Parallel Distributed Computational Microcontroller System for Adaptive Antenna Downlink Transmitter Power Optimization

K. Prajindra Sankar, S.K. Tiong, and S.P. Johnny Koh

Abstract—This paper presents a tested research concept that implements a complex evolutionary algorithm, genetic algorithm (GA), in a multi-microcontroller environment. Parallel Distributed Genetic Algorithm (PDGA) is employed in adaptive beam forming technique to reduce power usage of adaptive antenna at WCDMA base station. Adaptive antenna has dynamic beam that requires more advanced beam forming algorithm such as genetic algorithm which requires heavy computation and memory space. Microcontrollers are low resource platforms that are normally not associated with GAs, which are typically resource intensive. The aim of this project was to design a cooperative multiprocessor system by expanding the role of small scale PIC microcontrollers to optimize WCDMA base station transmitter power. Implementation results have shown that PDGA multi-microcontroller system returned optimal transmitted power compared to conventional GA.

Keywords—Microcontroller, Genetic Algorithm, Adaptive antenna, Power optimization.

I. INTRODUCTION

ONE of the main techniques to increase mobile communication system capacity is by implementing adaptive antenna into the system. Adaptive antenna beam is capable of adjusting traffic patterns at a given time to increase signal strength and quality. To adjust for frequency and channel use, the adaptive antenna uses multiple antenna elements and appropriately combines the signal received from different antenna elements to form single output. There are many existing algorithm for adaptive beam forming but lately much attention was drawn to algorithm with genetic algorithm that results in optimum beam forming performance [1], [2].

The optimization of the antenna array is typically subjected to a range of constraints and the classic methods of array weight optimization may get trapped in local optima and resulting in a suboptimum beam forming performance [3]. Therefore, GA is adapted for beam forming weight optimization problem as it is more robust and may offer

K.Prajindra Sankar is with the Department of Electronic and Communication Engineering, Universiti Tenaga Nasional, Km7, Kajang-Puchong, 43009 Kajang, Malaysia (e-mail: Sankar@ uniten.edu.my).

S.K.Tiong is with the Department of Electrical Power Engineering, Universiti Tenaga Nasional, Km7, Kajang-Puchong, 43009 Kajang, Malaysia (e-mail: siehkiong@uniten.edu.my).

S.P. Johnny Koh is with the Department of Electronic and Communication Engineering, Universiti Tenaga Nasional, Km7, Kajang-Puchong, 43009 Kajang, Malaysia (e-mail: JohnnyKoh@uniten.edu.my).

significant benefits over more typical optimization techniques such as linear programming, heuristic, etc.

The calculation cost is high in conventional GA because all chromosomes are evaluated throughout all generations and the population size remains the same throughout all generations. To reduce the computation load of genetic search, a lot of methods for searching in parallel and distributed manner have been studied [5]-[7]. In Parallel Distributed GA (PDGA), GA runs in parallel in two or more subgroups of population called islands, and searching is executed independently in each island. PDGA has better search performance and the computation time is shorter than conventional GA [4]. This model avoids the overhead of central control and presents a nature of parallelism, which makes it easy to implement this model on multiprocessor environment and drastically speed up evolution process [8].

Parallel distributed computational microcontroller system for downlink transmitter power optimization is a cooperative multiprocessor device using multiple SoCs (system on a chip), referred to as microcontrollers, to optimize power usage at WCDMA base station with the help of GA. The primary motivation is to design a hardware that is modular, inexpensive, but capable of solving a complex optimization algorithm. Optimization is performed by clusters of microcontrollers, each microcontroller having independent capability to perform the task. When integrated with other microcontrollers providing identical task solving abilities using distributed parallel model, they will provide a powerful suite of tools to solve complex problems. Multiprocessor architecture is employed to enhance the computing time as multiple processors can process data much faster compared to a single processor. By using microcontrollers, the embedded architecture becomes more cost-effective compared to conventional desktop system.

Each slave microcontroller is connected via a serial bus using USART module to the master microcontroller which is connected to a host computer via the USB port. This is so that the host computer can offload the genetic algorithm processing required for the WCDMA simulator to the multiprocessor device. The design consisted of three major components. The first was to design a hardware system using microcontrollers in which one designated controller (master) could communicate with the others (slave) and interacts with a host computer that runs the WCDMA simulator. The second

step was to design a protocol for the communication system for the controllers and finally, write firmware for each controller to give each one a role in processing GA.

The primary goal of this project was to design a cooperative multiprocessor device that would be used by a host computer to offload a part of a program that would benefit from being processed on a multiple processor system. It would be a great advantage to develop microcontrollers with an evolutionary capability to find solutions to such problems that could be integrated within a cluster. In addition to this, the device must have a very high performance to power consumption ratio so that many processors can be added for greater processing ability without needing special power and cooling requirements. The processor also must be readily available and cheap.

The organization of the paper is as follows. Sect. II gives the details of the dynamic WCDMA simulator. Sect. III explains the hardware specifications which consist of the basic processing units and the design architecture. This is followed by a detailed description on the firmware which encompasses the genetic algorithm processes, the fitness function for beam forming application and communication protocols in Sect. IV. Sect. V describes the actual implementation. Finally, Sect. VI concludes the whole paper by summarizing main points and discusses future works.

II. WCDMA SIMULATOR

In this research, adaptive antenna configurations are adopted from [3] for implementation purpose. It is assumed that a linear circular array of L elements is placed at the far field of M uncorrelated point sources and the elements are separated by a distance of d = $\lambda/2$, where λ is the wavelength of the sources. Thus, GA will be employed to optimize a set of L complex array weights based on the fitness function.

It is assumed that the configurations of adaptive antenna to be a 9-array antenna ($L = 9$) which is placed in circular. Each antenna array can cover an area of maximum 40 degree angle of its own which sums up to an overall of 360 degree angle. The unit equipment (UE) represents the user in the network and is generated randomly by the simulator according to a particular protocol. The randomized UEs with their Direction of Arrival (DOA) in angles are to be offloaded to the microcontroller system via USB to optimize beam forming angles. The WCDMA simulator can generate up to 5000 randomized UEs but if all the angles were to be sent to the microcontrollers in real value numbers, it would be impossible for the controller to cater for such large incoming data with such meager memory space. As an alternative, all the real value numbers will be encoded using binary representation and this will be elaborated further in the next section.

The chromosome for genetic algorithm represents the weights (starting and ending angle) of antenna beam to cover the UEs. Real number representations are used to represent each weight solution and there are 9 arrays, so a total of 18 real value numbers will be used in each individual. The distribution of UEs among microcontrollers will be explained further in Sect. III.

A. Binary Encoded UEs

A simple way to safe memory space and also to address all the randomized UEs is by representing each angle in 360° by one bit as illustrated in Table I. In this way, only 360 bits are required to represent each angle in a circle which is equivalent to 45 bytes considering one byte equal to 8 bits. By using this encoded scheme, even though there are 5000 UEs generated, all the UEs can be covered easily as the UEs location have to fall within the range of 0° - 360°. In brief, the WCDMA simulator will be sending only 45 bytes of data to the multiprocessor device for any one time optimization. Take note that 360° is equated to 0°.

TABLE I
BINARY ENCODED UEs

Angle	7°	6°	5°	4°	3°	2°	1°	0°
UEs*	0	1	1	0	0	0	1	1
Bit	7 (MSB)	6	5	4	3	2	1	0 (LSB)
Binary Encoded UE	01100011 = 0x63							

* The UEs are represented by '1' if they exist under the particular angle and '0' if otherwise.

III. HARDWARE SPECIFICATIONS

A. Basic Processing Units

Parallel distributed artificial intelligence systems solve problems using multiple, cooperative agents. In these systems, control and information are often distributed among the agents. This reduces the complexity of each agent and allows agents to work in parallel and increases problem solving speed. Also, each agent has resource limitations which could limit the ability of a single agent system to solve large, complex problems. Allowing multiple agents to work on these types of problems may be the only way to realistically solve them.

This section shows the features of the functional basic unit that allow the implementation of parallel distributed genetic algorithm. Three processing units will be able to run the necessary program for optimizing beam forming angles.

Various manufacturers offer wide ranges of microcontrollers for all type of applications. But, without doubt, one of the accepted microprocessors, for their great quality, reliability and low cost is nowadays PIC microcontroller unit manufactured by Microchip Technology Inc, a leading manufacturer in the world of microcontrollers [9].

The selection of PIC (Peripheral Interface Controller) 18F4550, comes marked by its low cost, that allows solutions with a high grade of potential parallelism and provides the minimum necessary requirements: a good processing speed (48Mhz), a suitable word wide (8bits) and a possibility of manipulating data with a sufficient precision (8bits). In addition to that, the PIC18 family supports much larger

program space up to 2MB which is suitable for genetic algorithm implementation.

One added advantage of this PIC18F4550 is it's built in USB peripheral. The PIC18F4550 offers full-speed and low-speed compatible USB Serial Interface Engine (SIE) that allows fast communication between any USB host and the microcontroller [10]. This built in module enables the simulator to offload the genetic algorithm computation to the microcontroller via USB

B. Parallel Distributed Architecture

The selected design is to have multiple processors that use the PIC18F4550 as dedicated CPUs to perform genetic operations. A primary master PIC18F4550 CPU links the slave CPUs via serial bus. This design is to modularize the processor environment that has a single master which receives the UE data from a dynamic WCDMA simulator via USB port and distributes the UEs to appropriate slaves to perform the optimization operations. Fig. 1 illustrates the parallel distributed architecture of the microcontrollers.
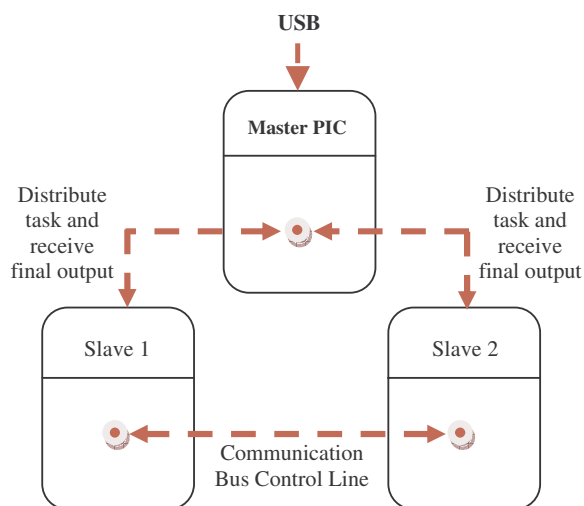

Fig. 1 Parallel Distributed Model

C. Master Microcontroller

The master processor receives the UE data from a dynamic WCDMA simulator via USB. The UEs are distributed first before sent to the slaves using two dedicated pins C6 and C7 (Tx/Rx). The UEs are distributed as follows.

Recall that UEs are user locations in angles (degree) that are generated randomly by the WCDMA simulator. The master processor will segregate the UEs into two classes according to predefined angles. The UE values that falls within the range of 350°~170° (based on maximum coverage angle of antenna element 0-3) will be sent to Slave 1 whereas the balance UEs within the range of 150°~10° (based on maximum coverage angle of antenna element 4-8) will be distributed to Slave 2.

The chromosome represents the weights (starting and ending angle) of antenna beam. Real number representations are used to represent each weight solution and each array requires 2 real value numbers so for 9 antenna elements, 18 real value numbers are needed. Each real value numbers is

encoded to 8 bits. If each of the microcontrollers were to perform optimization on the actual length of the chromosome which is 18 real value numbers (144 bits), then the computation time will be at least doubled and memory resource might be an issue as well. Note that by distributing the UEs, the chromosome length is reduced almost by half which also means faster computation time.

D. Slave Microcontrollers

The slaves are in charge of executing GA optimization in parallel based on UE values the master has sent. It does not start processing information until receives any data from the master. Slave 1 will optimize beam for antenna element 0 to 3 which requires the chromosome length of real value numbers to be 8 (64 bits). Slave 2 will require a chromosome length of 10 real value numbers (80 bits) as it covers antenna element 4 to 8. Slave CPUs solely perform optimization process. Once the slaves are done with the optimization process, the optimized chromosomes are sent back to master to recombine the chromosome. The master processor will then sent the best chromosome to the WCDMA simulator.

IV. FIRMWARE

Once the hardware design architecture layout was chalked out, the application specific firmware had to be written for each of the three microcontrollers. Two of the controllers would receive the distributed chromosome each for GA optimization, and one controller would be designated as a communication node that would pass information to and from the host computer and the slave nodes. The following sections would explain about the GA process, fitness function required for GA evaluation and communication protocols.

A. Genetic Algorithm

To reduce the overhead of central control and global comparison, the UE values are distributed to some microcontrollers. The fitness values of all individuals are calculated in each microcontroller or subpopulation. For a constant number of generations, the subpopulations evolve independently, evaluating their respective individuals and breeding new ones through the process of selection, crossover and mutation.

The selection process for next generation is done by the roulette wheel way with the elite strategy. Following the selection, crossover and mutation are performed based on a distributed environment scheme which doesn't require any parameter settings. In this way, the most prospective species get greatest space to expand and evolve. The followings are outlines of the procedure for slave and master in our scheme:

```
0:   procedure for Master;
1:   while (simulation runs) begin
2:       receive binary encoded UE values;
3:       distribute and send UE values to slave workers;
4:       receive optimized chromosome from both slaves
5:       combine both chromosomes to a single chromosome
6:       send final chromosome to WCDMA simulator (USB)
7:   end while;
```

```
0:   procedure for Slave(s);
1:   begin
2:      receive UE values and a signal to start execution
3:      initialize a random chromosome;
4:      while (termination condition not hold) begin
5:         fitness evaluation;
6:         selection of chromosome for next generation by
7:         Roulette wheel method with the elite strategy;
8:         perform crossover and mutation according to
9:         distributed environment scheme;
10:     end while;
11: end
```

GAs have several parameters which affect both the accuracy of the solutions and the computing time. To get the best parameter settings, many researchers used adaptive methods. For instance, Miki et al. [6] introduced a new method called Distributed Environmental Genetic Algorithms in which the crossover and mutation rates are different in each subpopulation. Using this method, the best solutions are derived automatically in all the trials conducted.

In this research, the GA parameters such as the mutation rate and the crossover rates in each slave are different from each other. With such scheme, the tuning of GA parameters is not necessary and it can be expected that a global optimum can be easily obtained without any pre experiments.

The GA assisted beam forming process requires a fitness function to evaluate the fitness of each individual throughout the evolutionary optimization. The fitness function developed by Tiong *et al.* [3] is integrated with each microcontroller to evaluate the fitness of individual in the aspect of power usage at Node B. The function is shown in Equation 1.

$$F.F = \sum_{l}^{L} \prod_{m}^{M} U_{l,m} + \sum_{m}^{M}(w_m + A_m + \varepsilon P_m) \qquad (1)$$

where,

$$U_l = \begin{cases} 0, & \omega_{m,min} \leq UE_1 \leq \omega_{m,max} \\ 1, & otherwise \end{cases}$$
$$w_m = \begin{cases} 0, & otherwise \\ 1, & \omega_{m,max} - \omega_{m,min} < \rho \end{cases}$$

$\omega_{m,max}$ and $\omega_{m,min}$ are maximum coverage angle and minimum coverage angle for smart antenna, m and ρ refer to minimum separation angle between beam which will be assumed as 5˚.

$$A = \begin{cases} 1, & (\omega_{m,min} \cup \omega_{m,max} \leq (M-1)\delta) \\ & \cup (\omega_{m,min} \cup \omega_{m,max} \leq (M+1)\delta) \\ 0, & (M-1)\delta \leq (\omega_{m,min} \cup \omega_{m,max}) \leq (M+1)\delta \end{cases}$$

$\delta = 360/M$ and M is number of smart antenna elements and;

$$\varepsilon P_m = p_m w_m * 10^{0.25} \qquad (2)$$

Where p_m is poynting power transmitted, ε is weight factor for p_m ($\varepsilon = 0.001$ in this study). w_m is solid angle of antenna beam and cable loss is assumed to be 2.5dB.

B. Communication Protocol

The master microcontroller communicates with its slaves using two dedicated hardware pins for transmit and receiving data which is shared among the two slaves. First, the UEs are received from the WCDMA simulator and then distributed to the respective slaves. Subsequently, the slave processors will perform genetic algorithm computation. After the slaves are done with the computation, the optimized chromosomes are sent back to the master processor. Since both the slaves are sharing the bus line, there is a need for a protocol to eliminate the possibility of data collision which may occur in shared data bus. A simple protocol is developed by using an additional line referred to as Communication Bus Control Line which is connected to port C1 of each slaves. Any slave that wishes to send the final optimized chromosome will keep the line high so that the other slave would not send its chromosome at the same time. Finally, the master processor combines all the chromosomes and sends it to the WCDMA simulator.

Besides the communication protocol among the multiprocessors, there is also a need to define the USB communication protocol between the host computer and the master processor. The following explanation provides a detailed specification of the USB communication between WCDMA simulator (host computer) and master processor (PIC). Table II shows all the supported commands in this communication protocol. Note that all data are sent or received in packets of 8 bits (1 byte).

TABLE II
COMMAND DESCRIPTION

| Command	Sub Code	Group	Function
SEND_SORTED ANGLE	0xF0	W	Send all UE angles to PIC for GA process
GA_CHROMOSOME	0x0F	R	PIC returns optimized beam angles to cover all UEs
RESET	0xE0	W	Send reset command so PIC returns to initial stage

Group is classified as *W* if it is for writing and *R* when it is for reading. The WCDMA sends the UE angles through a set of packets which is classified under *SEND_SORTED ANGLE* command. The length of this packet is 48 bytes as illustrated in Fig. 2. The first byte contains the sub code which describes a particular function. The second and the third bytes represent the number of angle with UE which will be used for error checking in the master PIC. If there are UEs at 6°, 45° and 180°, the length will be 3. The rest of the packets will be used to indicate whether there is any UE at that particular angle. Take for instance, if Angle [0] is 1001 0011 it means that UE exists at 0°, 1°, 4° and 7°. For Angle [1], it will cover from angle 8-15 and so forth. Since a circle has 360° (360° = 0°), the size of Angle array is 44.
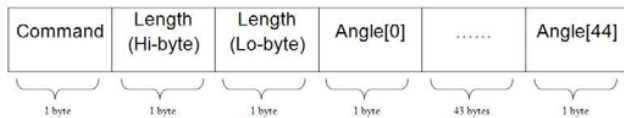
Fig. 2 *SEND_SORTED ANGLE* packets

The packet classified under *RESET* command has a length of 1 byte. It is used to initialize the master PIC in case of any unwanted interruption either internally or externally. The acknowledgement packet is also 1 byte and can come from either side to confirm reception of data.

Finally, the optimized chromosome is sent to master PIC through a set of packets which is classified under *GA_CHROMOSOME* command. Fig. 3 illustrates these packets. The first byte contains the sub code which is described in Table III. The second byte defines the total number of operating beams which can be used for error checking. If 7 beams are operating, then the length should be 7. W_{min} refers to opening beam angle and W_{max} is the closing beam angle which is displayed in Fig. 4.
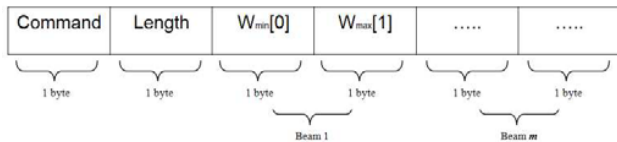
Fig. 3 *GA_CHROMOSOME* packets

m refers to the number of antenna beam in WCDMA base station. For instance, assume beam 0 has an opening angle at $20°$ and closing angle at $34°$ then $m =0$, $W_{min} = 20$ and $W_{max} = 14$. The arrow lines in Fig. 4 represents beam coverage angle.
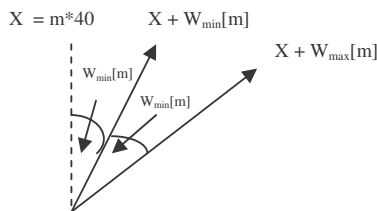
Fig. 4 W_{min} and W_{max} angles

V. IMPLEMENTATION

This parallel distributed computational microcontroller system for downlink transmitter power optimization has been proven functional. The testing of this concept was carried out in one master and two slaves to optimize the adaptive beam angles by which the downlink transmitter power is also optimized since the area of the angle openings are proportionate to the power transmitted to mobile users. The downlink transmitted power for both conventional GA and PDGA using computational microcontroller system is displayed in Table III. From the table it can be easily concluded that the parallel distributed computational microcontroller system always returns an optimal transmitted power compared to conventional GA.

TABLE III
DOWNLINK TRANSMITTED POWER (W/M^2) COMPARISON

No. of Simulation	Conventional GA	Parallel Distributed Computational System	Improvement Percentage (%)
20	0.0328109	0.026736	18.51
40	0.0360733	0.028199	21.83
60	0.0412481	0.031236	24.27

* Transmitted power is calculated from Equation 2.

VI. CONCLUSION

Built around a cooperative multiprocessor structure, the parallel distributed computational microcontroller system is a proof of concept that implements a GA which is computationally efficient for adaptive antenna downlink power transmission and having a high searching performance for adaptive beam forming in WCDMA system.

In future, the prototype design of this architecture will be embedded on a single board and have a rather compact design. It will serve as a 'plug-and-play' device where it enables easy connectivity with any desktop system or portable computers using USB port.

REFERENCES

[1] R. L. Haupt, "Phase-Only Adaptive Nulling with a Genetic Algorithm", IEEE Transactions on Antennas and Propagation, vol. 45, No. 6, June 1997. pp. 1009-1015.
[2] Y. Yashchyshyn and Piasecki M., "Improved Model of Smart Antenna Controlled by Genetic Algorithm", VI-th International Conference on The Experience of Designing and Application of CAD Systems in Microelectronics. Ukraine, 2001. pp. 147-150.
[3] S. K. Tiong, M. Ismail and A. Hassan. "Dynamic Characterized Genetic Algorithm for Adaptive Beam Forming in WCDMA System", IEEE International Conference on Communication, Nov 2005, pp.219-220.
[4] Takuma Jumonji, Goutam Chakraborty, Hiroshi Mabuchi and Masafumi Matsuhara, "A novel distributed genetic algorithm implementation with variable number of islands", Proc. IEEE Congress on Evolutionary Computation, Sept 2007, pp. 4698.
[5] Erick Cant`u-Paz, "A survey of parallel genetic algorithms", Calculateurs Paralleles, Reseaux et Systems Repartis, Vol.10, No.2, pp.141-171, 1998.
[6] M. Miki, T. Hiroyasu, M. Kaneko, K. Hatanaka, "A Parallel Genetic Algorithm with Distributed Environment Scheme", GECCO '00, pp.376-376, 2000.
[7] Erick Cant`u-Paz, David E. Goldberg, "Are Multiple Runs of Genetic Algorithms Better than One?", GECCO '02, pp.801-812, 2002.
[8] Weili Yi, Qizhen Liu and Yongbao He, "Dynamic distributed genetic algorithms", Proc. IEEE Congress on Evolutionary Computation, July 2000, pp.1132.
[9] www.microchip.com.
[10] "PIC18F4550 Datasheet", [Online]. Available: www.microchip.com.