FPGA-Based Parallel DNA Algorithm for Optimal Configurations of an Omnidirectional Mobile Service Robot Performing Fire Extinguishment

Ching-Chih Tsai, *Senior Member, IEEE*, Hsu-Chih Huang, *Member, IEEE*, and Shui-Chun Lin, *Member, IEEE*

***Abstract*—This paper presents a coarse-grain parallel deoxyribonucleic acid (PDNA) algorithm for optimal configurations of an omnidirectional mobile robot with a five-link robotic arm. This efficient coarse-grain PDNA is proposed to search for the global optimum of the redundant inverse kinematics problem with minimal movement, thereby showing better population diversity and avoiding premature convergence. Moreover, the pipelined hardware implementation, hardware/software co-design, and System-on-a-Programmable-Chip (SoPC) technology on a field-programmable gate array (FPGA) chip are employed to realize the proposed PDNA in order to significantly shorten its processing time. Simulations and experimental results are conducted to illustrate the merit and superiority of the proposed FPGA-based PDNA algorithm in comparison with conventional genetic algorithms (GAs) for omnidirectional mobile robot performing fire extinguishment.**

***Index Terms*—DNA algorithm, embedded system, field-programmable gate-array (FPGA), omnidirectional mobile robot, parallel processing.**

I. Introduction

RECENTLY, there has been an increasing interest in applying biological algorithms to solve optimization problems. Genetic algorithms (GAs) and evolutionary algorithms (EAs) have been well applied to find optimal solutions in many applications but they have a chance of converging into the local optimum rather than global optimum [1]. To circumvent this shortcoming, several researchers have considered a deoxyribonucleic acid (DNA) algorithm as another powerful optimization technique. DNA algorithm emulating the concept of the bimolecular evolution was first proposed by Aldelman [2], who showed the potential of using biomolecules for finding optimal solutions of complicated computational problems. DNA computing methods have successfully been used to solve complex problems in many disciplines [1]–[5] because they have more plentiful genetic information than conventional GAs or EAs. For example, Zhu *et al.* [3] presented a DNA algorithm of image recognition based on syntax and its application in isosceles triangle recognition, Lin *et al.* [1] proposed a self-organizing proportional-integral-derivative (PID) control design based on DNA computing method (this method presented how to solve the optimal problem more effectively), and Ding *et al.* [6] introduced the DNA genetic algorithm for design of the generalized membership-type Takagi–Sugeno fuzzy control system. However, DNA algorithms suffer from their complicated computations so that they are not suitable for real-time applications. This difficulty can be avoided using parallel processing of DNA algorithms.

Parallel architectures have been widely used to accelerate various computational algorithms by using distributed processing. They can be roughly divided into four main models as follows: 1) global (master-slave) model; 2) coarse-grain model; 3) fine-grain model; and 4) hybrid model (global plus coarse-grain) [7]. Among these models, the coarse-grain parallel model is particularly useful and pragmatic for chip-based implementation of parallel DNA algorithm, thereby speeding up DNA computing. Moreover, this kind of parallel model with System-on-a-Programmable-Chip (SoPC) implementation has better capability to solve optimal problems than software implementations do.

The SoPC technology has been bringing a major revolution in the design of integrated circuits [8]–[11] which efficiently integrate embedded processor intellectual properties (IPs) and application IPs into an FPGA chip. This technology has gained benefits of low cost, low power consumption, small circuit size, IP reusability, and reprogrammable hardware/software co-design. Owing to these advantages, the SoPC technology has been shown as a powerful means to combine flexible software modules and high-performance hardware units for implementing sophisticated signal processing algorithms, and high-performance but computation-intensive algorithms [8]–[11]. However, to date, no attempt has been made to integrating DNA algorithm, coarse-grain parallel architecture, and the SoPC technology for developing a new and efficient parallel DNA algorithm and then exploring its application to an omnidirectional mobile robot with one or two more on board manipulator(s).

Omnidirectional-wheeled mobile robots with one or two more on board manipulator(s) have been extensively used in many applications such as manufacturing, warehouse, healthcare, hazardous exploration, and so on [12]–[16]. Compared

Manuscript received August 30, 2008; revised May 30, 2009 and January 29, 2010; accepted March 24, 2010. Date of publication April 29, 2010; date of current version February 11, 2011. This work was supported by the National Science Council, Taiwan, under the Grant NSC 96-2221-E-005-106-MY2 and NSC 98-2218-E-241-003.

C.-C. Tsai is with the Department of Electrical Engineering, National Chung-Hsing University, Taichung 402, Taiwan (e-mail: cctsai@nchu.edu.tw).

H.-C. Huang is with the Department of Computer Science and Information Engineering, HungKuang University, Taichung 433, Taiwan (e-mail: hchuang@sunrise.hk.edu.tw).

S.-C. Lin is with the Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan (e-mail: lsc@ncut.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2010.2048291

with several car-like robots [17]–[20], the type of omnidirectional mobile platform has the superior agile capability to move toward any position and to simultaneously attain any desired orientation. However, in the trajectory planning of the omnidirectional mobile robot, there exists an inverse kinematics redundancy formed by the overlap in the degrees of freedom resulting from the mobile platform and the robotic arm(s). The redundant problem involves the optimal performance index which may be formed by the minimum movement or by the limits of joints, actuator speeds, and torques. For most practical fetch-and-carry applications, the trajectory planning problem for an omnidirectional mobile robot can be divided into a sequence of point-to-point tasks, which are defined as the combinations of a series of positions and orientations of an end-effector of the manipulator. The optimal trajectory problem has been investigated by several researchers [14], [21]–[26]. In particular, Kang *et al.* [25] presented a study on optimal configuration for the mobile manipulator considering minimal movement, and Kiguchi *et al.* [26] presented the DNA computing for trajectory planning of mobile platforms. As the authors' best understanding, the FPGA-based PDNA algorithm has not yet been presented and applied to accelerate the findings of the optimal configurations of the omnidirectional mobile robot for a given task.

The objectives of this paper are to develop a coarse-grain PDNA algorithm and its parallel implementation in FPGA and then to solve for the optimal point-to-point configurations, or the inverse kinematics problem, of an omnidirectional mobile robot with one arm based on the criteria of minimal joint displacement and shortest movement of the mobile service robot. The proposed coarse-grain PDNA algorithm will be exemplified by performing fire extinguishment. Overall, the contributions of the paper are threefold and are as follows:

1) A coarse-grain PDNA algorithm is proposed to increase the diversity of searching space and decrease the probability of convergence in a local optimum. This method is more likely to find the global optimum than the conventional GAs [27], [28] or parallel GAs [7] do.
2) A pipelined PDNA algorithm implementation using hardware/software co-design and SoPC technology in FPGA is employed to significantly shorten its computation time.
3) The FPGA-based PDNA algorithm is applied to efficiently solve the inverse kinematics redundant problem for an omnidirectional mobile robot performing sequential point-to-point tasks.

The rest of this paper is organized as follows. In Section II, the coarse-grain PDNA algorithm is proposed to significantly accelerate the computing and diversify the searching space. Section III elaborates the FPGA implementation of the proposed parallel DNA algorithm using the SoPC technology. Section IV elucidates the procedure of how to apply the PDNA algorithm to solve the redundant inverse kinematics problem which occurs in the point-to-point task planning for the omnidirectional mobile service robot. Section V conducts several experimental results to show the effectiveness and merit of the proposed methods. Section VI concludes this paper.

II. Parallel DNA Algorithm

This section aims to develop the efficient coarse-grain PDNA algorithm. After brief description of conventional DNA computing by including its coding scheme and DNA operators, the PDNA algorithm is proposed by using the coarse-grain parallel architecture, which takes the advantages of FPGA-based implementation.

A. DNA Computing

DNA computing is an intelligent searching strategy supported by biological evolution. DNA algorithms are very similar to GAs since their own natural genetic operators help the evolution of the genes generation by generation, such as crossover and mutation. To date, there have been four classified components of DNA computing: Adenine (A), Guanine (G), Cytosine (C), and Thymine (T). In particular, the DNA algorithms provide two new operators, namely, enzyme and virus, which are very useful to enhance the effect of mutation. Moreover, the coding schemes of DNA algorithms are quite different from those of GAs. In what follows, the coding scheme and core operators of DNA algorithms are recalled in some detail.

Coding Scheme: DNA algorithms use A, G, T, and C to stand for their chromosome; for example, one can define $\mathrm{A} = 0$, $\mathrm{G} = 1$, $\mathrm{T} = 2$, and $\mathrm{C} = 3$. Moreover, (1) can easily be applied to define the range and precision of a parameter in the DNA algorithm [1]

$$\pi = \frac{U_{\max} - U_{\min}}{4^l - 1} \tag{1}$$

where π denotes the precision, l stands for how many bits will be used, $U_{\max}$ is the maximum of the parameter, and $U_{\min}$ denotes the minimum of the parameter. Worthy of mention is that (1) hinges on the fact that DNA algorithms use four bits for the coding scheme but GAs only use two bits.

Selection: The main task of the selection module is to select individuals from the populations so that these individuals can be sent to the crossover and mutation modules in order to attain new offsprings. Selection is one of the key operators that ensure survival of the fitness. There are several selection methods with different characteristics such as roulette selection, rank selection, and tournament selection.

Crossover: Crossover is the fundamental mechanism of genetic rearrangement in DNA algorithms. This is done by the exchange of strings between two parent chromosomes from the selection module. Crossover occurs when two chromosomes break and then reconnect but to different end pieces. Although there are various crossover schemes such as one-point crossover, two-point crossover, uniform crossover, and arithmetic crossover, the one-point crossover is adopted throughout the paper due to high speed operation.

Mutation (Enzyme and Virus Operation): Mutation is the process which consists of making small alterations to the bits of the chromosomes by applying some kind of randomized changes such as single-point or multipoint mutation processes. In DNA algorithms, there are two special mutation operators, namely, enzyme and virus, which are more effective than GAs.

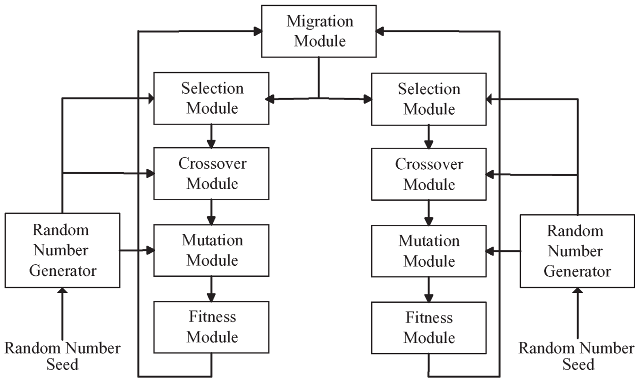


Fig. 1. Block diagram of the proposed coarse-grain PDNA algorithm using two nodes.

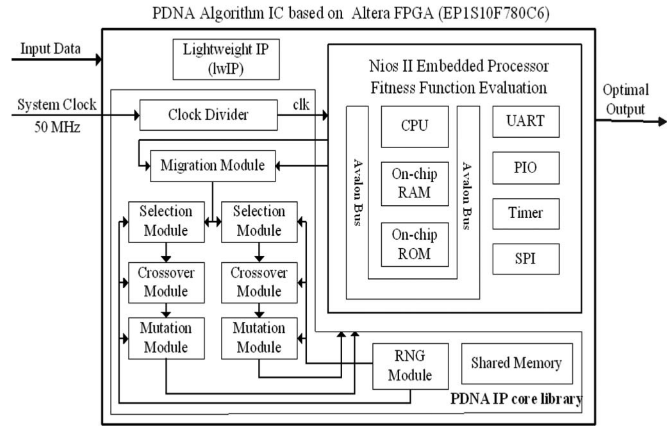


Fig. 2. FPGA-based coarse-grain PDNA algorithm.

The enzyme operator refers to deletion, in which one or more base pairs are removed, while the virus operator refers to insertion, in which one or more base pairs are inserted into the sequence. These two operators are used to enlarge or reduce the chromosome sequences in order to change the step length of searching [1], [3]–[5], thereby increase diversity of the searching space in the PDNA algorithm. However, the step length should be small.

Fitness Function: The fitness function is application-specific and is always designed according to the problem to be optimized. The fitness of new chromosomes from genetic operations such as crossover and mutation should be evaluated based on the fitness function. For complex problems, the computation time becomes dominant in the overall performance.

B. Coarse-Grain PDNA Algorithm

Although DNA algorithms have been shown to find a better optimal solution than GAs do [1]–[6], they require many computations and iterations that cause enormous time consumption. The parallel DNA algorithm has a more powerful capability to solve optimal problems much faster than conventional or serial DNA algorithms. This subsection intends to design a useful coarse-grain PDNA algorithm based on the coarse-grain model [7]. Fig. 1 depicts the architecture of the proposed coarse-grain PDNA algorithm. Such a PDNA algorithm is composed of two DNA algorithms which are concurrently executed on two processors with data interchange. Compared with conventional DNA algorithms, this PDNA algorithm gains benefits of distributed computations in which more searching space can be explored and much faster computing speed is employed to find an optimal solution. Hence, this proposed algorithm will significantly increase its possibility to seek for a global optimum and accelerate its computation.

The parallel DNA algorithm has been implemented on two nodes, each of which has its own DNA algorithm. Although the two nodes perform the DNA operation independently and separately, a migration operator is designed to allow them to exchange their best members. The searching spaces for both nodes are independent, and the populations on the nodes evolve separately for a certain number of generations, called isolation time. After the isolation time, a number of the best subpopulations (migration rate) from the two nodes are exchanged via the migration operator. Accordingly, the migration operator effectively increases the diversity among the individuals in each sub-population, and significantly decreases the probability of parallel DNA algorithm to get stuck in a local optimum.

III. FPGA Implementation of the Proposed PDNA Algorithm

The software implementation of PDNA algorithm for applying large searching spaces may cause unacceptable delays due to enormous computations and iterations. An alternative to this approach is the hardware implementation of the PDNA algorithm in order to achieve tremendous speedup over software implementation. This subsection is dedicated to design the PDNA algorithm in FPGA for solving optimal problems. Fig. 2 depicts the architecture of the Altera FPGA implementation for the proposed PDNA algorithm using two nodes. The fitness module for the PDNA algorithm has been implemented into the 32-bit Nios II processor whose numerical precision and computation speed are high enough to realize the fitness function. The user IP cores (custom logic) for this PDNA algorithm have been developed by VHDL (VHSIC Hardware Description Language), including the random number generator (RNG) module, selection module, crossover module, mutation module, and migration module. The software-based fitness module and hardware-based custom logics for PDNA algorithm are connected to the system-interconnected fabric via Avalon-MM in one FPGA chip. The used FPGA chip is the Altera Stratix EP1S10F780C6 with 10 570 Logic Elements (LEs), 426 user I/O pins, six DSP blocks, 920 448 RAM bits memory, six Phase-Lock Loops (PLLs), and an embedded Nios II 32-bit Reduced Instruction Set Computer (RISC) processor.

In what follows, special efforts will be paid to design several modules in the proposed FPGA-based PDNA algorithm shown in Fig. 2 such as RNG module, selection module, crossover module, mutation module, fitness module, and migration module. The software-based fitness module running in the embedded processor has the flexibility for realizing different fitness functions, whereas the VHDL hardware-based modules in FPGA exploit the features of pipelining and parallelization so that they can be further synthesized to application-specific integrated circuits (ASICs). This hardware/software co-design

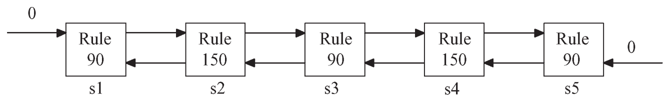


Fig. 3. Example of the cellular automata RNG.

technique is more efficient than pure hardware-oriented FPGA designs in [19], [27]. Moreover, with the SoPC technology, a real-time operating system (RTOS) can be ported into the same FPGA chip for embedded applications such as many kinds of network service. Worthy of mention is that the reusable PDNA IP core library is very useful not only for the proposed PDNA algorithm but also for other embedded genetic algorithms.

A. RNG Module

This pseudorandom number generator (PRNG) is a crucial module required in the proposed parallel DNA algorithm. The module generates a sequence of pseudorandom bits for executing the crossover module, mutation module, and selection module. Although there are two methods [linear feedback shift register (LFSR) and cellular automata (CA)] commonly used for the RNG module, the CA method has been shown to generate better random numbers than the LFSR method does [30]. Thus, the CA method is adopted to produce desired pseudorandom numbers by using several alternating cells which change their states based on the two rules: rule 90 and rule 150, given by

$$\text{Rule 90}: s_i^{+} = s_{i-1} \oplus s_{i+1}$$
$$\text{Rules 150}: s_i^{+} = s_{i-1} \oplus s_i \oplus s_{i+1}$$

where s_i denotes the current state of cell i; s_i^{+} denotes the next state for s_i; $\oplus$ is the exclusive OR operator. Fig. 3 shows the example of CA-based PRNG.

For efficient hardware realization, the VHDL language is adopted to implement this CA module instead of the software-based RNG [30]. The main advantage of hardware-based cellular automata RNG hinges on its inherent speed. Once the seed value is given, the VHDL-based 16-bit RNG efficiently generates a random sequence by the cell outputs on every system clock cycle (0.02 μs).

B. Selection Module

The aim of the selection module is to ensure survival of the fittest chromosomes (parents) to create new offspring. In the proposed PDNA algorithm, the selected two chromosomes are sent to the crossover module and then the mutation module in the pipelined parallel architecture in Fig. 1. Roulette selection is the most popular selection scheme, however, it is required to sum up the fitness values for all the genes and to sort all the genes in the current population. Hence, the tournament selection method is adopted in this module by taking the advantages of both chip size and computing speed to converge into the optimum as fast as possible.

The procedure of the selection module in this proposed PDNA algorithm is similar to the conventional GA selection modules. In comparison to the conventional GAs, the hardware-based selection module is particularly designed to achieve the selection task. By taking the advantage of hardware implementation, the 16-bit efficient comparators are employed to select the best chromosomes using VHDL combinational logic in the FPGA.

C. Crossover Module

The crossover module is a mechanism of genetic information exchange used in the proposed PDNA algorithm. Once the selection process is completed in Section III-B, the two selected chromosomes are passed to this module. The crossover module generates two new offspring; this is done by exchanging the strings of the two parent chromosomes. The high-speed one-point crossover scheme is chosen in this PDNA algorithm because two offspring are obtained after the crossover process. The one-point crossover module has been implemented in combinational logic VHDL code to perform the high-speed crossover process more efficiently than other crossover schemes do. Note that the random single crossover point reported in this module comes from the RNG module in Section III-A. Once the crossover position is randomly determined, the genetic information of the two-parent chromosomes is directly exchanged to perform one-point crossover and the altered chromosomes are then sent to the mutation module.

D. Mutation Module (Enzyme and Virus Operation)

Similar to the mutation module in GAs, the mutation position is randomly determined, however, the chromosome lengths will be changed after this special module in PDNA algorithm. These special mutation operators, including enzyme and virus, have been implemented using combinational logic VHDL codes. After receiving two offspring from the crossover process, the mutation module starts to proceed with the mutation process, which is quite different from that in genetic algorithms because the enzyme and virus operators are used to alter the chromosome lengths. The enzyme and virus operations can be coded by 2 bits, and their inserted bit and deleted bit positions are randomly generated from the PRNG module in Section III-A. Worthy of mention is that a low mutation rate is preferred so that the two offspring will resemble their parents.

E. Fitness Module

As mentioned in Section II-A, the fitness function can be defined for its corresponding optimal problem. The purpose of this fitness module is to evaluate the population members after mutation and insert them into a new population. Because the fitness evaluation is usually problem specific and user defined by different applications, the fitness module has been efficiently implemented by software running in the embedded processor Altera Nios II. With the software-based fitness module, different fitness functions can be implemented with the predesigned software program. Furthermore, the Nios II C-to-hardware acceleration (C2H) compiler is employed to create custom hardware accelerators directly from ANSI C fitness

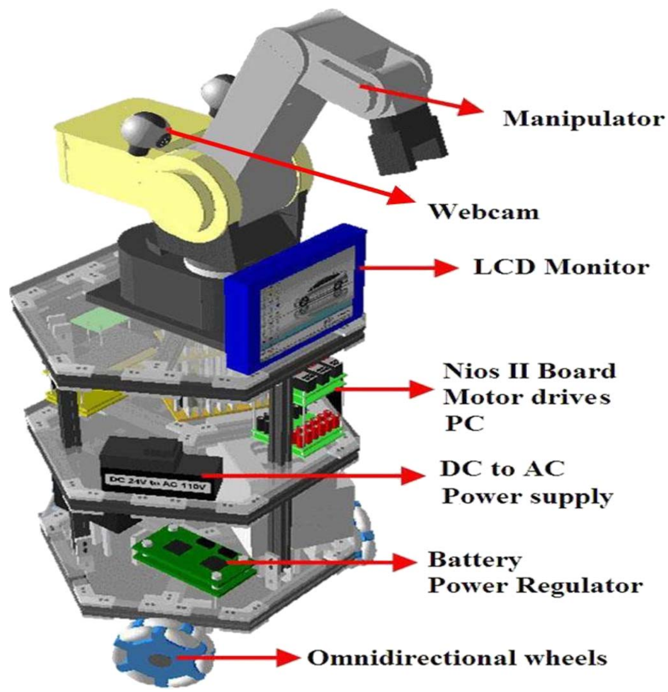


Fig. 4. Physical structure of the omnidirectional mobile robot.

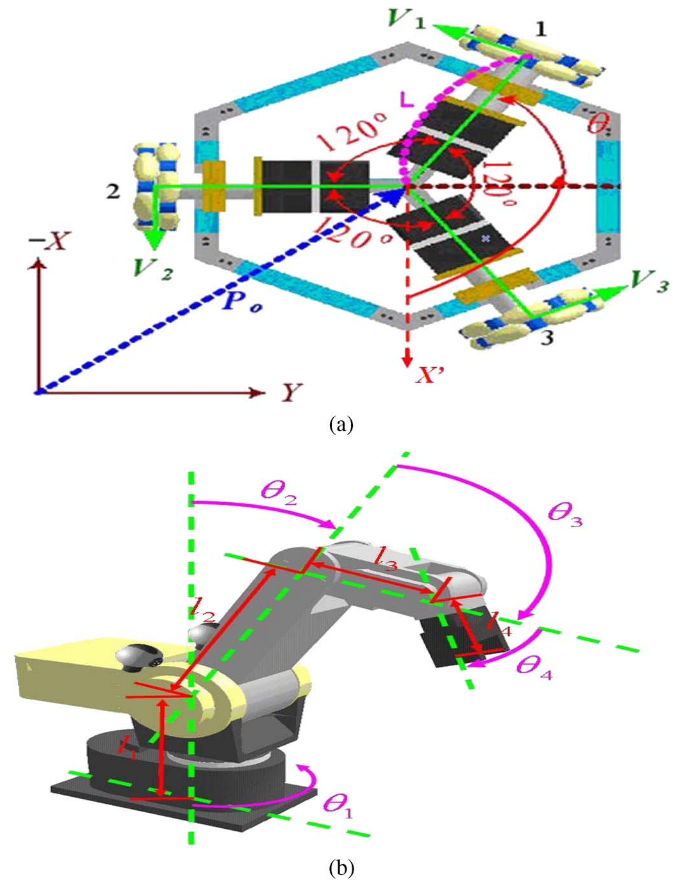


Fig. 5. (a) Geometry of the omnidirectional mobile platform. (b) Geometry of the manipulator.

module source code, thus significantly improving execution performance of the software-based fitness module.

F. Migration Module

In the proposed coarse-grain PDNA algorithm, there are two separate DNA engines working concurrently. Both DNA engines communicate with each other via the migration module; in particular, the migration module is designed to allow the two separate DNA engines to exchange their two best chromosomes after the isolation time. This module was implemented by VHDL to take the advantages of hardware realization. In this migration module, a VHDL-based timer is used to deal with data exchange from the two DNA engines. This hardware timer was implemented in FPGA by using a simple digital counter with a clock rate of 50 MHz. Once the isolation time is up, the data exchange task is triggered and the best populations of the two separate DNA engines are directly exchanged using a simple combinational logic.

IV. Application to Redundant Inverse Kinematics Problem

A. Redundant Inverse Kinematics Problem

As shown in Fig. 4, the omnidirectional mobile service robot is equipped with one three-wheeled-omnidirectional mobile platform and one five-link robot arm. This kind of service robot is designed to accomplish more complex tasks compared with the platform-only [12]–[14], [19], [20] or arm-only robots [21]–[25], [29]. However, there exists a well-known redundant problem for the service robot due to its more degrees of freedom (DOFs).

In this kind of mobile service robot, there are eight DOFs (three DOFs for the mobile platform and five DOFs for the robot arm). The PDNA algorithm will be applied to find the optimal solution for the mobile platform and robot arm of the mobile service robot.

In order to perform an assigned task, it is necessary to find a final posture for the mobile service robot. However, there exist infinitely many postures for the platform and the robotic arm from any starting pose to any destination pose. Therefore, the PDNA algorithm will be used to minimize the whole movement of the robot in the trajectory planning problem. The configuration of the mobile service robot can be represented by the following vector matrix:

$$q = [q_v \quad q_m]^T \tag{2}$$

where $q_v = [x_v \quad y_v \quad \theta_v]^T$ for the omnidirectional mobile platform and $q_m = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4]^T$ for the robot arm in Fig. 5. The vector q represents both the position information of the mobile vehicle including the position of center of mass as well as the orientation angle and the angle configurations composed by each joint of the manipulator. Note that only $\theta_1 \sim \theta_4$ are defined in (2) because the fifth axis is not used in this study. To determine the minimal movement of the mobile robot, one defines a fitness function composed of the differences from the initial states of mobile robot and the final states of the mobile robot.

The initial states q_i of the mobile robot are denoted by

$$\begin{aligned} q_i &= [q_{v,i} \quad q_{m,i}]^T \\ &= [x_{v,i} \quad y_{v,i} \quad \theta_{v,i} \quad \theta_{1,i} \quad \theta_{2,i} \quad \theta_{3,i} \quad \theta_{4,i}]^T. \end{aligned} \tag{3}$$

The final states of the mobile robot are expressed by

$$\begin{aligned} q_f &= [q_{v,f} \quad q_{m,f}]^T \\ &= [x_{v,f} \quad y_{v,f} \quad \theta_{v,f} \quad \theta_{1,f} \quad \theta_{2,f} \quad \theta_{3,f} \quad \theta_{4,f}]^T. \end{aligned} \tag{4}$$

The desired final position of the end-effector is represented as $X_{m,f} = [x_{m,f} \quad y_{m,f} \quad z_{m,f}]^T$. Based on the desired final position of the end-effector and the inverse kinematics of the mobile robot, one can obtain the final position of the mobile platform and all the joint angles of the manipulator. By considering the minimal movement, one defines the cost function L' as

$$\begin{aligned} L' &= (q_f - q_i)^T (q_f - q_i) \\ &= (q_{v,f} - q_{v,i})^T (q_{v,f} - q_{v,i}) + (q_{m,f} - q_{m,i})^T (q_{m,f} - q_{m,i}). \end{aligned} \tag{5}$$

The position of the end-effector in the moving frame can be now represented in the world frame. The kinematics equation of the mobile manipulator and the rotation matrix are available to convert the position of end-effector in the mobile vehicle frame into that in the world frame

$$X_{m,f} = X_{v,f} + R(\theta_{v,f})K(q_{m,f}) + D \tag{6}$$

where $X_{v,f}$ represents the final position of the mobile platform in the world frame, $R(\theta_{v,f})$ represents the rotation matrix from the moving frame to the world frame, and $K(q_{m,f})$ stands for the kinematics equation of the manipulator configuration. In addition, for the case in which the manipulator is not put on the center of the omnidirectional mobile robot, the notation D represents the modified distance. The rotation matrix and modified distance are given in the following:

$$R(\theta_{v,f}) = \begin{bmatrix} \cos\theta_{v,f} & -\sin\theta_{v,f} & 0 \\ \sin\theta_{v,f} & \cos\theta_{v,f} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathrm{D} = \begin{bmatrix} d \cdot \cos\theta_b \\ d \cdot \sin\theta_b \\ 0 \end{bmatrix} \tag{7}$$

where $\theta_{v,f}$ is the final heading angle of the mobile vehicle with respect to the x-axis and $\theta_b = \theta_v + 60^\circ$. The kinematics equation of the manipulator is given by the following matrix:

$$K(q_{m,f}) = \begin{bmatrix} \cos(\theta_1)\,[l_2 \sin(\theta_2) + l_3 \sin(\theta_2+\theta_3) + l_4 \sin(\theta_2+\theta_3+\theta_4)] \\ \sin(\theta_1)\,[l_2 \sin(\theta_2) + l_3 \sin(\theta_2+\theta_3) + l_4 \sin(\theta_2+\theta_3+\theta_4)] \\ l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2+\theta_3) + l_4 \cos(\theta_2+\theta_3+\theta_4) \end{bmatrix} \tag{8}$$

where l_1, l_2, l_3, and l_4 are respectively the lengths of the manipulator axes, and θ_1, θ_2, θ_3 and θ_4 are respectively the joint angles of the links of the manipulator. Fig. 6 shows the overall structure of the mobile robot system. Moreover, from (6), one obtains

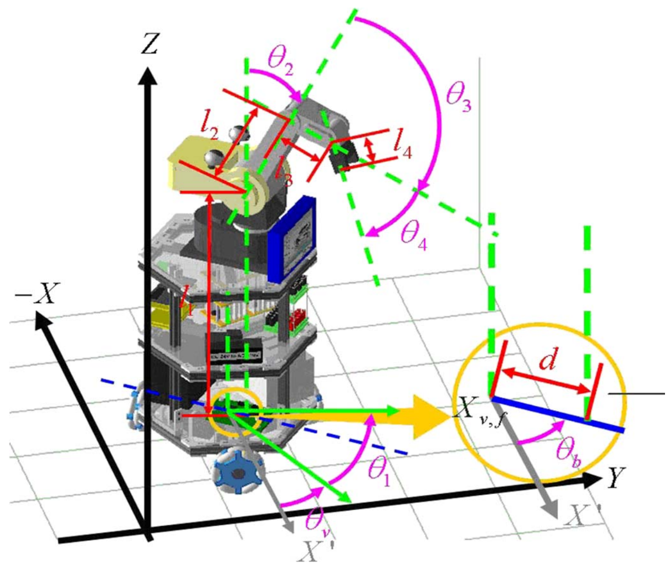


Fig. 6. Parameters definitions for the mobile service robot.

$$X_{v,f} = X_{m,f} - R(\theta_{v,f})K(q_{m,f}) - D. \tag{9}$$

Substituting (7) and (8) into (9) gives

$$\begin{bmatrix} x_{v,f} \\ y_{v,f} \\ z_{v,f} \end{bmatrix} = \begin{bmatrix} x_{m,f} \\ y_{m,f} \\ z_{m,f} \end{bmatrix} - \begin{bmatrix} d \cdot \cos\theta_b \\ d \cdot \sin\theta_b \\ 0 \end{bmatrix} - \begin{bmatrix} \cos\theta_{v,f} & -\sin\theta_{v,f} & 0 \\ \sin\theta_{v,f} & \cos\theta_{v,f} & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta_1)\,[l_2 \sin(\theta_2) + l_3 \sin(\theta_2+\theta_3) + l_4 \sin(\theta_2+\theta_3+\theta_4)] \\ \sin(\theta_1)\,[l_2 \sin(\theta_2) + l_3 \sin(\theta_2+\theta_3) + l_4 \sin(\theta_2+\theta_3+\theta_4)] \\ l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2+\theta_3) + l_4 \cos(\theta_2+\theta_3+\theta_4) \end{bmatrix}. \tag{10}$$

The selection of cost function for solving the redundant inverse kinematics problem is based on the minimum of weighted squared errors of the manipulator and the mobile platform

$$\begin{aligned} F_{fitness} = {} & \omega \left[(x_{v,f} - x_{v,i})^2 + (y_{v,f} - y_{v,i})^2 + (\theta_{v,f} - \theta_{v,i})^2 \right] \\ & + (\theta_{1,f} - \theta_{1,i})^2 + (\theta_{2,f} - \theta_{2,i})^2 + (\theta_{3,f} - \theta_{3,i})^2 + (\theta_{4,f} - \theta_{4,i})^2 \end{aligned} \tag{11}$$

where ω is the weighting factor which is larger than one; the parameter ω is given in (11) because the mobile platform is much heavier than the arm. Thus, the fitness function to be minimized can be reformulated as

$$F = L_{\max} - F_{fitness} \tag{12}$$

where $L_{\max}$ is any positive value and is always larger than the maximum value of the fitness function. The proposed PDNA algorithm is then adopted to find the maximum value of the function F in (12), namely that the best fitness value is thus obtained and the optimal configuration is determined. These optimal parameters, including x_v, y_v, θ_v, θ_1, θ_2, θ_3, and θ_4 are required for robot arm controller and platform controller.

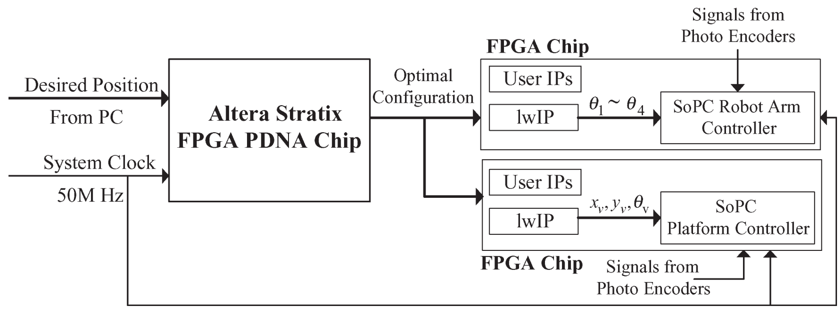


Fig. 7. FPGA-based PDNA algorithm for the omnidirectional mobile service robot.

B. FPGA-Based PDNA Algorithm for Point-to-Point Task Planning

This section is aimed to apply the FPGA-based PDNA algorithm for the point-to-point trajectory planning problem of the mobile service robot. The PDNA algorithm for solving the redundant problem with the fitness function in (12) is described by the following steps.

1) Use the CA-based PRNG to randomly generate the parameters θ_{v}, θ_1, θ_2, θ_3, and θ_4 to be chromosomes and make sure that all of them fit with the basic requirements.
2) Set the two parents from the VHDL-based tournament selection.
3) Execute the procedure of crossover and also check whether new chromosomes are acceptable. If the new chromosomes do not satisfy the requirement, this procedure must be repeated until acceptable chromosomes are obtained.
4) Perform the mutation process (enzyme and virus operation) with low mutation rate and ensure that new chromosomes are reasonable.
5) Repeat these four steps again until the specified condition is reached.

Fig. 7 presents the FPGA-based PDNA algorithm for optimal configurations of an omnidirectional mobile service robot. The FPGA PDNA chip consists of PDNA IP core library and system IPs. The Nios II embedded processor IP was used to evaluate the fitness function (12) and the PDNA IP core library, including RNG, selection, crossover, and mutation, was presented to perform the PDNA operators. A 50 MHz quartz oscillator is used to supply the PDNA FPGA chip in Altera Stratix development board. The VHDL-based clock divider and PLL system IP are employed to generate the desired clock frequency. With the input desired position of the mobile robot's end-effector, the proposed FPGA-based PDNA algorithm seeks for the optimal parameters, x_v, y_v, θ_{v}, θ_1, θ_2, θ_3, and θ_4. These software and hardware components were all integrated in one FPGA chip performing the PDNA algorithm more quickly and efficiently. These optimal parameters, x_v, y_v, θ_{v}, θ_1, θ_2, θ_3, and θ_4, from the PDNA chip were sent to the SoPC-based manipulator controller and SoPC-based platform controller, respectively, for performing motion feedback control of a given task.

Furthermore, the real-time OS MicroC/OS-II was ported into the FPGA PDNA chip to deal with the data communication via TCP/IP protocol with the embedded platform and manipulator control systems. The TCP/IP stack was constructed using the prevalent real-time OS MicroC/OS-II which is regarded as a portable, ROMable, scalable, preemptive real-time, and multitasking kernel for microprocessors. This kind of stack also includes the standard UNIX sockets application programming interface (API). Moreover, the embedded softcore Nios II processor works with the lightweight IP (lwIP) for the Ethernet connectivity, thereby significantly reducing resource usage. The resource usage of the proposed PDNA algorithm is 9 014 LEs (85% of total LEs), 690 152 memory bits (75% of total RAM bits).

V. Experimental Results and Discussion

A. Fitness Value in PDNA Computing

This section aims at constructing experiments to examine the feasibility and efficacy of the proposed PDNA computing for a mobile service robot with the fitness function (12). In the experiments, the proposed FPGA-based PDNA algorithm worked with the embedded platform and manipulator controllers to achieve the fire extinguishment. The weighting factor is given by $\omega = 1.5$ and the five joint angles θ_{v}, θ_1, θ_2, θ_3, and θ_4 are limited by $0^\circ \leq \theta_{\mathrm{v}} \leq 360^\circ$, $-90^\circ \leq \theta_1 \leq 210^\circ$, $-10^\circ \leq \theta_2 \leq 120^\circ$, $0^\circ \leq \theta_3 \leq 110^\circ$, and $-90^\circ \leq \theta_4 \leq 90^\circ$. The initial position of the end-effector was set by (22 cm, 39 cm, 102 cm) and all joints angles were given by $\theta_{1,i} = 45^\circ$, $\theta_{2,i} = 30^\circ$, $\theta_{3,i} = 60^\circ$, $\theta_{4,i} = 60^\circ$, and $\theta_{v,i} = 0^\circ$. The lengths of all links are represented as follows: $l_1 = 95$ cm, $l_2 = 26.5$ cm, $l_3 = 16.5$ cm, and $l_4 = 16.7$ cm. In addition, the modified distance was $d = 7$ cm and $L_{\max} = 100\,000$. The desired final position of the end-effector was (100 cm, 100 cm, 110 cm). The PDNA algorithm was terminated at fifth second. The optimal solution was found by

$$(x_{v,f}, y_{v,f}, \theta_{1,f}, \theta_{2,f}, \theta_{3,f}, \theta_{4,f}, \theta_{v,f}) = (65\text{ cm}, 55\text{ cm}, 41^\circ, 43^\circ, 43^\circ, 23^\circ, 8^\circ).$$

Substituting this solution into (10) obtains the end-effector's position $X_{m,f} = [100.63\text{ cm}, 99.47\text{ cm}, 110.09\text{ cm}]$, whose error is [$-0.63$ cm, 0.53 cm, -0.09 cm] which is very close to the exact destination position. On the other hand, under the same experimental setting, the SoPC parallel GA (PGA) [7] obtained the solution for mobile robot $(x_{v,f}, y_{v,f}, \theta_{1,f}, \theta_{2,f}, \theta_{3,f}, \theta_{4,f}, \theta_{v,f}) = (67\text{ cm}, 75\text{ cm}, 15^\circ, 15^\circ, 69^\circ, 51^\circ, 15^\circ)$ and the resultant position of the end-effector is given by $X_{m,f} = [99.20\text{ cm}, 99.30\text{ cm}, 110.51\text{ cm}]$, whose error is

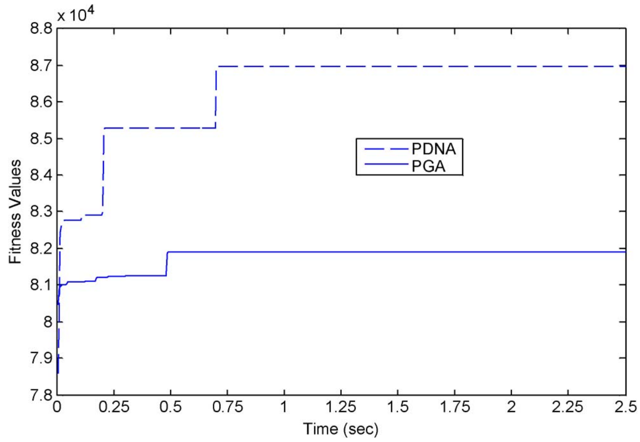


Fig. 8. Fitness values in the PDNA algorithm and parallel GA.

TABLE I
COMPARISON OF RESOURCE USAGE AND EXECUTION TIME OF THE SOFTWARE-BASED AND SOPC-BASED ALGORITHMS

	Software GA	SoPC GA [7]	SoPC Parallel GA [7]	Software DNA	Coarse-grain PDNA
Resource usage (LEs)		5357	8357		9014
Execution time (sec)	42.55	1.5	0.73	69.22	0.76

[0.8 cm, 0.7 cm, −0.51 cm]. These experimental results clearly indicate that the proposed PDNA algorithm is capable of finding a better solution than the PGA algorithm does.

Fig. 8 presents the time histories of variations of fitness values for PDNA algorithm and parallel GA; the fitness values are 86 964 and 81 898 for the PDNA algorithm and parallel GA, respectively. Compared with the parallel GA [7], the fitness value in PDNA algorithm is clearly higher than that in parallel GA. These results indicate that the proposed FPGA-based PDNA algorithm outperforms the SoPC parallel GA.

B. Performance Evaluation of the FPGA-Based PDNA Algorithm

The aim of the second experiment is to examine the merit and performance of the proposed coarse-grain PDNA algorithm. The software fitness module implemented by the softcore processor Nios II works with the PDNA IP library in FPGA to find the optimal solution. This pipelined and parallel strategies significantly diverse the searching space and shorten the execution time.

To compare with the conventional methods to solve the redundant problem in mobile service robot, Table I presents the detailed analyses for resource usage (LEs) and average execution time of five different implementations in which the chromosome length is 16 bits for each parameter. These algorithms were executed 20 times with distinct random seeds to report the analyses. Software GA and DNA algorithm Matlab codes were running on a PC with Pentium D 3.4 GHz CPU. The population size is 256, the crossover probability is 0.45, and the mutation probability is 0.1. The fitness module in SoPC GA and coarse-grain PDNA algorithm were realized in software running on the softcore embedded processor (Nios II) which evaluates the fitness of each chromosome and passes the randomly generated chromosomes with two other chromosomes (parents) to the selection module. The selection unit compared these chromosomes and obtained two better chromosomes passed to the next generation. The software GA, software DNA algorithm, and SoPC GA were terminated in 500th generations. The proposed coarse-grain PDNA algorithm and SoPC parallel GA [7] terminated at the approximate fitness value for software DNA algorithm and software GA, respectively. With efficient FPGA implementation, the proposed coarse-grain PDNA algorithm found the optimal configuration and shortened the computation time which is 90 times of its corresponding software DNA computing. Compared with the SoPC parallel GA [7], the proposed FPGA-based coarse-grain PDNA algorithm obtained a better optimal solution (with higher fitness value) by using slightly more LEs and execution time. Worthy of mention is that as shown in Table I, the data regarding the software-based GA and the software-based DNA algorithms were obtained from computer simulations, whereas the data of the SoPC-based GA, the SoPC-based parallel GA, and the SoPC-based PDNA algorithms were taken from real experiments.

C. Brief Description of the Experimental Omnidirectional Mobile Service Robot

The aim of the following experiment is to apply the proposed FPGA-based coarse-grain PDNA algorithm to construct a mobile service robot incorporating with the autonomous mobile platform and an on board manipulator to perform the fire extinguishment task. Fig. 9 depicts the block diagram of the experimental omnidirectional mobile service robot which has three subsystems: vision system, SoPC embedded one-arm control system, and SoPC embedded mobile platform control system.

To connect these three subsystems, the experimental setup constructed the networked embedded system using the embedded processors, RTOS and lwIP. The vision system is composed of one compact PC and one FPGA-based PDNA embedded system. With the compact PC and CCD cameras, the true position of the desired object can be obtained by executing the image processing algorithms. The position information was fed back to the FPGA-based coarse-grain PDNA algorithm for resolving redundant problem to find the optimal configuration of the mobile platform and the robotic arm. Ethernet network environment was constructed between the PC and PDNA embedded system for data communication. Fig. 10 depicts the recent picture of the experimental omnidirectional mobile service robot.

The optimal configuration information for the whole service robot was sent to the two embedded systems to control the mobile platform and the manipulator via the TCP/IP protocol. The two embedded systems perform conventional Proportional Integral (PI) motion control for the servo motors. All the hardware and software design of the omnidirectional mobile service robot were integrated into the FPGA chips. The proposed

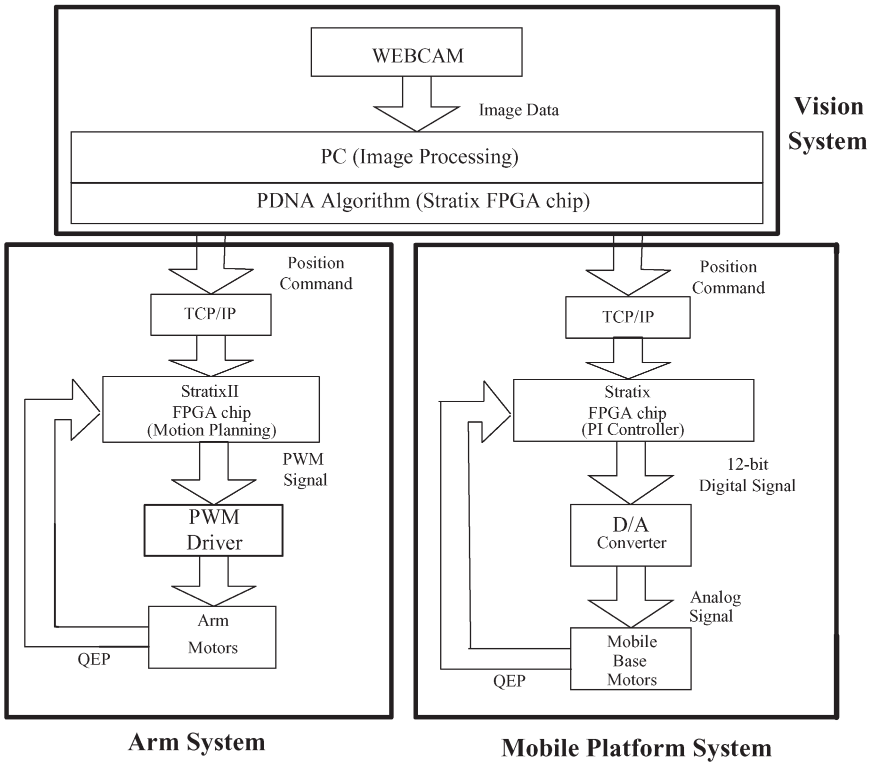


Fig. 9. Block diagram of the experimental omnidirectional mobile service robot.

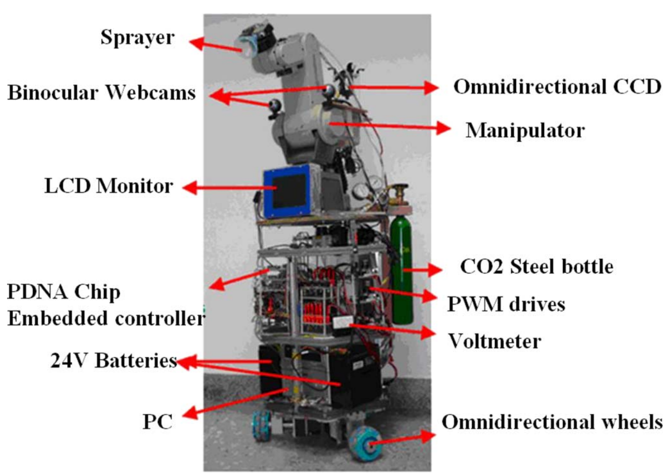


Fig. 10. Picture of the experimental omnidirectional mobile service robot.

coarse-grain PDNA algorithm was implemented in FPGA chip which integrates the embedded processor, RTOS, lwIP, and VHDL-based IP circuits (PDNA IP core library) for solving the redundant problem.

D. Fire Extinguishment Task Execution

This subsection is concerned with how the SoPC embedded mobile service robot with the proposed FPGA-based coarse-grain PDNA algorithm performs a fire extinguishment task in the following manner. In the beginning, the dual CCD cameras of the service robot detected the fire and calculated its 3-D position through image processing [31]. The FPGA-based coarse-grain PDNA algorithm was employed to solve the redundant problem of the mobile platform and the manipulator. The optimal information for the mobile platform and manipulator was then sent to the embedded platform controller and the embedded manipulator controller via Ethernet network.

Once the optimal configuration is obtained, the trapezoid velocity profile [24] with constant acceleration and deceleration was employed to drive the motors by considering the maximum velocity, maximum acceleration, and deceleration for each motor, thus steering the mobile service robot toward the desired posture smoothly. Afterwards, the robotic arm executed the motion profile to bring the CO2 sprayer to the desired position. Then, the electric switch was turned on and CO2 was sprayed on the fire to achieve the fire extinguishment task. Fig. 11 shows the experimental results of the fire extinguishment task, where the sequential still images were captured from the real-time fire extinguishment experimental video. The results show that the experimental mobile service robot with the proposed FPGA-based coarse-grain PDNA algorithm and two SoPC-based controllers is capable of successfully achieving the fire extinguishment task.

VI. Conclusion

This paper has presented a coarse-grain parallel DNA algorithm for optimal configurations of an omnidirectional mobile service robot performing fire extinguishment task. The proposed coarse-grain PDNA algorithm has been efficiently implemented into a FPGA chip using the hardware/software co-design technique and SoPC technique to solve the redundant problem. The coarse-grain PDNA algorithm has been rapidly developed in the FPGA chip by incorporating with the embedded processor and the RTOS in the same chip. Through

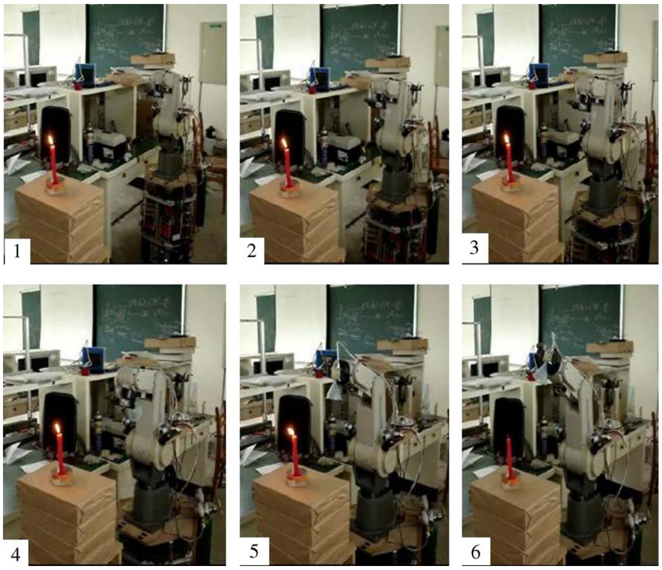


Fig. 11. Sequential still images captured from the experimental video for fire extinguishment task.

experimental results, the proposed FPGA-based PDNA algorithm outperforms conventional software-based DNA algorithm, GAs, or parallel GAs. Last but not least, this FPGA-based coarse-grain PDNA algorithm has successfully been applied to the mobile service robot to perform a fire extinguishment task.

Although the proposed PDNA algorithm has only been applied in the paper to optimal configuration of an omnidirectional mobile robot, this PDNA algorithm can be easily applied to other optimal problems which have been addressed by conventional GA and DNA algorithms and obtain better results. For example, the problems can be the optimal controller design [1], [6], image recognition [2], graph problems [3], trajectory problem [26], [32], navigation problem [29], and so on. Due to page limit, no further comparison tests in different testing scenarios are provided but the proposed method is believed to obtain better solutions for these problems.

REFERENCES

[1] C. L. Lin, H. Y. Jan, and T. H. Huang, "Self-organizing PID control design based on DNA computing method," in *Proc. IEEE Int. Conf. Control Appl.*, 2004, pp. 568–573.

[2] L. M. Adeleman, "Molecular computing of solutions to combinational problems," *Science*, vol. 266, no. 5187, pp. 1021–1024, Nov. 1994.

[3] Y. Zhu, Y. Ding, W. Li, and L. A. Zadeh, "DNA algorithm of image recognition and its application," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, 2006, pp. 375–379.

[4] X. Liu and Y. Li, "Efficient DNA algorithms for chromatic number of graph problems," in *Proc. IEEE Int. Conf. Autom. Logistics*, 2007, pp. 450–454.

[5] M. H. Garzon and R. J. Deaton, "Biomolecular computing and programming," *IEEE Trans. Evol. Comput.*, vol. 3, no. 3, pp. 236–250, Sep. 1999.

[6] Y. Ding and L. Ren, "DNA genetic algorithm for design of the generalized membership-type Takagi-Sugeno fuzzy control system," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2000, pp. 3862–3867.

[7] M. S. Jelodar, M. Kamal, S. M. Fakhraie, and M. N. Ahmadabadi, "SOPC-based parallel genetic algorithm," in *Proc. IEEE Congr. Evol. Comput.*, 2006, pp. 2800–2806.

[8] S. S. Solano, A. J. Cabrera, I. Baturone, F. J. Moreno-Velo, and M. Brox, "FPGA implementation of embedded fuzzy controllers for robotic applications," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1937–1945, Aug. 2007.

[9] Y. F. Chan, M. Moallem, and W. Wang, "Design and implementation of modular FPGA-based PID controllers," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1898–1906, Aug. 2007.

[10] D. Zhang and H. Li, "A stochastic-based FPGA controller for an induction motor drive with integrated neural network algorithms," *IEEE Trans. Ind. Electron.*, vol. 55, no. 2, pp. 551–561, Feb. 2008.

[11] H. C. Huang and C. C. Tsai, "FPGA implementation of an embedded robust adaptive controller for autonomous omnidirectional mobile platform," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1604–1616, May 2009.

[12] R. L. Williams, II, B. E. Carter, P. Gallina, and G. Rosati, "Dynamic model with slip for wheeled omnidirectional robots," *IEEE Trans. Robot. Autom.*, vol. 18, no. 3, pp. 285–293, Jun. 2002.

[13] D. K. Chwa, "Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 4, pp. 637–644, Jul. 2004.

[14] T. Kalmár-Nagy, R. D'Andrea, and P. Ganguly, "Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle," *Robot. Auton. Syst.*, vol. 46, no. 1, pp. 47–64, Jan. 2004.

[15] H. C. Huang and C. C. Tsai, "Simultaneous tracking and stabilization of an omnidirectional mobile robot in polar coordinates: A unified control approach," *Robotica*, vol. 27, no. 3, pp. 447–458, 2009.

[16] H. C. Huang and C. C. Tsai, "Adaptive robust control of an omnidirectional mobile platform for autonomous service robots in polar coordinates," *J. Intell. Robot. Syst.*, vol. 51, no. 4, pp. 439–460, Apr. 2008.

[17] Z. P. Jiang and H. Nijmeijer, "A recursive technique for tracking control of nonholonomic systems in chained form," *IEEE Trans. Autom. Control*, vol. 44, no. 2, pp. 265–279, Feb. 1999.

[18] T. C. Lee, K. T. Song, C. H. Lee, and C. C. Teng, "Tracking control of unicycle-modeled mobile robots using a saturation feedback controller," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 2, pp. 305–318, Mar. 2001.

[19] T. H. Li, S. J. Chang, and Y. X. Chen, "Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot," *IEEE Trans. Ind. Electron.*, vol. 50, no. 5, pp. 867–880, Oct. 2003.

[20] K. D. Do, Z. P. Jiang, and J. Pan, "Simultaneous tracking and stabilization of mobile robots: An adaptive approach," *IEEE Trans. Autom. Control*, vol. 49, no. 7, pp. 1147–1151, Jul. 2004.

[21] B. E. Bishop, "On the use of redundant manipulator techniques for control of platoons of cooperating robotic vehicles," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 5, pp. 608–615, Sep. 2003.

[22] J. P. Puga and L. E. Chiang, "Optimal trajectory planning for a redundant mobile manipulator with non-holonomic constraints performing push–pull tasks," *Robotica*, vol. 26, no. 3, pp. 385–394, May 2008.

[23] O. Chocron, "Evolutionary design of modular robotic arms," *Robotica*, vol. 26, no. 3, pp. 323–330, May 2008.

[24] Y. S. Kung and G. S. Shu, "Design and implementation of a control IC for vertical articulated robot arm using SOPC technology," in *Proc. IEEE Int. Conf. Mechatronics*, 2005, pp. 532–536.

[25] J. G. Kang and J. M. Lee, "A study on optimal configuration for the mobile manipulator considering the minimal movement," in *Proc. IEEE Int. Symp. Ind. Electron.*, 2000, vol. 2, pp. 546–551.

[26] K. Kiguchi, K. Watanabe, and T. Fukuda, "Trajectory planning of mobile robots using DNA computing," *J. Adv. Comput. Intell., Intell. Informat.*, vol. 8, no. 3, pp. 295–301, 2004.

[27] B. Shackleford, G. Snider, R. J. Carter, E. Okushi, M. Yasuda, K. Seo, and H. Yasuura, "A high-performance, pipelined, FPGA-based genetic algorithm machine," *Genetic Programm. Evolvable Mach.*, vol. 2, no. 1, pp. 33–60, Mar. 2001.

[28] J. C. Gallagher, S. Vigraham, and G. Kramer, "A family of compact genetic algorithms for intrinsic evolvable hardware," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 111–126, Apr. 2004.

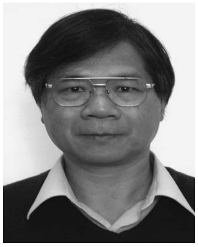
[29] H. C. Hsu and A. Liu, "A flexible architecture for navigation control of a mobile robot," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 3, pp. 310–318, May 2007.

[30] M. Serra, T. Slater, J. C. Muzio, and D. M. Miller, "The analysis of one-dimensional linear cellular automata and their aliasing properties," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 9, no. 7, pp. 767–778, Jul. 1990.

[31] R. E. Woods and R. C. Gonzalez, *Digital Image Processing,* 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[32] S. Yue, D. Henrich, W. L. Xu, and S. K. Tso, "Point-to-point trajectory planning of flexible redundant robot manipulators using genetic algorithms," *Robotica*, vol. 20, no. 3, pp. 269–280, May 2002.

[33] T. W. Manikas, K. Ashenayi, and R. L. Wainwright, "Genetic algorithms for autonomous robot navigation," *IEEE Instrum. Meas. Mag.*, vol. 10, no. 6, pp. 26–31, Dec. 2007.



Ching-Chih Tsai (S'90–M'92–SM'00) received the Diplomat in electrical engineering from the National Taipei Institute of Technology, Taipei, Taiwan, the M.S. degree in control engineering from the National Chiao-Tung University, Hsinchu, Taiwan, and the Ph.D. degree in electrical engineering from the Northwestern University, Evanston, IL, in 1981, 1986, and 1991, respectively.

He has published and coauthored over 280 technical papers and he is currently a Distinguished Professor in the Department of Electrical Engineering, National Chung-Hsing University, Taichung, Taiwan. From 2003 to 2005, he served as the Director at Center for Research Development and Engineering Technology, College of Engineering, National Chung-Hsing University. In 2006, he served as the Chair, Taipei chapter, IEEE Robotics and Automation Society and the Director at the Center for Advanced Industry Technology and Precision, National Chung-Hsing University. Since 2007, he has been the President, Taichung Chapter, the Chinese Institute of Engineers, Taiwan, and since 2009, he has been the Chair, Taichung Chapter, IEEE Systems, Man, and Cybernetics Society. He is an IET Fellow and a Fellow of Chinese Automatic Control Society. His current research interests include mechatronics, advanced control methods, intelligent control, embedded control systems, and their applications to mobile robots and industrial control.

Hsu-Chih Huang (S'08–M'09) received the M.S. degree in the Institute of Biomedical Engineering, National Cheng-Kung University, Tainan, Taiwan, and the Ph.D. degree in electrical engineering from the National Chung-Hsing University, Taichung, Taiwan, in 1999 and 2008, respectively.

He is currently an Assistant Professor in the Department of Computer Science and Information Engineering, HungKuang University, Taichung, Taiwan. His current research interests include intelligent control, mobile robots, embedded systems, SoPC, and nonlinear control.

Shui-Chun Lin (S'06–M'09) received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from the National Chung Hsing University, Taichung, Taiwan, in 2002 and 2008, respectively.

He is currently an Associate Professor in the Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung, Taiwan. His current research interests include nonlinear control, adaptive control, sliding mode control, and their applications to power electronics and industrial systems.