D., Chiel, H.J. and Gallagher, J.C. (1999). Evolution and analysis of model CPGs for walking II. general principles and individual variability. in *J. Computational Neuroscience* 7(2):119–147.
4. Chiel, H.J., Beer, R.D. and Gallagher, J.C. (1999). Evolution and analysis of model CPGs for walking I. dynamical modules. in *J. Computational Neuroscience* 7:(2):99–118.
5. Funahashi, K & Nakamura, Y. (1993), Approximation of dynamical systems by continuous time recurrent neural networks, in *Neural Networks* 6:801–806
6. Gallagher, J.C. & Fiore, J.M., (2000). Continuous time recurrent neural networks: a paradigm for evolvable analog controller circuits, in *The Proceedings of the 51st National Aerospace and Electronics Conference*
7. Gallagher, J.C., Vigraham, S., Kramer, G. (2002). A family of compact genetic algorithms for intrinsic evolvable hardware. Submitted to *IEEE Transactions on Evolutionary Computation*
8. Gallagher, J.C. & Vigraham, S. (2002). A modified compact genetic algorithm for the intrinsic evolution of continuous time recurrent neural networks. in *The Proceedings of the 2002 Genetic and Evolutionary Computation Conference*. Morgan-Kaufmann.
9. Harik, G., Lobo, F., & Goldberg, D.E. (1999). The compact genetic algorithm. In *IEEE Transactions on Evolutionary Computation*. Vol 3, No. 4. pp. 287–297

10. Hathout, J.P., Annaswamy, A.M., Fleifil, M. and Ghoniem, A.F. (1998). Model-based active control design for thermoacoustic instability. in *Combustion Science and Technology*, 132: 99–138
11. Hathout, J.P., Fleifil, M., Rumsey, J.W., Annaswamy, A.M., and Ghoniem, A.F. (1997). Model-based analysis and design of active control of thermoacoustic instability. in *IEEE Conference on Control Applications*, Hartford, CT, October 1997.
12. Hopfield, J.J. (1984). Neurons with graded response properties have collective computational properties like those of two-state neurons, in *Proceedings of the National Academy of Sciences* 81:3088–3092
13. Maass, W. and Bishop, C. (1999). *Pulsed Neural Networks.* MIT Press.
14. Mead, C.A., (1989). *Analog VLSI and Neural Systems*, Addison-Wesley, New York
15. Murray, A. and Tarassenko, L. (1994). *Analogue Neural VLSI : A Pulse Stream Approach.* Chapman and Hall, London.