

FPGA Based Offline 3D UAV Local Path Planner Using Evolutionary Algorithms for Unknown Environments

Abdurrahman Bayrak

Department of Computer Engineering
Hacettepe University
Ankara, Turkey
abayrak26@gmail.com

Mehmet Önder Efe

Department of Computer Engineering
Hacettepe University
Ankara, Turkey
onderefe@gmail.com

Abstract— This paper presents an FPGA based synthesizable offline UAV local path planner implementation using Evolutionary Algorithms for 3D unknown environments. A Genetic Algorithm is selected as the path planning algorithm and all units of it are executed on a single FPGA board. In this study, Nexys 4 Artix-7 FPGA board is selected as the target device and Xilinx Vivado 2015.4 software is used for synthesis and analysis of HDL design. Local path planner is designed in a way that it has two flight modes: free elevation flight mode and fixed elevation flight mode. Designed FPGA based local path planner which has 74 MHz operating frequency and 62% logic slice utilization is tested into two different unknown environments generated by a LIDAR sensor. Results show that both GAs are efficient path planning algorithms for UAV applications and FPGAs are very suitable platforms for flight planning periphery.

Keywords— evolutionary algorithms; FPGA; genetic algorithm; LIDAR; local path planner; UAVs

I. INTRODUCTION

Over the last few decades, Unmanned Aerial Vehicles (UAVs) have been used in many different areas from military contexts to civilian contexts including surveillance, reconnaissance, targeting, weather forecasting, fire detection and avoidance of danger zones, etc. Their uses are still in progress as a strong tool for all fields, and the improving the autonomous structure of them is still a hot research topic because of their many advantages such as high mobility, low cost, lightweight structure and no vital risk.

To obtain a more autonomous UAV and achieve all UAV missions, a common task is path planning which is an NP-hard problem. The main objective of path planning is to find the optimal path minimizing the overall path length and maximizing of the safety of the vehicles from the starting point to the destination point by taking into account of some operational constraints relating to the environment, the mission and the UAV dynamics. There are many different path planning algorithms which are used in both aerial and ground vehicle applications in the literature [1-10]. Among them, Evolutionary Algorithms (EAs), which are effective optimization tools, have been typically implemented to UAV path planning problems [11-17]. Even though EAs do not guarantee the best solution,

they generate near optimal solutions for UAV path planning problems when compared to other path planning algorithms. However, the main disadvantage of EAs for path planning applications, when UAV mission is a constrained one, the computational time required to find the optimal solution increases. To overcome this problem, and at the same time, to improve the search performance, developing parallel models of EAs has a critical importance.

Field Programmable Gate Array (FPGA), which is an integrated circuit and generally uses a Hardware Description Language (HDL), is often preferred to obtain a hardware parallelism because of its many advantages such as parallel processing capability, lightweight, lower cost and low power requirements, especially for on board applications. It is also used for real-time path planning applications [18, 19, 20, 21] and to obtain path planner using a parallelization model of EAs [22, 23].

In [18], an FPGA based parallel hardware implementation of dynamic trajectory planning for mobile robot navigation applications has been presented. Harmonic control is chosen to estimate trajectories. In [19], the authors have presented a path planning implementation for mobile robots. They have used Breadth First Search Algorithm (BFSA) as path planning algorithm, and applied it both in software and in hardware. [20] proposes a novel hardware directed algorithm to find a path for a mobile robot. The authors have used the Euclidean distance transform of the binary image captured from an overhead camera of an environment. In [21], an efficient FPGA implementation of a parallel path planning algorithm based on marching pixels has been developed.

In [22], a real time path planner is presented. The authors have used Genetic Algorithms (GAs) which were modified by Cocaud, [24], as path planning algorithm and improved the EAs on FPGA board. They directly implement all modules of GAs on FPGA except evaluation module, which is performed on PC because of high computation processing requirement. In [23], the authors have proposed an FPGA based UAV path planner architecture for a 3D 512×512×512 size environment using Cocaud's modified GAs. They have implemented all modules of GAs entirely on FPGA and used LIDAR data as terrain information.

In this paper, an FPGA based offline UAV local path planner implementation using Cocaud's modified GAs for 3D unknown environments is presented and explained in detail. FPGA path planning architecture and the approach of this study is similar to that in [23] in that it resembles in the population, the chromosome and the evaluation hardware parameters, yet it differs from [23] in performing finite state machine representation of the overall system and in the implementation of GA processes such as the used sorting algorithm. All modules of EAs is directly run on a single FPGA and the terrain data is obtained from a LIDAR sensor. The contribution of our work is the designing of an efficient FPGA based 3D local path planner, which can be used in real time UAV applications, especially for replanning applications.

The rest of paper is organized as follows: section II gives basic system descriptions for path planning. In section III, FPGA architecture of the overall system is explained in detail. The next section is devoted for experiments and results. In the last section, conclusion is presented.

II. BASIC SYSTEM DESCRIPTIONS FOR PATH PLANNING

Path planning is a mission critical issue in autonomous UAV flight. The fundamental problem of path planning is to find a collision free path between a given initial point and a destination point under some constraints in both static and dynamic environment. It is aimed to obtain a decision mechanism for UAV missions without a human operator verifying whether path is feasible or not. In order to achieve that, GAs, which are effective search methods, and a particular class of EAs are often used in real time UAV path planning applications.

A. A Path Planning Algorithm: Genetic Algorithm

Genetic algorithm which is an effective heuristic search method is based on the principle of natural selection and natural genetics. In this study, Cocaud's modified genetic algorithm used as path planning algorithm. The details about GA will be explained in Section III. Fig. 1 shows pseudo-code of modified GA used for this study.

```

Initialize the random population;
Evaluate initial population;
LOOP
  Selection;
  Genetic Operation;
  Update Population;
  IF (Premature Convergence)
    Reinitialize the random population;
  END-IF
END-LOOP

```

Fig. 1 Pseudo-code of modified genetic algorithm

B. Path Planning Constraints

Environment, mission and UAV constraints are important for autonomous UAV path planning applications as the constraints determine the capability of path planner.

In this study, terrain elevation information is obtained by using LIDAR, and obtained spatial coordinates are used to

generate $512 m^3$ integer grid map. Grid map is composed of an $X \times Y$ matrix and Z is the elevation value corresponding to the specific coordinate indicated by X and Y . To achieve a mission, designed path planner has two flight modes: free flight and fixed elevation flight, and it has four transitional waypoints in addition to starting and destination point. Finally, it is supposed that UAVs considered this study have maneuver capability in $1 m^3$. As a result, they are considered as a point. Mission and UAV constraints can be increased with additional constraints.

III. FPGA ARCHITECTURE

In this section, overall hardware architecture is explained. Fig.2 demonstrates FPGA path planner architecture. Proposed architecture is composed of a Control Unit (CU) which manages overall process iteratively and six subunits which are GA processes. In addition to the parent and offspring populations, environment (terrain information), mission and UAV constraints are stored in block random access memories (BRAMs) in the FPGA. Overall process begins when the activate signal is on (See Fig. 2).

A. Population and Chromosome Characteristics

GA chromosomes which constitute parent and offspring populations are possible solutions for an optimization process. In this study, a chromosome is composed of, four transitional waypoints and a fitness value corresponding to associated chromosome. Starting and destination points are constant during path planning process and stored in mission profile. Transitional waypoint number can be increased or decreased. However, four transitional waypoints are adequate to produce satisfactory results [23]. Each transitional waypoint includes three spatial coordinates (X , Y , and Z) each containing 9-bits integer value. Fitness value is a 32-bits integer value. Parent and offspring populations, which are stored in BRAMs, consist of 32 chromosomes.

B. Genetic Algorithm Processes

Overall architecture is designed by using Very High Speed Integrated Circuit Hardware Description Language (VHDL).

1) *Control Unit*: When the activate signal is on, the optimization process starts. CU manages overall process by running the Finite State Machine (FSM) shown in Fig. 3.

2) *Initialize Population Unit (IPU)*: IPU constitutes the first step of the proposed system. Designed IPU includes twelve 9-bit random number generators based on Linear Feedback Shift Register (LFSR), which run during the entire process. It should be noted that, some bits of them are used in the other random processes of GA with required bit length. Each one of them generates one spatial coordinate of one chromosome, which has four transitional waypoints including three spatial coordinates. Generating one chromosome in IPU takes one cycle. IPU generates 32 chromosomes and it saves them to parent population memory. For reinitialize process, when 'Reinit'

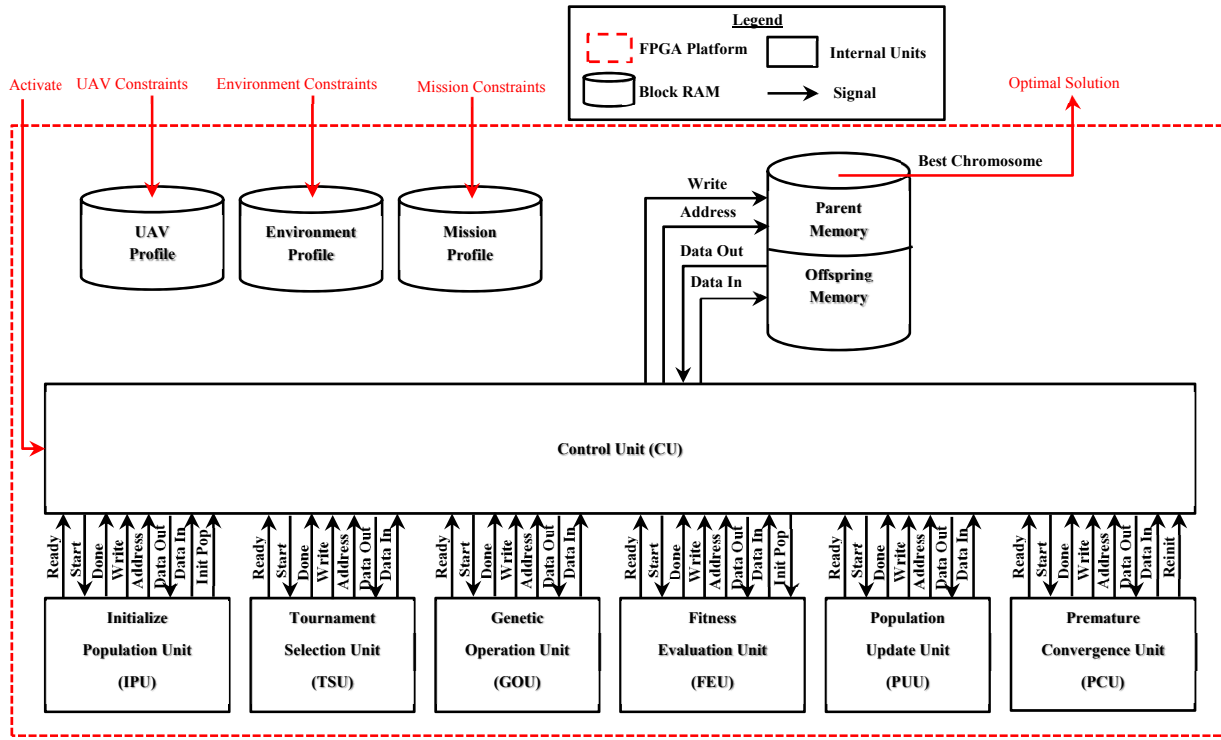


Fig.2 FPGA path planner overall architecture

signal is active (see Fig. 2), it preserves the best chromosome and regenerates 31 new random chromosomes.

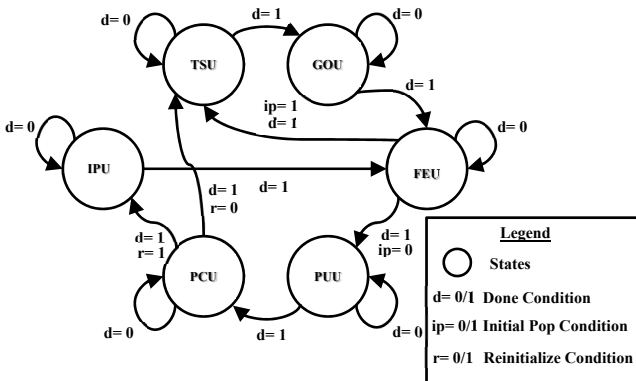


Fig. 3 CU state diagram

3) *Fitness Evaluation Unit (FEU)*: FEU determines the feasibility of a path (a chromosome) in the parent or offspring memory by calculating a fitness value concatenating 24-bit distance and 8-bit collision cost. Fig. 4 shows the structure of the fitness value. Distance cost is calculated by using square of the Euclidean distance as in (1).

$$D = (X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2 \quad (1)$$

Additional constraints can be added in this unit. These additional costs may be related to UAV kinematics, temporal needs, danger zones, energy resources or those derived from the mission requirements.

4) *Tournament Selection Unit (TSU)*: TSU determines chromosomes, which will undergo genetic operation by using tournament selection method. TSU includes 32 parallel comparator components, and each one of them selects two chromosomes from parent memory randomly and compares them according to their fitness values. Chromosome with lower fitness value (winner) is recorded as offspring memory.

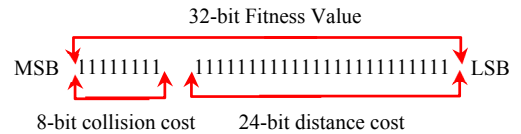


Fig. 4 The structure of the fitness value

5) *Genetic Operation Unit (GOU)*: GOU applies genetic operation including crossover and mutation to chromosomes of the offspring memory. Figs. 5-9 demonstrate genetic operation phases. For crossover operation, firstly, two chromosomes are selected randomly. Secondly, randomly selected two chromosomes are truncated from their splitting point, and finally truncated segments are swapped as shown in Fig. 5.

As mutation operation, four mutation types are used, namely, “global perturb”, “local perturb”, “delete” and “swap” mutation. Global perturb operation is as shown in Fig. 6. For this mutation operation, a randomly selected waypoint of a randomly selected chromosome is regenerated randomly within the associated chromosome boundaries. For local mutation operation, a randomly selected waypoint of a randomly selected chromosome is regenerated randomly within the previous and

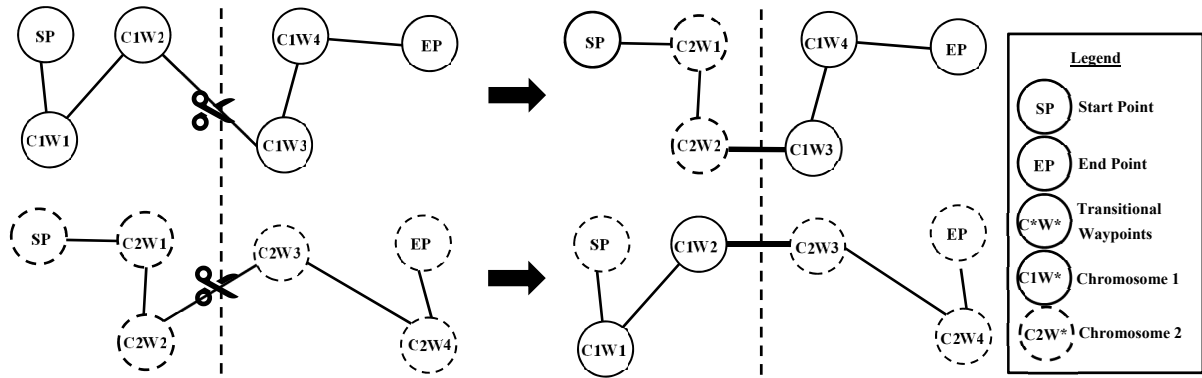


Fig. 5 An example of crossover operation

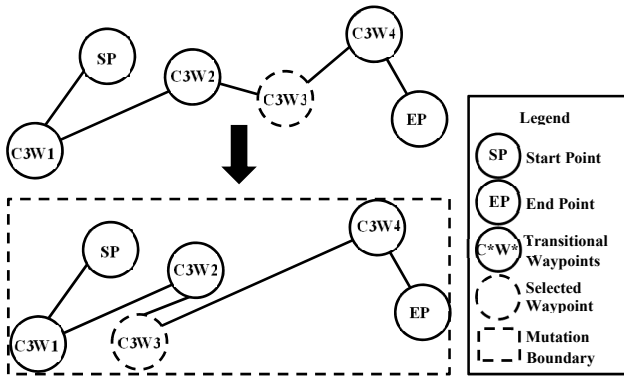


Fig. 6 Global perturb mutation operation

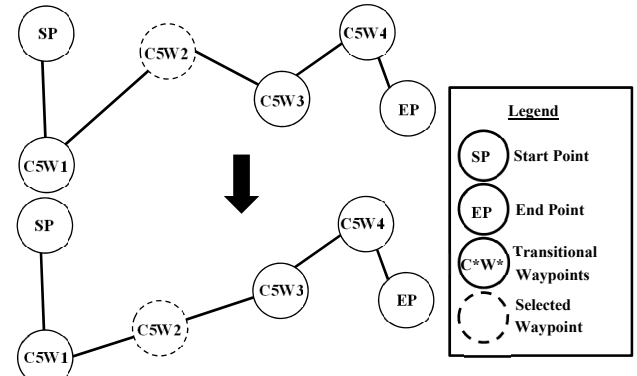


Fig. 8 Delete mutation operation

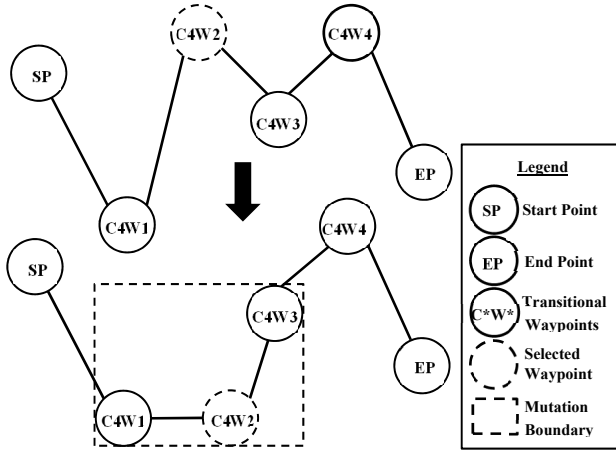


Fig. 7 Local perturb mutation operation

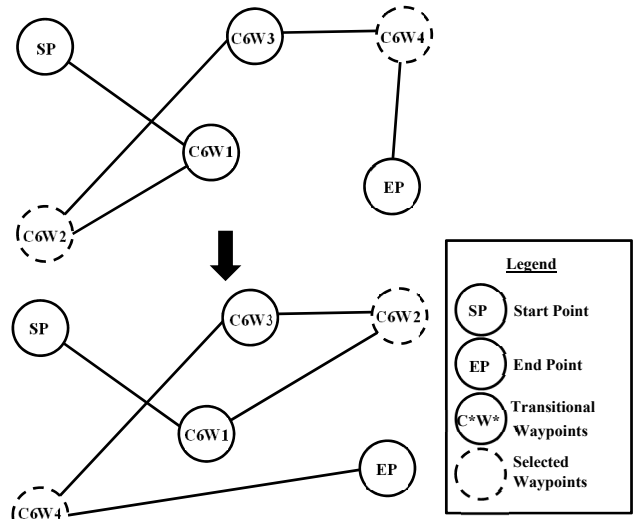


Fig. 9 Swap mutation operation

the next waypoint boundaries of selected waypoint. This is illustrated in Fig. 7. For “delete” type of mutation operation, a randomly selected waypoint of a randomly selected chromosome is taken as midpoint of the previous it (Fig. 8) and next waypoint of selected waypoint, instead of deleting it.

Finally, for “swap” type of mutation operation, two randomly selected waypoints of a randomly selected chromosome are exchanged as shown in Fig. 9.

GOU is composed of 24 crossover units, which corresponds to 75% of the offspring population size. The remaining 8 chromosomes are 2 for global perturb, 2 for local perturb, 2 for delete and 2 for swap mutation units, which corresponds to 25% of the offspring population size. It should be emphasized that all crossover units considered in this work are identical.

6) *Population Update Unit (POU)*: POU updates the offspring memory according to fitness value of 64 chromosomes in both parent and offspring memory. In this unit, parallel odd-even sorting algorithm is used, and parent and offspring chromosomes are sorted in ascending order according to their fitness values. Then, as the last step in this unit, the first 32 chromosomes are overwritten to offspring memory.

7) *Premature Convergence Unit (PCU)*: One of the disadvantages of GAs is premature convergence, that is, diversity is lost early. The loss of diversity could be understood by looking at offspring memory chromosomes. If offspring population chromosomes all are the same, the population diversity is lost. PCU determines whether population diversity is lost or not, and it is composed of 31 parallel comparators. If population diversity is lost, then ‘Reinit’ signal is set to high in order to start reinitialize process.

IV. EXPERIMENTS AND RESULTS

A. FPGA Synthesis

In this study, Nexys 4 Artix-7 FPGA board is selected as the target device and Xilinx Vivado 2015.4 software is used for synthesis and analysis of the HDL design. Overall system is synthesized successfully on the FPGA board. Table 1 shows the hardware synthesis details. In the table, Slice Look-Up Tables (LUTs) represent slice logic utilization, FF stands for flip-flops, BUFG denotes the global clock buffers and IO is the abbreviation for Input/Output. For the designed local path planner, the logic slice utilization percentage on the device is 62% and the maximum device operating frequency is 74 MHz.

TABLE I. SYNTHESIS DETAILS

Resource	Utilization	Available	%Utilization
Slice LUTs	39440	63400	62.21
FF	33250	126800	26.22
BRAM	73	135	54.07
IO	3	210	1.43
BUFG	1	32	3.12

B. Experiments

In this section, proposed path planning architecture is tested for two different unknown 512m high×512m long×512m wide environments generated from LIDAR data. LIDAR data is obtained from [25]. Generated environments are stored within FPGA BRAMs.

Designed path planning architecture is executed in two different flight modes for each environment. For a given starting and destination point, free elevation flight mode, called MODE-I, is a collision free flight, which does not have any elevation constraints. The other flight mode, fixed elevation flight mode, called MODE II, is a collision free flight, which has a predefined and constant flight elevation value in mission profile.

C. Results

Designed path planning architecture is executed during 1000 generations, and then it is observed that found optimal solutions have not changed after the generation numbers in Table II. Figs.

10-11 demonstrate the results for test environments. In Fig. 10, for a given start and destination point, MODE I took 147 generations and MODE II took 86 generations. In Fig. 11, MODE I took 90 generations and MODE II took 85 generations. Table II shows computational time results for convergence. Results indicate that parallel GAs are efficient path planning algorithms for UAV applications and results also show that FPGAs are very suitable devices for UAVs. Convergence times support this claim clearly.

TABLE II. AVERAGE COMPUTATIONAL TIME RESULTS AT CONVERGENCE

Flight Tests	Generation Number	Convergence Time
MODE I-ENV I	147	43.9 ms
MODE II-ENV I	86	25.8 ms
MODE I-ENV II	90	26.2 ms
MODE II-ENV II	85	25.5 ms

V. CONCLUSION

In this paper, an FPGA based synthesizable offline UAV local path planner implementation using Cocaud’s modified GAs for 3D unknown environments is presented and explained in detail. All processes are executed on a single FPGA module. Firstly, an environment elevation grid map is generated from LIDAR data and stored into FPGA BRAMs together with other constraints. Secondly, Cocaud’s modified GA is implemented on FPGA as shown in Fig. 2. Proposed local path planner is designed in a way that it would have two flight modes, namely, free elevation flight mode and fixed elevation flight mode. Finally, for given start and destination points, proposed architecture is tested into two different unknown environments. Furthermore, FPGA synthesis details are presented.

Results show that both parallel GAs are efficient path planning algorithms for UAV applications and FPGAs are very suitable devices for UAVs. Furthermore, convergence time results indicate that designed local path planner can be used in real time UAV replanning applications.

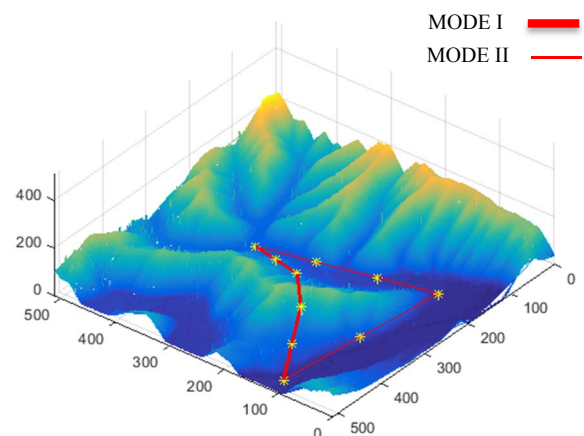


Fig. 10 MODE I and MODE II test results for environment I (Fixed elevation value is 40 meters for MODE II)

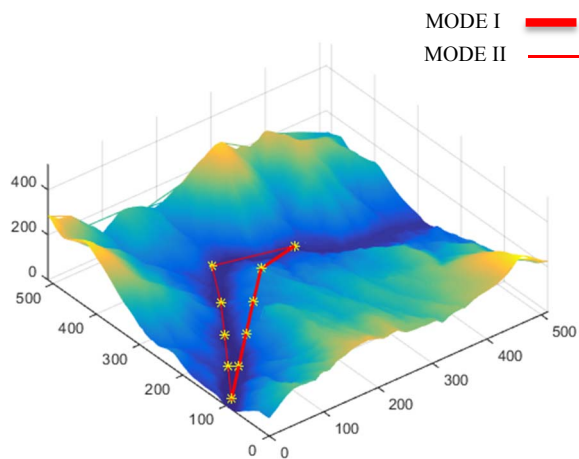


Fig. 11 MODE I and MODE II test results for environment II (Fixed elevation value is 20 meters for MODE II)

ACKNOWLEDGMENTS

The authors would like to thank Autonomous Systems Laboratory of Computer Engineering Department of Hacettepe University. The authors gratefully acknowledge the TÜBİTAK-ARRS Project, Reference No 114E954.

REFERENCES

- [1] A. Tsourdos, B. A. White, and M. Shanmugavel, "Cooperative path planning of unmanned aerial vehicles," Vol. 32. John Wiley & Sons, 2010.
- [2] G. Carbone, and F. Gomez-Bravo, eds, "Motion and operation planning of robotic systems: Background and Practical Approaches," Vol. 29. Springer, 2015.
- [3] K. P. Valavanis, ed, "Advances in unmanned aerial vehicles: state of the art and the road to autonomy," Vol. 33. Springer Science & Business Media, 2008.
- [4] H. Fazlollahabadi, and M. Saidi-Mehrabadi, "Autonomous guided vehicles," Springer, 2015.
- [5] E. G. Gilbert, and D. W. Johnson, "Distance functions and their application to robot path planning in the presence of obstacles," *Robotics and Automation, IEEE Journal of* 1.1 (1985): 21-30.
- [6] C. C. Tsai, H. C. Huang, and C. K. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *Industrial Electronics, IEEE Transactions on* 58.10 (2011): 4813-4821.
- [7] P. Raja, and S. Pugazhenthii, "Optimal path planning of mobile robots: A review," *International Journal of Physical Sciences* 7.9 (2012): 1314-1320.
- [8] Y. Zhang, D. W. Gong, and J. H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing* 103 (2013): 172-185.
- [9] L. De Filippis, G. Guglieri, and F. Quagliotti, "Path planning strategies for UAVS in 3D environments," *Journal of Intelligent & Robotic Systems* 65.1-4 (2012): 247-264.
- [10] L. Blackmore, Hui Li and B. Williams, "A probabilistic approach to optimal robust path planning with obstacles," *2006 American Control Conference*, Minneapolis, MN, 2006, pp. 7 pp.-. doi: 10.1109/ACC.2006.165665
- [11] N. Özalp, and O. K. Şahingöz, "Optimal UAV path planning in a 3D threat environment by using parallel evolutionary algorithms," *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*. IEEE, 2013.
- [12] S. Mittal and K. Deb, "Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms," *2007 IEEE Congress on Evolutionary Computation*, Singapore, 2007, pp. 3195-3202. doi: 10.1109/CEC.2007.4424880
- [13] D. Rathbun, S. Kragelund, A. Pongpunwattana and B. Capozzi, "An evolution based path planning algorithm for autonomous motion of a UAV

- through uncertain environments," *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*, 2002, pp. 8D2-1-8D2-12 vol.2. doi: 10.1109/DASC.2002.1052946
- [14] D. Jia and J. Vagners, "Parallel evolutionary algorithms for uav path planning," in *Proc. AIAA 1st Intelligent Systems Technical Conference*, 2004.
- [15] I. Hasircioglu, H. R. Topcuoglu, and M. Ermis, "3-D path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms," in *Proc. 10th Annu. Conf. Genetic Evol. Comput.*, Atlanta, GA, 2008, pp. 1499-1506.
- [16] I. K. Nikolos, E. S. Zografos, and A. N. Brintaki, "UAV path planning using evolutionary algorithms," in *Innovations in Intelligent Machines*, S. C. Javaan, C. J. Lakhmi, M. Akiko, and S.-I. Mika, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 77-111.
- [17] V. Roberge, M. Tarbouchi and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV Path Planning," in *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132-141, Feb. 2013. doi: 10.1109/TII.2012.2198665
- [18] B. Girau, A. Boumaza, "Embedded harmonic control for dynamic trajectory planning on FPGA," *The IASTED International Conference on Artificial Intelligence and Applications - AIA 2006*, Feb 2007, Innsbruck, Australia. 2007.
- [19] L. Vacariu, F. Roman, M. Timar, T. Stanciu, R. Banabic, O. Cret, "Mobile robot path-planning implementation in software and hardware," *Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation*. World Scientific and Engineering Academy and Society (WSEAS), 2007.
- [20] N. Sudha, and A. R. Mohan, "Design of a hardware accelerator for path planning on the Euclidean distance transform," *Journal of Systems Architecture* 54.1 (2008): 253-264.
- [21] M. Schmidt, and D. Fey, "An optimized FPGA implementation for a parallel path planning algorithm based on marching pixels," *Reconfigurable Computing and FPGAs (ReConFig), 2010 International Conference on*. IEEE, 2010.
- [22] F. C. J. Allaire, M. Tarbouchi, G. Labonte, G. Fusina, "FPGA implementation of genetic algorithm for UAV real-time path planning," *Unmanned Aircraft Systems*. Springer Netherlands, 2008. 495-510.
- [23] J. Kok, L. F. Gonzalez, and N. Kelson, "FPGA implementation of an evolutionary algorithm for autonomous unmanned aerial vehicle on-board path planning," *Evolutionary Computation, IEEE Transactions on* 17.2 (2013): 272-281.
- [24] C. Cocaud, "Autonomous tasks allocation and path generation of UAV's," Diss. University of Ottawa (Canada), 2007.
- [25] <http://opentopo.sdsc.edu/lidar>