An Evolvable Hardware Chip and Its Application as a Multi-Function Prosthetic Hand Controller

Isamu Kajitani and **Tsutomu Hoshino**
University of Tsukuba
1-1-1 Tennou-dai, Tsukuba, Ibaraki, 305-8573, JAPAN
i_kajita@etl.go.jp

Nobuki Kajihara
Adaptive Devices NEC Laboratory, RWCP

Masaya Iwata and **Tetsuya Higuchi**
Electrotechnical Laboratory

Abstract

This paper describes the application of genetic algorithms to the biomedical engineering problem of a multi-function myoelectric prosthetic hand controller. This is achieved by an innovative LSI chip (EHW chip), i.e., a VLSI implementation of Evolvable Hardware (EHW), which can adapt its own circuit structure to its environment autonomously and quickly by using genetic algorithms. Usually, a long training period (almost one month) is required before multi-function myoelectric prosthetic hands can be controlled, however, the EHW chip controller developed here can reduce this period and it has been designed for easy implementation within a prosthetic hand. There are plans to commercialize the prosthetic hand with the EHW chip, and the medical department of Hokkaido University has already decided to adopt this for clinical treatment.

Introduction

In contrast to conventional hardware, where the structure is irreversibly fixed in the design process, Evolvable Hardware (EHW) (Higuchi *et al.* 1993) is designed to adapt to changes in task requirements or changes in the environment, through its ability to reconfigure its own hardware structure dynamically and autonomously. This capacity for adaptation, achieved by employing efficient search algorithms known as genetic algorithms (GAs)(Goldberg 1989), has great potential for the development of innovative industrial applications.

Although most works on EHW citeYao98 have been done with software simulations, this paper presents a VLSI implementation of EHW (EHW chip) and its application to the biomedical engineering problem of a multi-function myoelectric prosthetic hand controller.

In designing this EHW chip, we have modified the genetic operations used in (Kajitani *et al.* 1998) to include a gene replacement operation to accelerate the adaptation speed of the EHW chip, which we refer to as Gene Replacement Genetic Algorithm (GRGA).

The myoelectric prosthetic hand is operated by the signals generated in muscular movement (electromyography, EMG). However, it takes a long time, usually almost one month, before a disabled person is able to control a multi-function prosthetic hand freely (Uchida, H, & Ninomija 1993). During this period, the disabled person has to undertake training to adapt to the myoelectric hand. We hope to reverse this situation, by having the myoelectric hand adapt itself to the disabled person.

Although, work is being done on applying neural networks for adaptable prosthetic hand controllers, this approach is not very promising due to implementation problems, because systems using neural networks are large and thus difficult to implement within a prosthetic hand.

In contrast, the system using the EHW chip is suitable for this kind of application, because of its compactness and high-speed adaptability, and, in this paper, we show that the EHW chip controller is a viable alternative to neural network controllers.

There are plans to commercialize the prosthetic hand with the EHW chip, and the medical department of Hokkaido University has already decided to adopt this for clinical treatment.

This paper is organized as follows. Section 2 provides some background to this research. In Section 3, the basic idea of EHW is explained. Section 4 introduces the EHW chip, and its application to the controller for a prosthetic artificial limb, which is presented in Section 5.

Background

EHW is based on the idea of combining a reconfigurable hardware device with GAs to execute reconfiguration autonomously (Higuchi *et al.* 1993).

In conventional works on EHW (Yao & Higuchi 1998), genetic operations are carried out with software on personal computers (PCs) or workstations (WSs). This makes it difficult to use EHW in situations that need circuits to be as small and light as possible. For

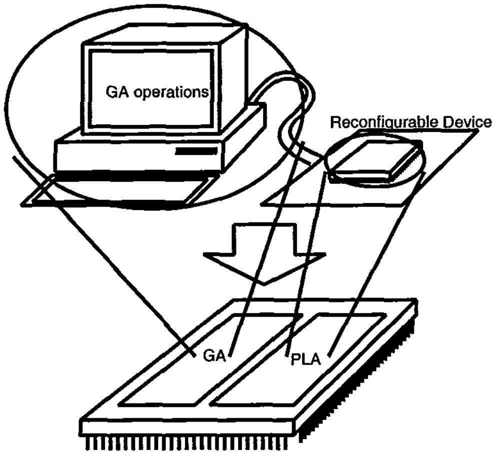

Figure 1: The basic idea of the EHW chip

example, a myoelectric prosthetic hand should be the same size as a human hand and weight less than 700 gram.

One solution to this is to incorporate the hardware that carries out the GA operations together with the reconfigurable hardware logic within a single LSI chip, as shown in Figure 1. Such compact and quickly reconfigurable EHW chips can serve as off-the-shelf devices for practical applications that require on-line hardware reconfiguration.

This paper describes the EHW chip, which is actually an improved version of the chip developed in (Kajitani *et al.* 1998)(Figure 2). The two improvements to this EHW chip are: 1. Speed of adaptation. 2. *On-line* circuit synthesis.

(Kajitani *et al.* 1998) demonstrated the possibility of employing an EHW chip as the pattern classification circuit for EMG signals in a multi-function myoelectric prosthetic hand controller.

In that application, the pattern classification circuit was synthesized *off-line*, in two distinct phases (i.e. "input-output pattern training sample phase" and "circuit synthesis with GA phase"). Although this *off-line* approach is used in most adaptable prosthetic hand applications of neural network controllers (such as (Kelly, Parker, & Scott 1990)), this approach is often ineffective due to changes in EMG signal features after construction of a training sample (Ito *et al.* 1991) (Fujii, Nishikawa, & Yokoi 1998).

To overcome this problem, we apply an *on-line* approach to the EHW chip prosthetic hand controller.

The Basic Idea of EHW

The basic idea of EHW is the combination of a reconfigurable hardware device and GAs (Higuchi *et al.* 1993).

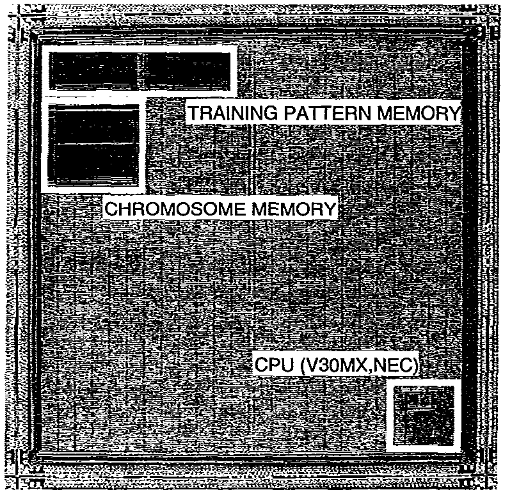

Figure 2: The earlier version of the EHW chip.(Kajitani *et al.* 1998).

PLA (Programmable Logic Array, Figure 3) is most commonly used as the reconfigurable hardware device *. A PLA consists of an AND-array and an OR-array as shown in Figure 3. In Figure 3, the black and white circles indicate switches, which determine the interconnections between the inputs and outputs (the black circles indicate connections). The row lines (product term lines) in the AND-array form logical products of connected inputs, and the column lines in the OR-array form logical sums of the connected row lines of the AND-array (i.e. product term lines). We can specify these switch settings by using a configuration bit string as shown in Figure 3.

The basic concept behind the combination of GAs and the PLA in EHW is to regard the configuration bit strings for the PLA as chromosomes for the GAs (Figure 4). If a fitness function is properly designed for a task, then the GAs can autonomously find the best hardware configuration in terms of the chromosomes (i.e. configuration bits).

Usually, a training sample of input-output patterns (e.g. Table 1) is used to evaluate chromosomes (Higuchi *et al.* 1993). In this case, the fitness value for a chromosome (i.e. circuit candidate) is the output pattern rate, that is, the rate at which actual output corresponds to the expected output pattern for a given training input pattern.

The EHW Chip.

The improvements to this EHW chip are in "Speed of adaptation" and "*On-line* circuit synthesis." These

*Other EHW works use special hardware, in which the circuit structure can be changed by arithmetic functional blocks or analogue circuit components.

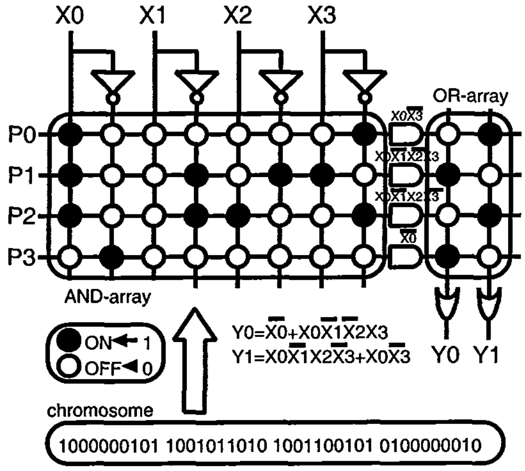

Figure 3: The basic structure of the PLA.

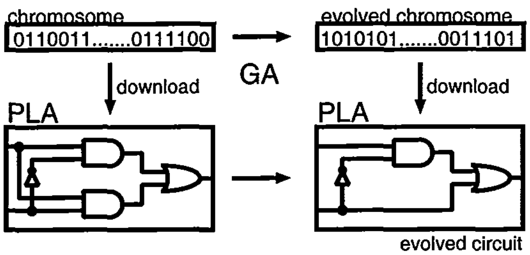

Figure 4: The basic idea of EHW.

improvements are achieved through two modifications; firstly, by employing GRGA (Gene Replacement Genetic Algorithm), and secondly, by adding an extra "*on-line* training pattern memory edit" mode, which is indispensable for an application such as the prosthetic hand controller.

Architecuture and Workflow of The EHW Chip.

The EHW chip consists of the following six functional blocks, shown in Figure 5.

- GA UNIT : This is the hardware that carries out the GA operations.
- CHROMOSOME MEMORY, TRAINING PATTERN MEMORY and MEMORY for FITNESS VALUE: These are the memories for the chromosomes, the training samples and the fitness values of each chromosome, respectively.
- PLA UNIT (2 arrays): Two PLAs are for parallel evaluation of two circuits.
- 16 bit CPU (8086 compatible, V30MX(NEC)): This is used as the interface between outside and inside the chip. It can be used to calculate fitness values for each circuit without using training patterns.

The adaptation of the EHW chip is carried out in the following way.

1. The GA UNIT reads two chromosomes from the CHROMOSOME MEMORY in units of 32 bits, and applies genetic operations on them to make two segments (32 bits) from the chromosomes.
2. These two segments are written to the PLA UNIT and are used to implement a circuit on both of the two PLAs. The two circuits are then evaluated.
3. Evaluations of the circuits on the PLAs are carried out by using the training samples, which are read from the TRAINING PATTERN MEMORY, and the fitness values are written to the MEMORY for FITNESS VALUE.

Table 1: Training input-output patterns and an example of fitness value.

training pattern		
input pattern $X0X1X2X3$	output pattern	output pattern of the circuit in Figure 3. $Y0Y1$
1001	10	10
0001	10	10
0101	10	10
1100	01	01
0011	10	10
0000	00	10
0100	10	10
0010	10	10
1010	00	01
1110	01	01
fitness value		0.8

GRGA (Gene Replacement Genetic Algorithm)

The basic idea of GRGA is to accelerate the genetic search by replacing a part of a chromosome with a bit string, referred to as the "chromosome candidate segment." In this application, the chromosome candidate segment is generated from a training input-output pattern used for the evaluation of circuit candidates, as shown in Figure 6.

To implement this replacement operation on the EHW chip, we combined this operation with the ER (Elitist Recombination)(Thierens & Goldberg 1994) and the UC (Uniform Crossover) operations used in the earlier version of the EHW chip (Kajitani *et al.* 1998) and have named this combination of genetic operations GRGA. This replacement operation is carried out when the output pattern for a training input pattern does not match the expected training output pattern, with the replacement of the chromosome candidate segment being carried out with a fixed probability.

For example, generation of a chromosome candidate segment, in the case of the first training pattern in Table 1 (expected output pattern for the input pattern "1001" is "10"), would proceed as follows.

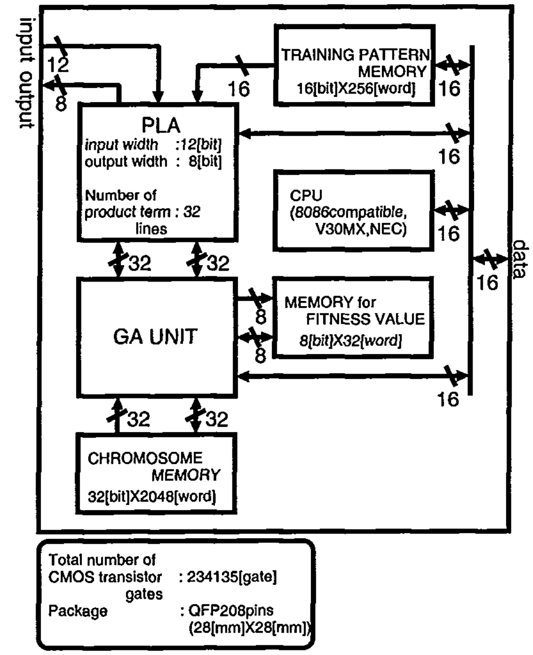

Figure 5: Block diagram of the EHW chip.

When the switches of a product term line in the PLA are set as they are in line P1 in Figure 3, the output pattern for the input pattern "1001" is "10," and the output patterns for all the other input patterns are "00". These switches can be set with the bit string "1001011010", which is treated as a chromosome candidate segment.

We present the results from simulations carried out to evaluate the adaptability of the GRGA. These simulated the number of evaluations required to synthesize a basic combination circuit for both the GRGA and for the combination of the ER and the UC operations. The three bit comparator circuit (six input bits and one output bit) was used as the target circuit in the simulations, because this circuit requires a long time to be synthesized by GAs (Higuchi *et al.* 1995) (Kajitani *et al.* 1996).

The result was that the combination of the ER and the UC operations took about 61874 evaluations (averaged ten times), and the GRGA took about 9710 evaluations (averaged ten times) to synthesize the circuit. This indicates that the evaluation time for the combination of the ER and the UC operations is over sixth times longer than that required with the GRGA.

On-line Training Pattern Memory Edit Mode.

In the earlier EHW chip, the TRAINING PATTERN MEMORY could not be edited after the GA UNIT had

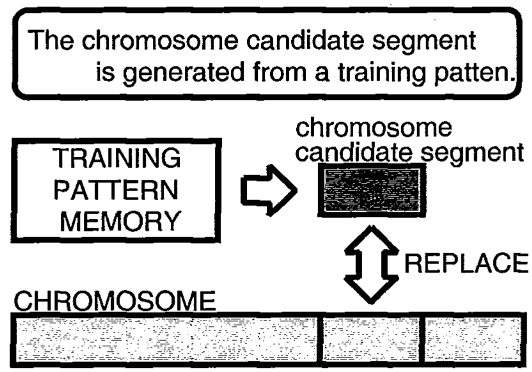

Figure 6: The basic idea of the GRGA.

begun to operate. Therefore, training sample patterns had to be made *off-line*(Kajitani *et al.* 1998).

To overcome this problem, we have incorporated an "*On-line* training patter n edit mode" within the chip. This allows us to terminate the genetic operation s when necessary, so that the TRAINING PATTERN MEMORY can be edited *on-line* , The ability to edit the TRAINING PATTERN MEMORY *on-line* helps to ensur e a smooth adaptation process for the prosthetic hand controller.

Prosthetic Hand Control By The EHW Chip

EMG Pattern Classification with The EHW Chip

The purpose of the EHW controller in the prosthetic hand is to synthesize pattern classification hardware to map input patterns (i.e. feature vectors of the two channel EMG signals, which are detected by two sensors (Kajitani *et al.* 1998), to desired actions of the hand (i.e. one of six actions in Figure 7). However, because EMG signals vary greatly between individuals, it is impossible to design in advance such a control (classification) circuit. Furthermore, even for a particular person, feature vectors of the EMG signals sometimes change even over short periods (Ito *et al.* 1991) (Fujii, Nishikawa, & Yokoi 1998). Therefore, the control hardware circuit must be synthesized adaptively.

Problems With Conventional Works

In conventional work on adaptable prosthetic hands, there are two problems.

1. Inefficiency of *off-line* construction of training sample patterns.
2. Computational overheads of preprocessing EMG signals.

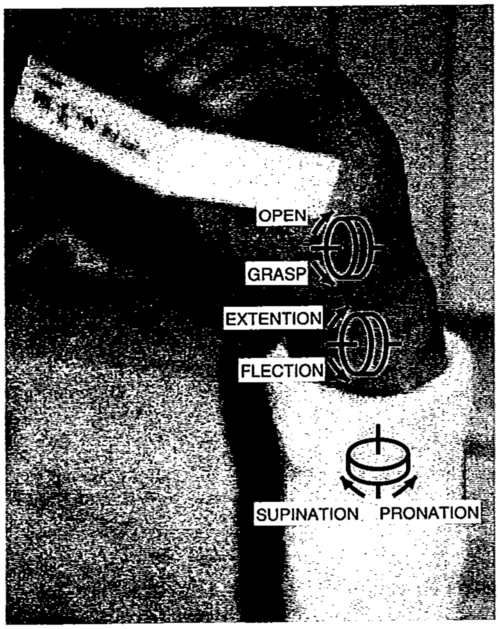

Figure 7: The artificial prosthesis used in our experiments.

Inefficiency of *Off-line* Construction of Training Sample Patterns. Although most adaptable prosthetic hand controllers using either neural networks (such as (Kelly, Parker, & Scott 1990)(Hudgins, Parker, & Scott 1993)) or the EHW chip (Kajitani *et al.* 1998) have taken an *off-line* approach to training, this is often ineffective due to changes in EMG signal features after construction of training sample patterns (Ito *et al.* 1991) (Fujii, Nishikawa, & Yokoi 1998).

To overcome this problem, an *on-line* approach has been applied to neural network controllers (Ito *et al.* 1991)(Fujii, Nishikawa, & Yokoi 1998), and, in this paper, we also adopt this *on-line* approach to the prosthetic hand controller using the EHW chip.

If the prosthetic hand fails to function as the user intends, it may be due to changes in EMG signal features. With this *on-line* approach, we can supplement the set of training samples with new patterns and can reconfigure the pattern classification circuit accordingly.

Computational Overheads of Preprocessing EMG Signals. In (Fujii, Nishikawa, & Yokoi 1998)(Kajitani *et al.* 1998), the frequency spectrum power of the detected EMG signals was used as the feature vectors of the EMG signals. Usually, the frequency spectrum powers are calculated using FFT (Fast Fourier Transform) that needs a high performance CPU (e.g. Pentium) or a DSP (Digital Signal Processor) to carry out calculations quickly. However, in general, systems using high performance CPUs or DSPs become large, and this makes it difficult to implement them within the prosthetic hand.

Therefore, we have decided to use integrated EMG(IEMG) signals (Ito *et al.* 1991), which are calculated by integrating the absolute value of a EMG signal for each channel within a fixed period (one second, in this paper), as the feature vectors of the EMG signals, in this prosthetic hand controller. These IEMGs are converted into four bit binary numbers to be input signals to the PLAs in the EHW chip.

Experiments

Overview This section explains some experiments on the synthesis of a pattern classification circuit for the EMG feature vectors. In this experiment, because the EHW chip introduced in this paper is still in debugging stage, its simulator on a PC (Pentium Pro, 200MHz) was used.

This experiment consisted of the following six stages.

1. Construction of a training sample of input-output patterns (sixty patterns; 10[pattern]X6[action]).
2. Circuit synthesis with GRGA for five minutes.
3. Test of the synthesized circuit.
4. Construction of additional training sample patterns (thirty patterns; 10[pattern]X3[action]).
5. Reconfiguration of the circuit with GRGA for five minutes.
6. Test of the reconfigured circuit.

Training Pattern Construction. A training sample of input-output patterns consists of the input patterns, i.e., binary expressions of the amplitude of the EMG signals, and the expected output patterns, which determine the action of the prosthetic hand (one of six actions). Training patterns were generated in the following way for each of the six prosthetic hand actions.

1. Envisage one of the prosthetic hand actions, and contract remnant muscles.
2. Enter key corresponding to the action. This operation generates ten training sample patterns.

The pattern classification circuit is synthesized using these training sample patterns (10[pattern] X 6[action] = 60[pattern]).

Test of The Synthesized Circuit. The synthesized circuit was tested in the following way.

1. Envisage one of the prosthetic hand actions, and contract remnant muscles.
2. Enter key corresponding to the action. This operation calculates the output pattern rate for the synthesized circuit, which is the same as the expected output pattern for an intended action, during ten seconds.

Table 2: Output pattern rates of synthesized circuit, which correspond to expected output patterns (averaged for three people).

	before training pattern addition (%)	after training pattern addtion (%)
SUPINATION	66	74
PRONATION	49	72
FLECTION	67	88
EXTENSION	84	95
GRASP	38	75
OPEN	36	84
AVERAGE	57	81

The results of this test are shown in Table 2 (middle column).

Then, ten additional training sample patterns are generated for the three actions with the lowest output pattern rates for the synthesized circuit. The circuit on the PLA is reconfigured using these training sample patterns (60[pattern] + 10[pattern] X 3[action] = 90[pattern]), and the reconfigured circuit is tested again. The results of this test are shown in Table 2 (right column).

Results The averaged output pattern rate (Table 2) for the EHW chip controller is 81.0[%]. In contrast, the averaged output pattern rate for neural network controllers, which are learned with training samples generated by an *on-line* approach (Fujii, Nishikawa, & Yokoi 1998), is 81.5[%]. These results indicate that the EMG pattern classification with the EHW chip is a viable alternative to neural networks.

Conclusion

This paper has described the EHW chip and its application as a myoelectric prosthetic hand controller. Recent improvements to the EHW chip in terms of both increased adaptation speeds and the addition of an *online* edit mode have greatly enhanced the performance of the EHW chip controller. Our software simulations show that this is a viable alternative to neural network controllers, and that prosthetic hands with the EHW chip can adapt to users in a short period (about ten minutes).

The EHW chip can adapt its circuit structure autonomously and quickly, and therefore, represents a breakthrough for applications that require compact implementation and high-speed adaptation (such as autonomous mobile vehicles (Keymeulen *et al.* 1998)).

Acknowledgments

This work is supported by MITI Real World Computing Project.We thank Dr. Otsu and Dr. Ohmaki at Electrotechnical Laboratory and Dr. Shimada of RWCP for their supports.

References

Fujii, S.; Nishikawa, D.; and Yokoi, H. 1998. Development of prosthetic hand using adaptable control method for human characteristics. In *IAS-5*, 360–367.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

Higuchi, T.; Niwa, T.; Tanaka, T.; Iba, H.; de Garis, H.; and Furuya, T. 1993. Evolvable hardware with genetic learning: A first step towards building a darwin machine. In *Proceedings of 2nd International Conference on the Simulation of Adaptive Behavior*, 417–424. MIT Press.

Higuchi, T.; Iwata, M.; Kajitani, I.; Iba, H.; Hirao, Y.; Furuya, T.; and Manderick, B. 1995. Evolvable hardware and its applications to pattern recognition and fault-tolerant systems. In Sanchez, E., and Tomassini, M., eds., *Towards Evolvable Hardware.* Springer.

Hudgins, B.; Parker, P.; and Scott, R. N. 1993. A new strategy for multifunction myoelectric control. *IEEE Transaction on Biomedical Engineering* 40(1):82–94.

Ito, K.; Tsuji, T.; Kato, A.; and Ito, M. 1991. Limb-function discrimination using emg signals by neural network and application to prosthetic forearm control. In *Proceedings of the IJCNN91*, 1214–1219.

Kajitani, I.; Hoshino, T.; Iwata, M.; and Higuchi, T. 1996. Variable length chromosome ga for evolvable hardware. In *Proceedings of ICEC96.*

Kajitani, I.; Hoshino, T.; Nishikawa, D.; Yokoi, H.; Nakaya, S.; Yamauchi, T.; Inuo, T.; Kajiwara, N.; Keymeulen, D.; Iwata, M.; and Higuchi, T. 1998. A gate-level ehw chip; implementing ga operations and reconfigurable hard ware on a single lsi. In *Evolvable Systems: From Biology to Hardware, Lecture Notes in Computer Science 1478, Springer Verlag*, 1–12.

Kelly, M. F.; Parker, P. A.; and Scott, R. N. 1990. The application of neural networks to myoelectric signal analysis: Preliminary study. *IEEE Transaction on Biomedical Engineering* 37(3):221–230.

Keymeulen, D.; Konaka, K.; Iwata, M.; Kuniyoshi, Y.; ; and Higuchi, T. 1998. Robot learning using gate-level evolvable hardware. In *Sixth European Workshop on Learning Robots (EWLR-6).*

Thierens, D., and Goldberg, D. 1994. Elitist recombination: an integrated selection recombination ga. In *Proceedings of First IEEE conference on Evolutionary Computation*, 508–512.

Uchida, M.; H, I.; and Ninomija, S. P. 1993. Control of a robot arm by myoelectric potential. In *Journal of Robotics and Mechatronics vol.5 no.3*, 259–265.

Yao, X., and Higuchi, T. 1998. Promises and challenges of evolvable hardware. *IEEE Transaction on Systems, Man, and Cybernetics, Part C* 28(4).