FPGA BASED OBSTACLE AVOIDANCE AND PATH PLANNING FOR UAV SYSTEMS USING LIDAR SENSOR

Murad Yaghi[1], Mehmet Önder Efe[2] and Kıvanç Dinçer[3]
Hacettepe University
Ankara, Turkey

ABSTRACT

An FPGA based path-planning algorithm is proposed for controlling unmanned air vehicles (UAV) system, which helps the UAV to track and avoid obstacles around it in real time. Such a UAV can avoid random obstacles with non-unified shape and determine the best route to follow without colliding with them. The environmental data is gathered via a laser based (LIDAR) sensor system on the UAV and is processed by the FPGA, also on the UAV, in real time. It is seen that onboard FPGA processing is a feasible alternative for such UAV and it can significantly speed up the reaction time as compared to traditional systems where such processing is done in the ground station.

1. INTRODUCTION

Path planning for unmanned air vehicles (UAV) is an important field of research that has many approaches varying from traditional optimal trajectory planning algorithms to innovative bio-inspired swarm behavior algorithms. Creating a realistic path depends on many complex factors such as the presence of multiple UAVs, obstacles, threats, multiple tasks for UAV and uncertainty in the environment [Chen, Chang and Agate, 2013]. These (UAV) systems allow untrained and unskilled humans to perform many tasks efficiently such as tele operated tasks in remote indoor environments, further, they can be used in military applications, security settings and environments that is considered to be hazardous [Ferrick, Fish, Venator and Lee, 2012]. In order to implement an obstacle avoidance algorithm, well-suited sensors should be used to gather accurate information about the environment and obstacles. Usually, pure vision systems are considered to be unreliable because of the variation in illumination conditions, which can affect the reading and processing of data adversely. The laser based (LIDAR) sensor systems are considered to be much more robust especially in off-road outdoor environments [Esteban and Rosales, 2011]. LIDAR based mapping approach is now considered as one of the most accurate scheme for generating spatial information about the shape and surface characteristics of any object. Recent advancements in LIDAR technologies for mapping allow researchers and practitioners to examine synthetic and natural environments that are composed of a wide range of scales with higher precision, accuracy and flexibility than ever before [Carter, Schmid, Waters, Betzhold, Hadley, Mataosky and Halleran, 2012]. LIDAR sensors also considered as an effective solution to the problem of obstacle detection and recognition as it is a tough problem because of the varying characteristics of obstacles in the real world. So the detection and recognition of these obstacles poses challenges to the image processing.

In [Hrabar, 2011], a new 3D goal-directed reactive obstacle avoidance algorithm has been presented, which is designed specifically for rotorcraft Unmanned Aerial Vehicles (RUAVs) that has trajectories of a point-to-point type. The proposed algorithm is designed in a way to create cylindrical safety volume, which is placed ahead of the UAV. The cylindrical safety volume is able to detect potential collisions

[1] Doctoral Researcher at the Department of Computer Engineering, Email: muradyaghi7@outlook.com

[2] Faculty at the Department of Computer Engineering, Email: onderefe@hacettepe.edu.tr

[3] Faculty at the Department of Computer Engineering, Email: kivanc.dincer@hacettepe.edu.tr

inside it. This safety volume is created in a 3D occupancy map which represents the environment around it. After that a search process is performed in order to find an escape point. In [Hoerner and Gmbh, 2012], a prototype of a lightweight, low power and fully autonomous 2D LIDAR system has been implemented. It has been tested as a sample application for scanning the surrounding area. The system is able to read data in a range of 0-74.5 m for distance measurements and 336 degrees as an angular coverage. It also has a maximum sample rate of 39.06 kHz and a square signal modulated laser diode which has quasi continuous wave (QCW) for high efficiency.

The work reported in [Buniyamin, Wan, Sariff and Mohamad, 2011] presents an overview of many path planning algorithms that are used for autonomous robots. The bug algorithm family is studied as a local path planning algorithm. Bug algorithms utilize limited information about the environment and they sense the nearest mobile robot as an obstacle toward the target. With this algorithm, border of an obstacle is used as a guide toward the target. The result of [Buniyamin, Wan, Sariff and Mohamad, 2011] is a new algorithm called Point Bug, which attempts to reduce the use of outer perimeter by looking for some special points on the outer perimeter of the obstacle area. These are critical points and the algorithm produces a complete route from source to destination. Ogworonjo [Ogworonjo, 2011] states the problem of obstacle detection and recognition as a complex problem because of the changing size and structure characteristics of obstacles in real world, so the detection and recognition of these obstacles poses serious challenges to the image processing. This task may be even more difficult if the obstacle space is dynamically changing. Analyzing and extracting the features of dynamic objects via image processing is considered to be a very challenging task as these kinds of objects will not just change their appearances; they will also be fully or partially hidden during the motion. In [Ogworonjo, 2011], Ogworonjo propose a solution for handling dynamic objects by using LIDAR sensors in addition to the image sequences to improve the task of reconstruction, recognition, classification and segmentation. In [Ladha, Kumar, Bhalla, Jain and Mittal, 2011], Ladha et al. developed a computationally efficient algorithm to test on a UAV platform, which is of type quad rotor. The vehicle is intended to plan its trajectory and to avoid the obstacles in an absolutely unknown environment. The endpoints are assumed to be unknown; further, there is no prior knowledge about the environment around the UAV. It is stated that this algorithm works well when there are multiple obstacles in the flight environment.

This paper is organized as follows: the second section is devoted to the sensor modules including the LIDAR sensor that are used for path planning with robots. The third section is about applying path planning on FPGA, which is the main contribution of our study. Finally, the experimental results are discussed at the end of the paper.

2. SENSOR MODULES IN PATH PLANNING AND LIDAR

In autonomous UAV systems, flight safety and obstacle avoidance are the core issues that need to be studied carefully. The vehicles are equipped with several sensors to perceive the environment and to avoid possible collisions. These sensors typically include LIDAR sensors, proximity sensors (infrared, ultrasound), and vision systems of different types. The choice of the sensor to be used on-board and its modularity is an important issue as the richness of the raw information is the fundamental property. Usually infrared sensors are used for indoor applications with small robots, so are sonar devices, but the sonar systems usually designed to be working with large robots. Vision-based systems are useful for both indoor and outdoor navigation as there are many alternatives with different technical capabilities. Pure vision systems are sometimes unreliable because of the variation in illumination conditions, which can adversely affect the reading and processing of data. The laser based (LIDAR) sensor systems are considered to be much more precise especially in off-road outdoor environments [Esteban and Rosales, 2011]. Present study focuses on the use of LIDAR sensor.

LIDAR stands for light detection and ranging, and it is considered as a remote sensing technology that emits a focused and intense beam of light and then calculate the distance for each beam based on the time it needs to go, hit the obstacle and then come back until it reaches the source again, depending on this, we could say that LIDAR is the same as radar (radio detection and ranging) except that the principle of LIDAR is based on emitting discrete pulses of laser light. The image that can be constructed from the information extracted from LIDAR pulses depends on the following factors:

- The distance between the object and the laser beam source
- The angle of emitting the laser beam from the source
- The exact position of the sensor

The advantages of using LIDAR are the high resolution observations, algorithmic possibilities of determining the objects in the collected point cloud data, and the possibility of sensing stationary objects precisely. However, the use of LIDAR requires the knowledge of the position to register the collected data, displays some sensitivity to meteorological conditions and difficulties in tracking moving objects.

The most important point in these disadvantages is the critical dependence to position information, which is typically equipped with inertial navigation systems, GPS support and simultaneous localization and mapping (SLAM) like techniques to fuse the information in an optimal sense to derive the most precise position information.

There are two major types of LIDAR, namely waveform LIDAR and discrete return LIDAR. The former is useful in large areas as every reflection is recorded separately whereas the latter is useful in narrow areas and its resolution is comparably good. Typically, a LIDAR system has the wavelength 900-1064 nanometers and vertical precision is about 5-15 cm whereas the horizontal precision is 30-50 cm. Depending on the needs of the application in hand, higher resolution hardware can be obtained at some cost. Today, military applications or space applications of LIDAR sensor has such versatile modules.

The output of LIDAR is a point cloud, which is essentially comprised of x, y and z data of the occupied points in the 3D space. Considering the motion of the sensor itself, it becomes important to utilize the registration of data to obtain best alignment and unification of the flight environment digitally.

Looking at the alternatives, one can easily see that there are auxiliary hardware for LIDAR sensors that provide some millions of data per second and processing of such an intensive scheme needs a versatile computing platform to cope with the challenges of time critical issues. Field Programmable Gate Array (FPGA) is a good auxiliary hardware to address this problem as the device offers a reconfigurable hardware to the designer, so that the designer can describe the processing hardware by using some hardware description language (HDL). In what follows, we explain the proposed technique to handle the measurements from the LIDAR sensor.

3. THE PROPOSED WORK

Here we will review the basic ideas about the proposed algorithm in [Ladha, Kumar, Bhalla, Jain and Mittal, 2011] and then we will modify this algorithm to let it more efficient and suitable to be synthesized on FPGA for real time computing.

3.1. The Growth Algorithm

In order to detect the threats around the UAV, it's important to study a certain degree-span directly projected in front of the UAV which is calculated based on knowing the minimum clearance that is required for the UAV to pass between two obstacle points (W), and a safe-distance (D). These are illustrated in Figure. 1.

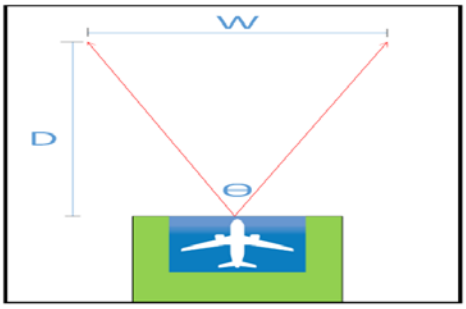

Figure 1: *UAV safety margins*.

$$cos(\frac{\theta}{2}) = D / y \quad (1)$$

$$y = D^2 + W^2/4 \quad (2)$$

where D is the safe-distance, W is the clearance width, θ is the range to be scanned. When scanning is started, every point resulted from the scanning process is considered as an obstacle if it is at the horizontal scanning range. The growth algorithm continues by drawing a circle of radius W/2 around every such point, then all LIDAR beams crossing the circles are removed since no beam can cross any circle as shown in Figure. 2.

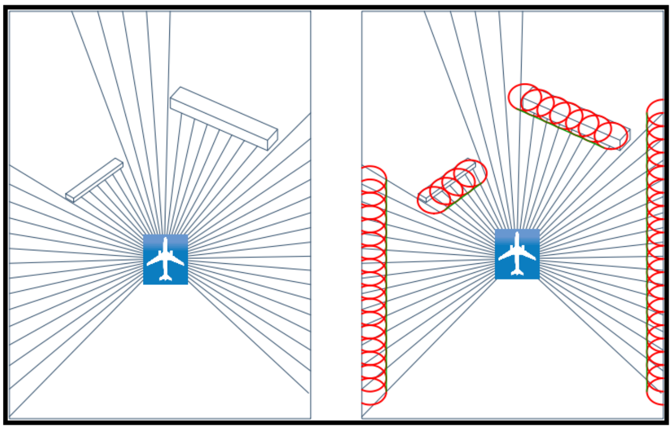

Figure 2: *LIDAR obstacle detection*.

This figure shows the scanning process and how to detect obstacles in a real environment, where the obstacles are considered as point objects. When the beam crosses a circle, we remove this beam and reduce the distance in order that no beam will enter inside any circle; by this we can compute the new lengths as shown in Figure. 3 & Figure. 4.

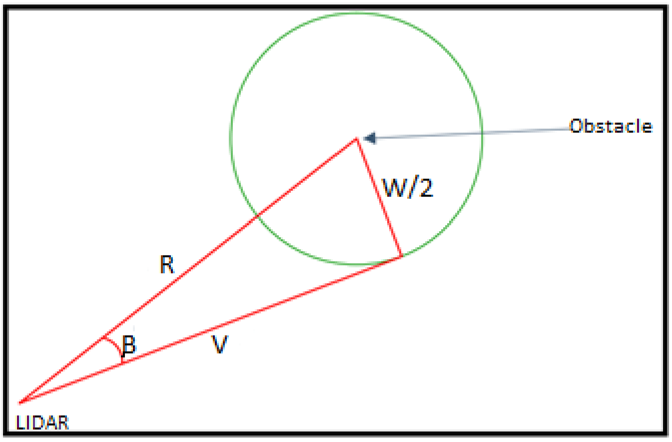

Figure 3: *Growth algorithm*.

$$\beta = cos^{-1}(\frac{V}{R}) \tag{3}$$

$$V = \sqrt{R^2 - (\frac{W}{2})^2} \tag{4}$$

where R is the distance between the LIDAR and the obstacle point, V is the length of the tangent, β is the angle between the LIDAR beam R and the circles' tangent V.

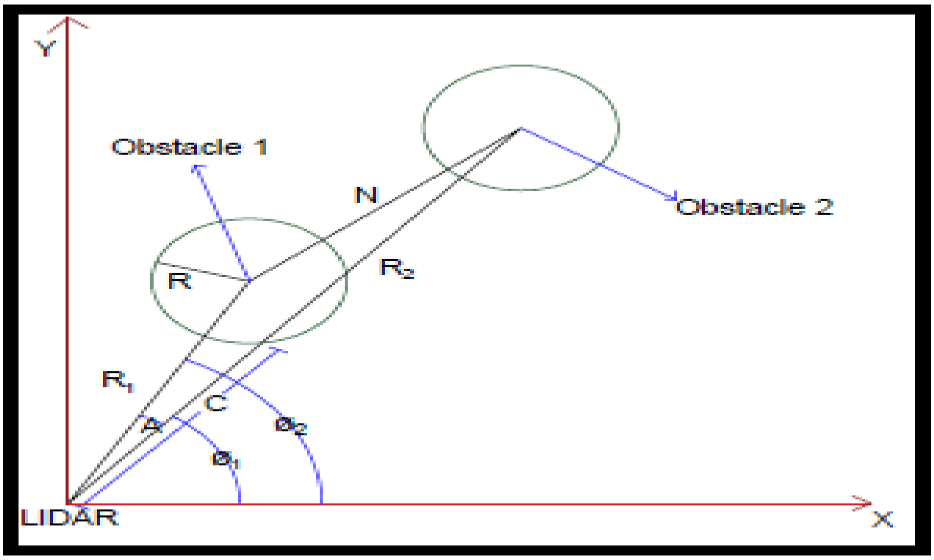

Figure 4: *Related margins for growth algorithm*.

$$A = cos^{-1}(\frac{R_1^2 + R_2^2 - N^2}{2R_1R_2}) \quad (5)$$

$$N = \sqrt{R_1^2 + R_2^2 - 2R_1R_2cos(\emptyset_2 - \emptyset_1)} \quad (6)$$

$$C = R_2\,cos(\emptyset_2 - \emptyset_1) \pm \sqrt{(R_2)^2\,cos^2(\emptyset_2 - \emptyset_1) - {R_2}^2 + R^2} \quad (7)$$

Utilizing these equations, we can find all the distances of obstacles in the scanning range. For example in Figure 4. The first obstacle has a distance of R_1, and the second obstacle has a distance of R_2. For each obstacle a circle should be drawn that demonstrates the clearance width (safe distance), and any other ray should not cross this circle. The second obstacle in the figure has a distance of R_2, but the ray toward this obstacle has crossed the circle of the first obstacle, so it should be cut and the value of N should be taken as a distance instead of R_2. By this and according to the growth algorithm we can measure all of the free distances that doesn't intersect with any obstacle or any circle around the obstacles and then chose the longest free distance as the safest path.

3.2 The Proposed Algorithm

In our work, we will propose a modification to the growth algorithm in order to make it applicable on FPGA devices. In this work, we will avoid using some mathematical functions proposed by the growth algorithm such as (cos^{-1} & square roots), which are considered to be unfriendly with FPGA devices, and at the same time we will use other functions which are more friendly with FPGA and give us the same final result for path planning. At any obstacle point, the distance from this obstacle point toward any other obstacle should be larger than a specified threshold (clearance width) which will allow the UAV to pass between these two obstacles safely, so in order to consider any beam passing between these two obstacles as a safe path, it should have a free space around it larger than the half of clearance width (W/2). So our goal here is to find whether or not there is a (W/2) wide free distance around each safe path. In order to apply this on FPGA using LIDAR system the only information that we have is:

1) The angle between two corresponding beams, which can be considered as a constant, the value of which depends on LIDAR resolution.
2) The distance between the obstacle point and the UAV.

Depending on these information we can calculate (W/2) for each path as shown in Figure. 5.

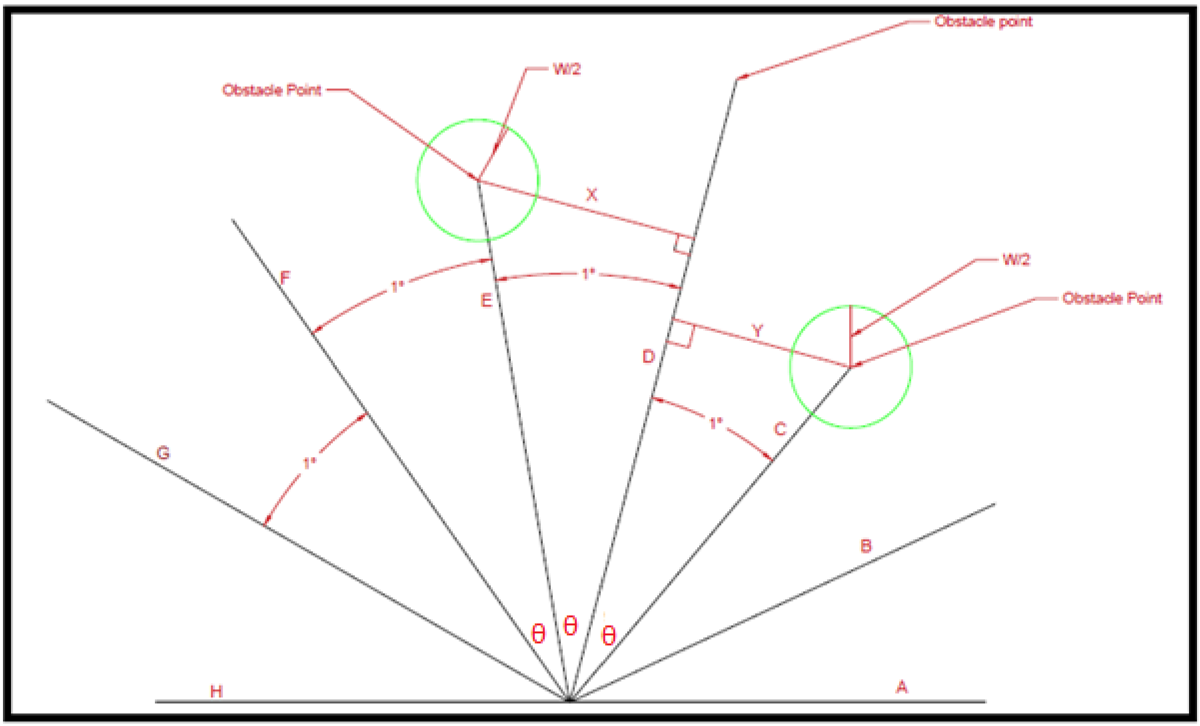

Figure 5: *Proposed path planning.*

In the proposed scheme, we will start the path planning by taking the longest path (D is the longest path as shown in Figure. 5), then we will make sure that this path is safe and it has free space more than W/2 along its way. Here we will assume that the clearance width needed for our UAV to pass through is 1 meter. So now we can calculate the safety of path (D) shown in Figure. 5 as the following equations.

$$Free\ Space = X + Y \tag{8}$$

$$\mathrm{X} = \sin(\boldsymbol{\theta})E \tag{9}$$

$$\mathrm{Y} = \sin(\boldsymbol{\theta})C \tag{10}$$

But for LIDAR each angle between two beams has a constant value, we will consider it here as 1°, so:

$$Free\ Space = 0.01745(\,E + C) \tag{11}$$

This equation is simple and it can be synthesized on an FPGA platform to compute path planning efficiently without the need to use expensive functions such as $\cos^{-1}$ or the square root.

3.3 FPGA Implementation

With the studied path planning algorithm, the FPGA will start by scanning a specified range of angles and then will store the corresponding distances for each angle, after that it will arrange the angles with its corresponding distances based on the value of distances, and finally it will calculate the free space around each path based on equation (11). If the value of free path is higher than a given threshold, (1 meter in our tests) we consider this path as safe and the UAV will follow this path, otherwise we should scan and chose other angles which has a safe path. When a safe path is found the searching process will stop and the angle of the safe path is output. Otherwise it continues searching until it computes the free space for all angles in the scanning range. If no safe path found, it will simply output the path that has the longest distance. Figure. 6 shows the flowchart of the proposed path planning implementation on FPGA.

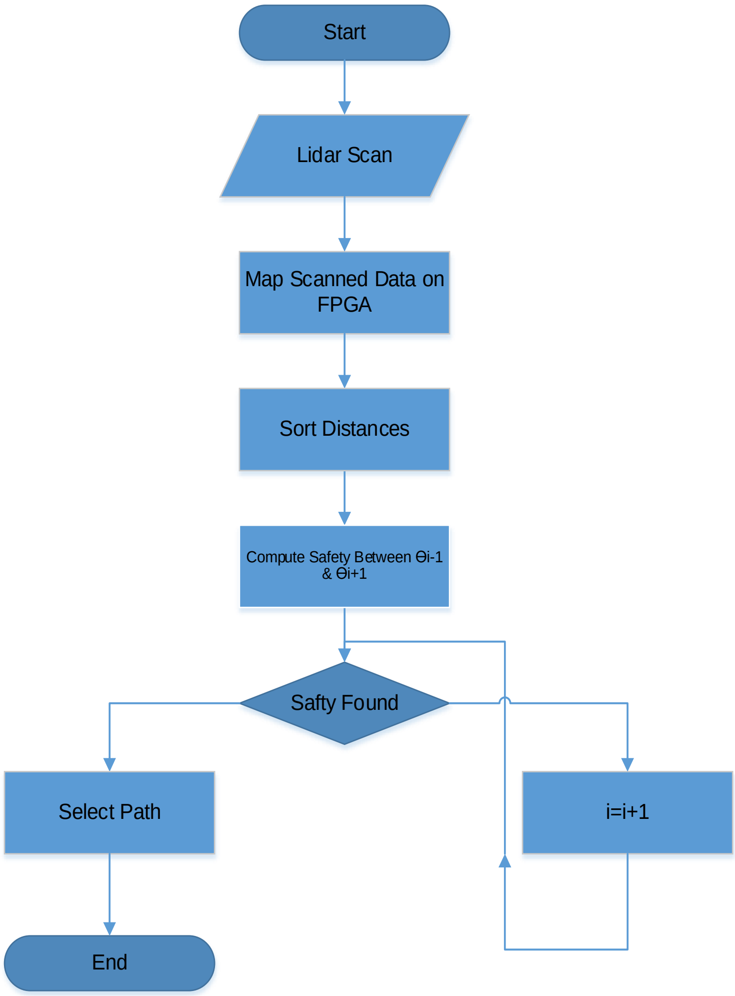

Figure 6: *Flow chart of proposed path planning algorithm*.

The proposed algorithm has been synthesized using (Xilinx Artix-7), it has been tested with a scan range of 15 degrees horizontally, and the final results for FPGA synthesizing has been recorded. Table 1 shows the usage percent for the components that has been used on the FPGA. Table 2 shows the detailed usage for every component, and Figure. 7 shows the percent of utilization for main components.

Table 1: *Usage percent for main components*.

Site Type	Used	Fixed	Available	Util%
LUT as Logic	33.549	0	63.400	52,91
F7 Muxes	1.132	0	31.700	3,57
F8 Muxes	134	0	15.850	0,84
DSPs	28	0	240	11.66
Bonded IOB	118	0	210	56.19

Table 2: *Detailed usage percent for components on FPGA.*

Ref Name	Used	Functional Category
LUT6	16.218	LUT
LUT5	13.966	LUT
LUT3	4.842	LUT
LUT4	2.894	LUT
MUXF7	1.132	MuxFx
LUT2	878	LUT
CARRY4	182	CarryLogic
MUXF8	134	MuxFx
IBUF	113	IO
DSP48E1	28	Block Arithmetic
LUT1	14	LUT
OBUF	5	IO

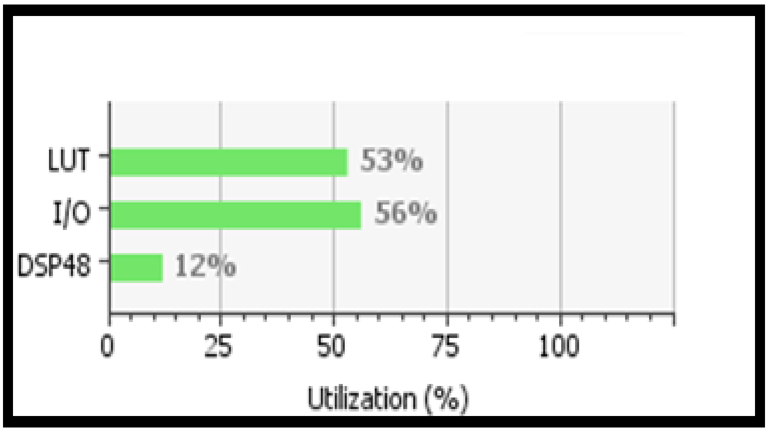

Figure 7: *Utilization percent for main components.*

4. CONCLUSIONS

In this study, we developed a fast and efficient algorithm for path planning for UAV Systems, which can be implemented on FPGA platforms and used to detect obstacles in undefined environments around them. The growth algorithm [Ladha, Kumar, Bhalla, Jain and Mittal, 2011] has been modified and implemented on an FPGA to obtain a fast and real time computing speed. We simulated the algorithm on Xilinx Vivado using Artix-7 as the target FPGA device. It has been simulated with 15 degrees scanning range to examine the accuracy of the algorithm. The input and output data has been implemented in Table 3.

TABLE 3: *Experimental inputs and output.*

Angles	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Ranges	7	30	20	99	15	12	77	15	14	33	60	89	70	27	29	73
Final Output	Optimal path = Angle (11)															

Using the above input data, the algorithm will start by sorting the data depending on the range values and then it will calculate the safety range for each path starting from the highest range, the highest range here is at angle 3 which is 99 meter, but this path is not safe because it has an obstacle on its right and left side which will make the width of this path less than 1 meter, so the algorithm will skip this path and go to the second longest path which is at angle 11, after checking the free space of angle 11 using equation 11, it will result a free space larger than 1 meter, so angle 11 will be selected as the final result.

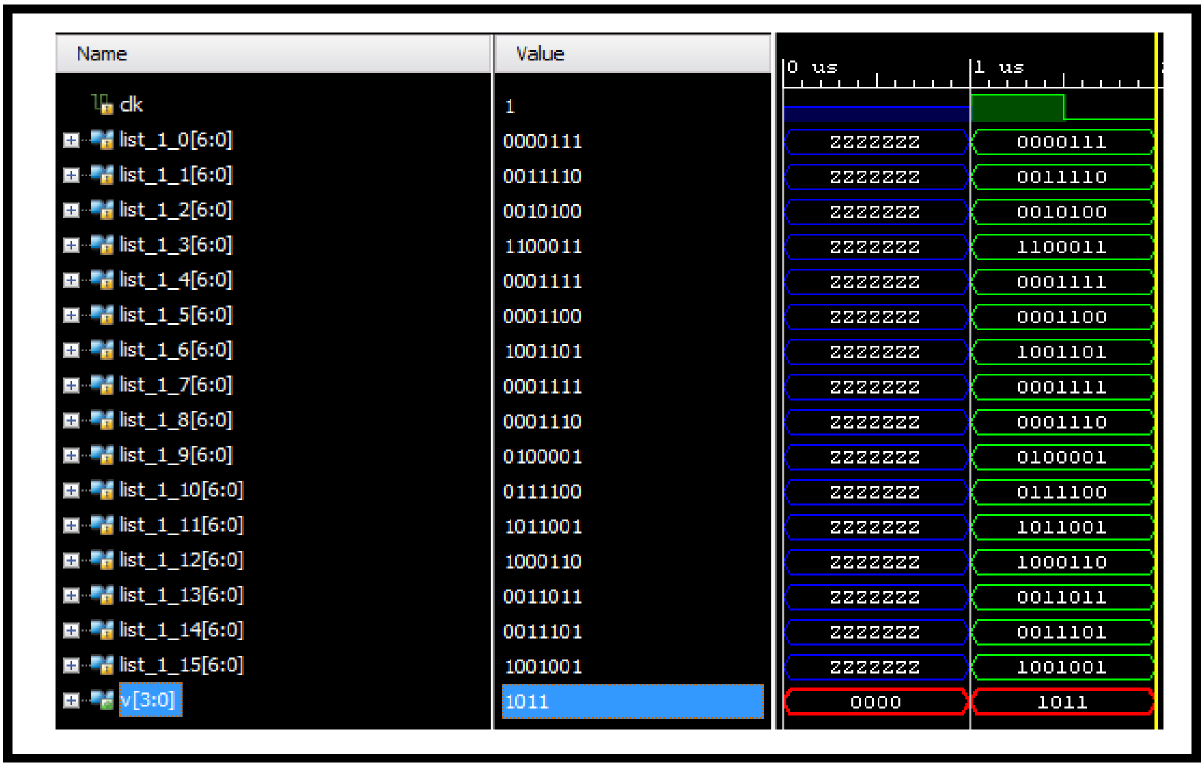

Figure 8: *FPGA synthesized waveform for experimental data.*

While running the code on MATLAB, the elapsed time was 0.004221 seconds to get the final result, but using FPGA it is getting the result in 1 clock cycle as shown in Fig. 8, by this we can conclude that real time FPGA implementation for UAV path planning is proved to be faster than software implementation by 1,899,450 times.

As a drawback to our experiment, the high efficiency of the design which is able to compute that optimal path in one clock cycle trades off the area used on FPGA, so it's hard to increase the scanning range too much. As a solution to this problem it's recommended to use more advanced FPGA rather than (Artix-7) which is in the general purpose category. We also found that junction temperature has been exceeded for our design. The problem of exceeding the junction temperature as mentioned by Xilinx can be solved by reducing the dynamic power with power optimization, lower ambient temperature and/or using heat sinks. In [Mason, 2005] an article about FPGA reliability in space-flight mentioned the junction temperature problem and as a solution for this problem, a reliable kind of FPGA (such as Actel space-flight FPGAs) should be chosen which is specifically designed for these kind of applications.

REFERENCES

Buniyamin N., Wan W.A.J., Sariff N. and Mohamad Z. (2011) “A Simple Local Path Planning Algorithm for Autonomous Mobile Robots,” International Journal of Systems Applications, Engineering & Development, Issue 2, Volume 5, 2011.

Carter, J., Schmid, K., Waters, K., Betzhold, L., Hadley, B., Mataosky, R., and Halleran, J. (2012) “An Introduction to Lidar Technology, Data, and Applications,” National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center, 2012.

Chen, H., Chang, K. and Agate, C.S. (2013) “UAV Path Planning With Tangent-Plus-Lyapunov Vector Field Guidance And Obstacle Avoidance,” IEEE Transactions on Aerospace And Electronic Systems Vol. 49, No. 2 April 2013.

Esteban, J. and Rosales, R., (2011) “Obstacle Avoidance For Mobile Robots Using Lidar Sensors,” Universitat Politècnica De Catalunya, 2011.

Ferrick, A., Fish, J., Venator, E. and Lee, G.S. (2012), “UAV Obstacle Avoidance Using Image Processing Techniques,” IEEE International Conference On Technologies for Practical Robot Applications (TePRA), PP. 73 – 78, April, 23-24 April, 2012.

Hoerner, V. and Gmbh, S. (2012) “A 2D Lidar System for Space Applications,” Schlossplatz 8, 68723 Schwetzingen, Schlossplatz 8, 68723 Schwetzingen, Germany, 2012.

Hrabar, S. (2011) “Reactive Obstacle Avoidance For Rotorcraft UAVs,” IEEE/RSJ International Conference on Intelligent Robots and Systems, September 25-30, 2011.

Ladha, S., Kumar, D. K., Bhalla, P., Jain, A. and Mittal, R.K. (2011) “Use of Lidar for Obstacle Avoidance by an Autonomous Aerial Vehicle,” Birla Institute of Technology and Science Pilani, 2011.

Mason, M. (2005), “FPGA Reliability in Space-Flight and Automotive Applications,” EEjournal, September 6, 2005.

Ogworonjo, H.C. (2011) “Machine Learning Based PID Controller Application to Autonomous Unmanned Aerial Vehicles (UAV),” Howard University, 2011.