Guidance and obstacle avoidance of MAV in uncertain urban environment

C. Kownacki
Technical University of Białystok, ul. Wiejska 45C, 15-815 Białystok, Poland

ABSTRACT

Obstacle avoidance and guidance of MAV in uncertain urban environment is the most complex and difficult part of the autonomous flight control problem. The paper presents an idea of the problem solution using a pair of miniature laser rangefinder and PID controllers. The main advantage of proposed algorithm is simplicity of its implementation in small size MAV and simultaneously it ensures low power consumption. All others algorithms, especially vision-based ones require high performance microprocessors what is an important limitation of their usage in real MAV. The proposed solution of autonomous flight control in uncertain urban environment was developed using MATLAB – SIMULINK software. The results of the performed simulations confirm the algorithm effectiveness and reliability of obstacle avoidance during a flight in streets canyons. Simulated flight paths always stop in the flight goal without collision with obstacles surrounding MAV.

1 INTRODUCTION

The main goal of autonomous flight control of MAV is to achieve the best MAV independence during flight in unknown and uncertain environment as much as it's possible. The urban uncertain environment is the most difficult and uncertain area where MAV can fly. Uncertain urban environment can be defined as a city area with unknown streets, buildings and other obstacles location and orientation. Assurance and rapid obstacle detection is expected because of high variability and variety of objects appearing near by the flight path. Waypoint navigation through urban environment should also generate the safe and optimal paths fitting maneuver possibilities in the actual MAV position. So the effectiveness of autonomous flight control is dependent from engaged navigation and control algorithms basing on utilized perception technique.

Many researcher works on improving of the effectiveness of autonomous flight control of small size MAV. Most of them are focused on vision-based systems and digital image processing [1, 3, 4, 5, 7, and 11]. Vision-based systems allow to observe a notable area around MAV and it makes the possibility to create map of environment used next by path planner. But the superior disadvantage is a complex imaging processing routines, which require efficient DSP processors that are much power consuming. In turn it impacts on available amount of power stored in on-board batteries. Increasing capacity of batteries also increases to weight of whole MAV. The image stabilization is an additional problem. Mechanical solution introduces extra on-board equipment then again electronic stabilization means more electronics. Vision-based systems are the future of autonomous flight control systems, but today's technology is unable to introduce them in real small size MAVs.

Designing of small size fixed-wing MAV requires to fulfill several restrictions concerning mostly less weight and low amount of energy which can be stored in on-board batteries. Hence it also influence on power saving feature of on-board equipment necessary to realize the flight control. In the other hand complex algorithms of autonomous guidance and obstacle avoiding need high performance microprocessor to compute safe flight path rapidly what saves time for quick MAV reaction. Most of the developed algorithms are only tested as computer simulation without performing verification in the real world [1-10]. So the challenge is the design of effectiveness autonomous flight control system which will be possible to the realization using only available equipment.

We propose an algorithm of autonomous flight control in urban environment which can be based on available equipment: two miniature laser rangefinders and advanced autopilot. All necessary computation can be run on autopilot onboard microprocessor. The paper presents algorithm effectiveness covered only by results of SIMULINK simulations. But there is a great chance to implement it in real MAV flight control system. It will be the next step of our research.

2 SYSTEM ARCHITECTUE

2.1 Miniature laser rangefinder

Two miniature laser rangefinders (LRF) are the base of proposed autonomous flight control algorithm (Figure 1). They will measure distances (ranges) between MAV and objects lying on both sides of vehicle. Controlling these distances on both sides of MAV with PID and additional logic controller allow to avoid obstacles and continuing flight in streets canyons. The great advantages of miniature laser rangefinder are its features which well predisposes it to onboard usage of small size MAVs. Features of miniature laser rangefinder are presented in Table 1.

Feature	**Value**
Weight	26 g
Power rating	<400mW
Size	1,25"x1,5"x1,6"
Pulse repetition	500Hz
Resolution	<0,2 m

· Email address: cezarkw@poczta.onet.pl

Range	~0.1m to >100m
Divergence angle	10x10 mrad

Table 1: Features of miniature laser rangefinder MRL100 [11]

The most important features are small size, weight, power efficiency and high repetition rate. Range limitation fits also specific conditions of urban environment flight.

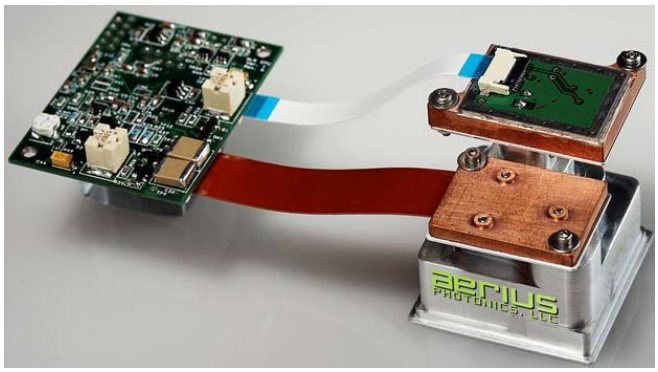

Figure 1: Miniature laser rangefinder MRL100 from AERIUS.

Configuration of miniature laser rangefinders is a significant aspect of the system reliability and usability. In our opinion the best choice of sensors configuration is as follows: both laser light beams are placed tangent to the MAV plane and there is a gape angle between them. So two laser beams will create "V" shape (Figure 2).

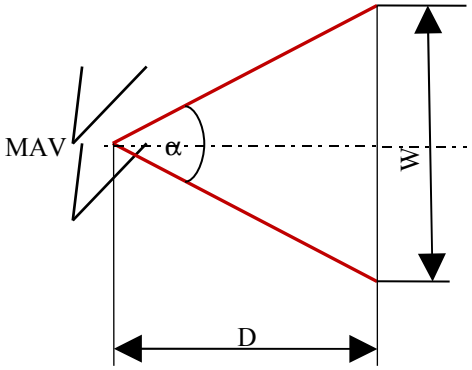

Figure 2: The "V" shape of configuration of miniature laser rangefinders: α - angle between laser beams, D – minimal distance ensuring obstacle avoidance, W – minimal width of obstacle.

The configuration of sensors satisfies both aspects of flight in urban environment: flight in streets canyons and obstacle avoidance. Angle between laser beams is given by formula:

$$(1) \quad \alpha = 2 \cdot \arctan\left(\frac{W}{2 \cdot D}\right)$$

where: W – minimal width of obstacle (building) which can be found in the environment,
D – minimal distance ensuring obstacle avoidance.

Values of parameters *D* and *W* should be determined for each MAV separately, because they depend on MAV size and motion possibilities. If we assume the minimal distance *D* is 10 m and minimal width of obstacle *W* is also 10 m, α angle will be about 53^0. These values were also used for algorithm simulations. Detailed research on those parameter should be done during the algorithm implementation and it's not a part of the paper.

Laser rangefinders are fixed to the MAV body. So measured signals should be independent from MAV space orientation, what is necessary to determine the real distance to obstacle (building wall). Hence coordinates of obstacle must be transformed from body frame to NED frame (North East Down) (Figure 3). Required actual roll and pitch angles are available from autopilot unit.

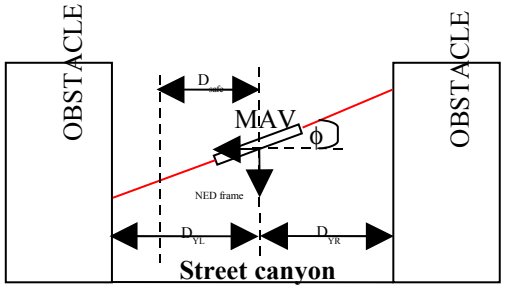

Figure 3: MAV flight in street canyon, ϕ - roll angle, D_{YL} – real *y* coordinate of obstacle on the left, D_{YR} – real *y* coordinate of obstacle on the right. D_{safe} – distance to obstacle, when PIDs start to control MAV.

2.2 *Overview of algorithm structure*

In the case of laser rangefinders usage in the autonomous flight control system, adequate control algorithm should be introduced. We propose a design of the algorithm using only PID controllers with an extra logic controller and signal conditioning. The algorithm will also cooperate with autopilot unit to realize waypoint navigation simultaneously. Logic controller will decide about actual priority of the flight goals i.e. waypoint navigation and obstacle avoidance with feature of the flight in streets canyons.

The diagram of designed autonomous flight control system (algorithm) is presented in Figure 4.

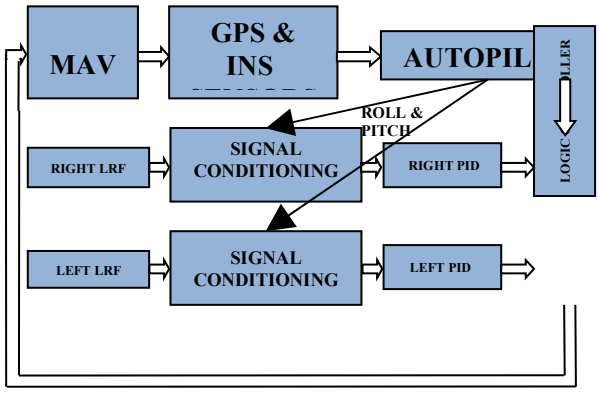

Figure 4: The diagram of designed autonomous flight control system, LRF – laser rangefinder

GPS and INS sensors are built in autopilot unit and they are mostly used for standard flight control and waypoint navigation. Rest part of the algorithm is responsible for the obstacle avoidance including flight in streets canyons.

Laser rangefinders should measure distance between MAV and buildings during flight in streets canyons. But there are many smaller objects i.e. street lights, traffic lights, trees,

road signs that can't be properly detected by the algorithm. This is a disadvantage. But all small objects will be detected as peaks of distance value or signal disturbances, what can affect MAV behavior negatively. The flight can be even unstable or nervous. So we decided to filter out these peaks making assumption that the MAV will be able to avoid only buildings and fly in streets canyons at specified higher altitude.

2.3 *Signal conditioning*

As it was mentioned, applying signal conditioning is a necessary part of the algorithm. It can be divided into several different steps (Figure 5).

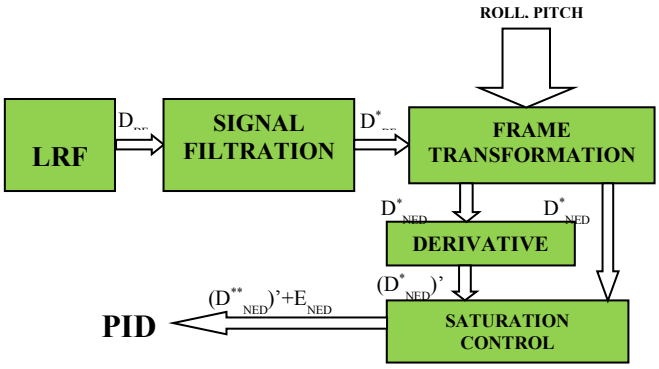

Figure 5: The diagram of applied signal conditioning, D_{BF}- signal of distance to obstacle (coordinates) in body frame, D^*_{BF} – filtered signal of distance to obstacle (coordinates) in body frame, D^*_{NED} - signal of distance to obstacle (coordinates) in NED frame, $(D^*_{NED})'$ – $d(D^*_{NED})/dt$ - speed of obstacle approach, E_{NED} + $(D^{**}_{NED})'$ – PIDs input signals.

The first step of signal conditioning is signal filtration that eliminates disturbances introduced by small objects in the urban environment. Low pass filter or better Kalman filter can be used here to achieve the best filtration results.
The next step is frame transformation. Obstacle coordinates in body frame can be calculated from the signal using value of angle α (Equation 1).

$$(2)\quad \begin{pmatrix} x_{BF} \\ y_{BF} \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right) \end{pmatrix} \cdot D^*_{BF}$$

where: D^*_{BF} – filtered distance signal (body frame),
α – angle between laser beams,
x_{BF}, y_{BF} – obstacle coordinates in body frame.

Body frame coordinates must be transformed into NED (North East Down) frame to determine real distance between obstacle and MAV. Because distance measurement should be independent from yaw angle and heading angle, the formula of transformation is as follows:

$$(3)\quad \begin{pmatrix} x_{NED} \\ y_{NED} \\ z_{NED} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \cdot \begin{pmatrix} \cos\theta & 0 & -\sin\vartheta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \cdot \begin{pmatrix} x_{BF} \\ y_{BF} \\ 0 \end{pmatrix}$$

where: x_{BF}, y_{BF} – body frame coordinates,
x_{NED}, y_{NED} – NED frame coordinates,
ϕ – roll angle,
θ – pitch angle.

Having NED frame coordinates we can compute distance D^*_{NED} to obstacle expressed in NED frame.

$$(4)\quad D^*_{NED} = \sqrt{\left(x^2_{NED} + y^2_{NED}\right)}$$

Determining derivative of D^*_{NED} makes the algorithm sensitive to velocity and direction of obstacle approach. If D^*_{NED} equals zero, it will mean that the MAV is flying parallel to the obstacle. The highest value (V/cos (α)) of the velocity of obstacle approach means that the MAV is flying normally to the obstacle and collision probability is very high.
The last step of signal conditioning is saturation control that allows to ignore safe distances above specified threshold value D_{safe} and also ignore receding obstacles – positive value of $(D^*_{NED})'$ (Equation 5). For these cases corresponded element of the right or the left PID input signal equals zero.

$$(5)\quad \begin{cases} E_{NED} = D^*_{NED} - D_{safe} & D^*_{NED} - D_{safe} < 0 \\ E_{NED} = 0 & D^*_{NED} - D_{safe} \geq 0 \\ \frac{d(D^{**}_{NED})}{dt} = \frac{d(D^*_{NED})}{dt} & \frac{d(D^*_{NED})}{dt} < 0 \\ \frac{d(D^{**}_{NED})}{dt} = 0 & \frac{d(D^*_{NED})}{dt} \geq 0 \end{cases}$$

Where: E_{NED} – first element of PID input signal – difference between actual distance to obstacle and safe distance D_{safe},
D_{safe} – distance to obstacle (normal to the flight direction), when PIDs start to control MAV,
D^*_{NED} – distance to obstacle expressed in NED frame,
$d(D^{**}_{NED})/dt$ – speed of obstacle approach - second element of PID input signal.

The sum of E_{NED} and velocity of obstacle approach $(D^{**}_{NED})'$ is defined as the input signal of PID controllers. Such solution would make the algorithm sensitive only to the nearest obstacle that is approaching forward to the MAV.

2.4 *The logic controller*

The role of the logic controller is to specify what goal of flight mission should be consider actually. Hence to realize that it needs output signals of both PIDs, output signal of autopilot and both D^*_{NED} distances. The algorithm of logic controller works as a kind of switching element. It switches MAV control between PIDs and autopilot dependently from actual state of its inputs. The direction of fixed-wing MAV flight is controlled by roll angle command. The logic controller algorithm is described by Equation 6.

$$(6)\quad \phi = \begin{cases} \phi_{autopilot} & ; \phi_{PIDright} = 0 \\ \phi_{PIDright} + \phi_{PIDleft} & ; \phi_{PIDright} \neq 0 \cap \phi_{PIDleft} \neq 0 \cap \phi_{PIDright} \neq \\ \phi_{PIDright} & ; \left(\phi_{PIDright} \neq 0 \cap \phi_{PIDleft} = 0\right) \cup \\ & \left(\phi_{PIDright} \neq 0 \cap \phi_{PIDleft} \neq 0 \cap \phi_{PIDright} - \phi_{PIDleft} \right. \\ & \left. D^{**}_{NEDleft} > D^{**}_{NEDright}\right) \\ \phi_{PIDleft} & ; \left(\phi_{PIDright} = 0 \cap \phi_{PIDleft} \neq 0\right) \cup \\ & \left(\phi_{PIDright} \neq 0 \cap \phi_{PIDleft} \neq 0 \cap \phi_{PIDright} = \phi_{PIDleft} \cap \right. \\ & \left. D^{**}_{NEDleft} \leq D^{**}_{NEDright}\right) \end{cases}$$

where: ϕ^C – commanded MAV roll angle,
$\phi_{autopilot}$ – autopilot controlled roll angle,
$\phi_{PIDleft}$ – left PID controlled roll angle,
$\phi_{PIDright}$ – right PID controlled roll angle,

If outputs of both PIDs are equal zero, only the autopilot will control the heading angle of MAV flight. In the case when both of them are not equal zero and additionally they have different values, commanded roll angle will be sum of both PIDs outputs. But at the moment it's necessary to point on important fact that the right and the left PIDs are differing only with opposite sign of output signals. Another case is when only the left PID output is zero or both PIDs outputs have different values not equal zero and simultaneously the left D^*_{NED} is greater than the right D^*_{NED}. In this case the right PID output will control the roll angle. The last case is opposite to pervious. When only the right PID output is zero or both PIDs outputs have different values not equal zero and simultaneously the left D^*_{NED} is smaller than the right D^*_{NED}, the left PID output commands the roll angle. One more important notice is that PIDs outputs can equal zero because of applying saturation control (Equation 5).
In summary the logic controller changes the way of MAV flight control accordingly to actual state of environment. Of course it can be replaced by other types of controllers, for example by more advanced fuzzy logic controller.

3 THE ALGORITHM SIMULATION

3.1 Fixed-wing MAV model

The simulations of proposed algorithm were performed in MATLAB – SIMULINK. Hence a model of MAV is required to simulate flight paths. We used fixed-wing MAV model defined by following equations [8]:

$$\dot{x} = V \cdot \cos\psi \ ,$$
$$\dot{y} = V \cdot \sin\psi \ ,$$
$$\dot{\psi} = \frac{g}{V} \cdot \tan\phi \ , \quad (7)$$
$$\dot{V} = \alpha_V (V^C - V),$$
$$\dot{\phi} = \alpha_\phi (\phi^C - \phi).$$

Where: x, y – MAV actual coordinates,
V – MAV actual velocity,
V^C – MAV commanded velocity,
ψ – MAV actual heading/track angle,
ϕ – actual roll angle,
ϕ^C – commanded roll angle,
α_ϕ, α_V – time constants of MAV dynamics,
g – gravity constant.

The model assumes that MAV flight is controlled by two motion properties: flight velocity and roll angle. Roll angle makes MAV to change its flight direction. In our simulations we assumed that the flight velocity V^C is constant and the algorithm controls the commanded roll angle ϕ^C only.

3.2 Autopilot model

A model of autopilot is also required to run simulation of the algorithm. The autopilot was modeled by a PID controller which controls MAV flight direction (heading/track angle) by minimizing actual bearing angle pointing on the next waypoint. So the model of autopilot can be formulated as follows:

$$\psi_{err} = \psi_{WP} - \psi \ ,$$
$$\psi_{WP} = \arctan\left(\frac{y_{WP} - y}{x_{WP} - x}\right) \quad (8)$$
$$\phi_{autopilot} = PID \cdot \psi_{err}$$

Where: ψ_{err} – track angle error,
ψ – actual MAV heading /track angle,
ψ_{WP} – actual bearing angle pointing on the next waypoint WP,
x, y – actual coordinates of MAV,
x_{WP}, y_{WP} – coordinates of the next waypoint,
PID – transitions function of PID controller,
$\phi_{autopilot}$ – roll angle given by autopilot.

The output of autopilot is the command of roll angle that is only used when the logic controller detects zero values at the left and the right PIDs outputs (Figure 4).
The SIMULINK model of the whole algorithm is presented in figure below.

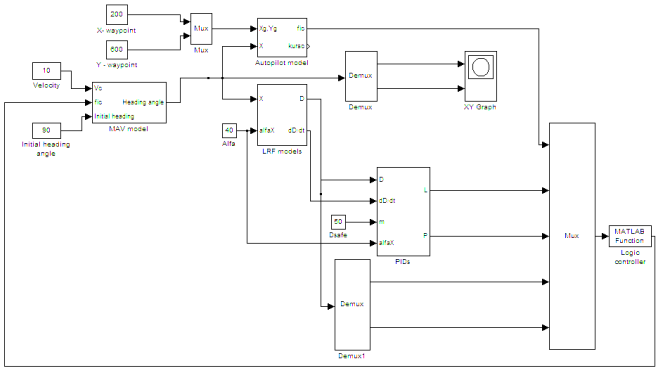

Figure 6: The SIMULINK model of proposed autonomous flight control algorithm.

Synthetic maps representing urban environments used in flight simulations are prepared as graphic files in bitmap format. Next they were loaded into MATLAB/SIMULINK workspace. Because of that they are only 2D maps, distance signals filtration and frame transformation weren't included in the SIMULINK model.

4 RESULTS

Four maps of environment characterizing different flight situation are prepared to prove the algorithm effectiveness. Two of them are concerning possibilities of the MAV flight in streets canyons. Next two present the MAV obstacle avoidance. In each simulation the MAV started its flight in the start point SP with given initial heading/track angle and

the flight goal was the next waypoint WP. Results of simulations are presented in Figures 7-10. Obstacles are displayed as areas in black color. Boundaries of the maps are treated as obstacles so MAV should also avoid them. Following values of parameters of MAV and simulation model were used: V^C=10 m/s, D_{safe}=50 m, $\alpha=40^0$, alfa_V=0.1, alfa_phi=0.1.

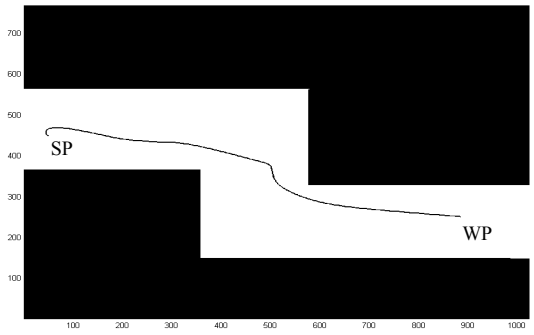

Figure 7: The MAV flight path in the first simulation of street canyon fight mission. Initial heading / track angle – 180^0, starting point SP – x=50,y=450, the next waypoint WP – x=900,y=250.

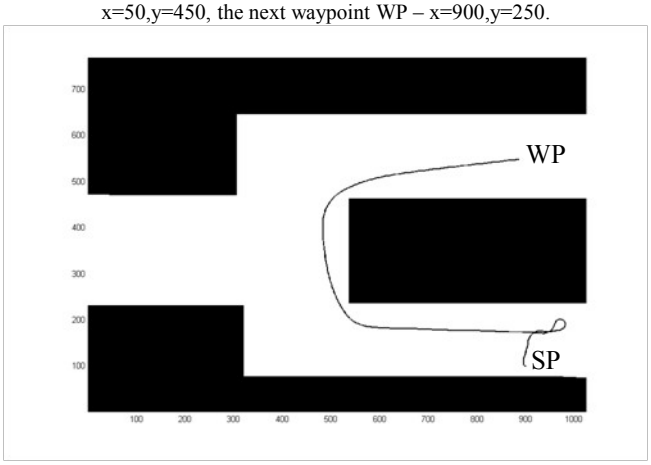

Figure 8: The MAV flight path in the second simulation of street canyon fight mission. Initial heading / track angle – 180^0, starting point SP – x=900,y=100, the next waypoint WP – x=900,y=550.

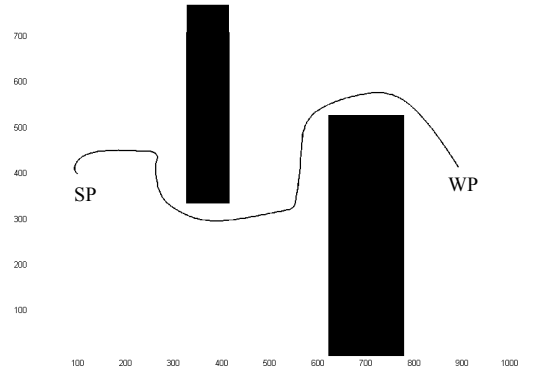

Figure 9: The MAV flight path in the first simulation of obstacle avoidance. Initial heading / track angle – 180^0, starting point SP – x=100,y=400, the next waypoint WP – x=900,y=400.

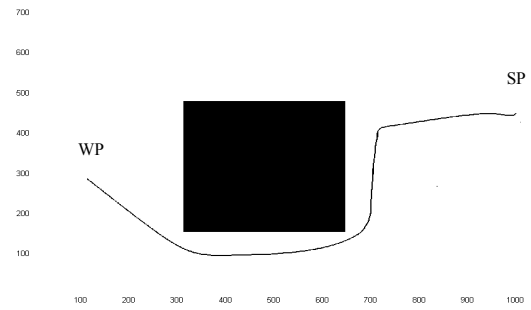

Figure 10: The MAV flight path in the second simulation of obstacle avoidance. Initial heading / track angle – -90^0, starting point SP – x=1000,y=450, the next waypoint WP – x=100,y=300.

If we take a look on results and presented flight paths, we will notice that the MAV always flies holding a safe distance to obstacles. Of course there is an important assumption that sizes of all obstacles are much bigger than MAV size. So in all figures the MAV are displayed by a single point on the flight path.

Other obvious notice is that all flight paths are not the shortest path of possible flight (for example in Figure 10). But the introducing of optimal flight paths requires using additional flight planner. As it was mention above in the introduction, the flight planner would increase complexity of the autonomous flight control algorithm. And that's opposite to the paper aim.

5 Conclusion

Figures 7-10 present results of four different simulations of the autonomous flight control algorithm. They are clearly proving its effectiveness. In each case the MAV flight mission succeeded in reaching the next waypoint without collision with surrounding obstacles. Results displayed in Figures 7-8 present also possibilities of the MAV flight in streets canyons. This is the important feature, which is essential for the MAV flight in uncertain urban environment. The algorithm meets also requirements of obstacle avoidance what was shown in Figures 9-10.

In summary it can be clearly underscored that the aim of the paper was achieved. The proposed autonomous flight control algorithm is effective, reliable and simultaneously it is not too complex. So it wouldn't be impossible to realize by using advanced autopilots and small size MAVs. The advantage of the algorithm is also real-time signal processing. Others methods which use path planners always contain some delay due to much computation time.

The algorithm implementation will be the next step of our research. PID controllers tuning seems to be the most important and laborious task due to difficulties in simulating small size MAV dynamics. So the research and all examinations should be executed with the real object utilization

Acknowledgment

The research was funded by finances of Polish Ministry of Science and Higher Education for science in 2008-2010 as developmental project No. O R00 0059 06.

References

[1] Eric W. Frew, Raja Sengupta: Obstacle Avoidance with Sensor Uncertainty for Small Unmanned Aircraft, *43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[2] Eric W. Frew, Stephen Spry, Adam Howell, J. Karl Hedrick, Raja Sengupta: Flight Demonstrations of Self-Directed Collaborative Navigation of Small Unmanned Aircraft, *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop, & Exhibit*, Chicago, IL, September 2004.

[3] Eric W. Frew et. al., Vision-Based Road Following Using a Small Autonomous Aircraft, *IEEE Aerospace Conference*, Big Sky, MT, March 2004.

[4] Eric W. Frew et. al., Stereo-Vision-Based Control of a Small Autonomous Aircraft Following a Road, *Second Annual Swarming Conference*, Crystal City, MD, June 2004.

[5] Eric W. Frew, Raja Sengupta: A Small Unmanned Aircraft Navigating through a Field of Uncertain Obstacles.

[6] Eric W. Frew, Jack Langelan: Receding Time Horizon Control for Passive, Non-cooperative UAV See-and-Avoid, *IEEE International Conference on Robotics and Automation,* Barcelona, Spain, April 2005.

[7] Zhihai He, Ram Venkataraman Iyer, Phillip R. Chandler: Vision-based UAV flight control and obstacle avoidance, *IEEE Automatic Control Conference* 2006.

[8] Jeffery B. Saunders, Brandon Call, Andrew Curtis, Randal W. Beard, Timothy W. McLain, Static and Dynamic Obstacle Avoidance in Miniature Air Vehicles, *Infotech@Aerospace*, *AIAA*, 26 - 29 September 2005, Arlington, Virginia.

[9] E. Anderson. Extremal control and unmanned air vehicle trajectory generation. Master's thesis, Brigham Young University, April 2002.

[10] Tao Dong, X. H. Liao, R. Zhang, Zhao Sun and Y. D. Song: Path Tracking and Obstacle Avoidance of UAVs - Fuzzy Logic Approach, FUZZ '05. The 14th IEEE International Conference on Fuzzy Systems, Reno, 2005.

[11] www.aeriusphotonics.com

[12] A. Lilienthal, T. Duckett. Experimental Analysis of Smelling Braitenberg Vehicles, ICAR 2003.

[13] D. Watman, T. Furukawa. A Visualization System for Analysis of Micro Aerial Vehicle Scaled Flapping Wings, J Intell Robot Syst No. 51, 2008.

[14] J. Serres, D. Dray, F. Ruffier, N. Franceschini. A vision-based autopilot for a miniature air vehicle:joint speed control and lateral obstacle avoidance. Auton Robot No. 25, 2008.