

Towards An Autonomous Landing System in Presence of Uncertain Obstacles in Indoor Environments

Thi Thoa Mac^{1*}, Cosmin Copot², Ricardo Cajo^{3,4,5}

¹School of Mechanical Engineering, Hanoi University of Science and Technology, No. 1, Dai Co Viet, 100000 Hanoi, Vietnam; thoa.mac@hust.edu.vn

²Department of Electromechanics, Op3Mech, University of Antwerp, Groenenborgerlaan 171, 2020 Antwerp, Belgium; cosmin.copot@uantwerpen.be

³Research lab on Dynamical Systems and Control, Ghent University, Tech Lane Science Park 125, 9052 Ghent, Belgium; RicardoAlfredo.CajoDiaz@UGent.be

⁴Facultad de Ingeniería en Electricidad y Computación, Escuela Superior Politécnica del Litoral, ESPOL, Campus Gustavo Galindo Km 30.5 Vía Perimetral, P.O. Box 09-01-5863, 090150 Guayaquil, Ecuador

⁵Core Lab EEDT-Energy Efficient Drive Trains, Flanders Make, 9052 Ghent, Belgium

*Corresponding author

Abstract: The landing task is fundamental to Micro air vehicles (MAVs) when attempting to land in an unpredictable environment (e.g., presence of static obstacles or moving obstacles). The MAV should immediately detect the environment through its sensors and decide its actions for landing. This paper addresses the problem of the autonomous landing approach of a commercial AR. Drone 2.0 in presence of uncertain obstacles in an indoor environment. A localization methodology to estimate the drone's pose based on the sensor fusion techniques which fuses IMU and Poxyz signals is proposed. In addition, a vision-based approach to detect and estimate the velocity, position of the moving obstacle in the drone's working environment is presented. To control the drone landing accurately, a cascade control based on an Accelerated Particle Swarm Optimization algorithm (APSO) is designed. The simulation and experimental results demonstrate that the obtained model is appropriate for the measured data.

Keywords: Micro air vehicles (MAVs); autonomous landing; obstacle avoidance; dynamics obstacle' velocity estimation; sensor fusion; optimal control

1 Introduction

Autonomous navigation is an essential requirement for intelligent mobile platforms [1], [2], [3]. Nowadays, intelligent autonomous navigation at low or super low-altitude is an important requirement for aircraft that must complete missions close to the ground, and such ability is extensively desired. Search and rescue operations allow benefit points at low altitudes. Similarly, Unmanned Aerial Vehicles (UAVs) performing reconnaissance, the military must fly at low altitude in the presence of known and unknown obstacles. Until now, UAVs have to land on a prepared position with prior knowledge of obstacle-free trajectories. In order to land on a location at which the safety requirement is unknown (e.g., presence of static obstacles or moving obstacles), the UAV should immediately detect the environment through its sensors and decide its actions for landing [4].

An interesting survey of advances in guidance, navigation, and control of UAV systems is described in [5]. The paper first presents the main research groups, then the development of frameworks and algorithms. These algorithms and systems are classified into different categories based on the autonomy level. In [6], a feasible trajectory for landing is investigated to make the decision during flight time to guarantee safety. In addition, the authors proposed the formula to reduce computation time. A required translation of the operating bounds of the aircraft is the disadvantage of this work. In [7], landing on a moving target has been simulated. The optimal trajectory is determined based on the Variational Hamiltonian and Euler-Lagrange equations. The kinematic model of the helicopter is used to derive an optimal controller to track and land on an arbitrary moving target. However, this model is a simplified version, therefore, there are some drawbacks in this research.

The Autonomous landing of a UAV on static and mobile platforms in absence of GPS is presented in [8]. Among different strategies, electro-optical sensors have been the most popular modality for landing site evaluation [9]. In [10], a vision system is used for control, terrain reconstruction, and tracking. Another study is in [11], where the UAV altitude is obtained by fusing GPS signals with stereo vision. Another safe landing area system based on multiple impulse-radio ultra-wide-band (IR-UWB) radar is proposed in [12]. In which, the detecting of ground property methodologies (slope and roughness) and obstacles using IR-UWB are studied.

Light Detection and Ranging (LIDAR) sensors have been broadly studied for safe landing-area determination for small helicopters [13], [14] and fixed-wing airplanes [15] while Global Positioning System (GPS) is chosen for navigation of aircraft [16], [17], [18]. An intelligent system, based on image segmentation procedures to recognize the relative orientation of the UAV and a platform, is presented in [19]. In [20], a robust adaptive nonlinear guidance and control design based on neuron-adaptive design philosophy is presented for the robust landing operation. The numerous simulations using the six-degrees-of-freedom nonlinear model of a prototype UAV is used to verify the proposed methodology. In which,

wind disturbances and ground effects are investigated. In [21], a biomimetic system is proposed that determines terrain appearances, such as large obstacles and precipitous slopes by using a monocular camera. In [23], a nonlinear controller model for take-off and landing purpose (VTOL) of a UAV is described using measurement optical flow. This method enables the UAV to hover and land on a moving platform, such as the deck of a sea-going vessel. Two different assignments are concerned in that study: (1) the stabilization of the UAV; (2) the regulation of automatic landing of the UAV onto a moving platform.

A dynamic target tracking and obstacle avoidance of a UAV is studied in [24]. A method of cooperation between two UAVs at different high and low altitudes is presented in [25] for autonomous navigation and landing. Based on the high flexibility and extensive vision, the higher UAV measures the position of the lower UAV and controls it to track the marker and land on it. In [26], a pose estimation process of a UAV using parallel image technique is studied. The system performs the capabilities of a high-performance Graphics Processing Unit/ Central Processing Unit (CPU/GPU) embedded system to provide safely autonomous take-off and landing processes.

In this study, the UAV landing is performed in presence of uncertain obstacles using sensor fusion which fuses Pozyx and onboard sensors' signals. In order to use Pozyx, some modifications had to be made to the drone which caused an imbalance at hovering. Additionally, we developed an algorithm to estimate the moving obstacle's velocity based on the onboard vision system. This velocity information can help autonomous UAV to interpret for the landing task in complex situations. The main contributions of this article are:

- the development of an autonomous landing system for a UAV.
- the design and implementation of sensor fusion technique.
- the estimation of moving obstacle's velocity based on vision approach.
- the design and implementation of a cascade control APSO algorithm for the drone landing task.

This paper is organized as follows. Section II provides a description of the setup system and localization method using onboard sensors, Pozyx, and sensor fusion method. The obstacle detection and landing algorithm are briefly outlined in Section III. Section IV presents the PID controller based on an APSO algorithm using transfer functions obtained by the identification process. Section V demonstrates simulations and experiment results based on the proposed approach. Section VI presents the conclusions and future works.

2 System Setup and Localization Method

In this section, the system setup and proposed localization based on sensor fusion are presented. Assume that the UAV performs the landing task for the victim's rescue application in presence of uncertain static and moving obstacles (Lego Robot) with the designed target (victim) position (marked as the blue star) as shown in Fig. 1.

Our approach consists of three major components running on a laptop (ground station) connected to the UAV via wireless communication. The first component is Pozyx localization, which allows the UAV to determine its coordinates in a working space. The second component is an inertial measurement unit (IMU) data process, which delivers and receives signals between the quadrotor and the ground station. To calculate a more accurate position of the UAV, the sensor fusion method is implemented. The last component is a cascade control based on the APSO algorithm. The ground station is a laptop Intel Core i7-6600U, 2.81Hz, 64-bit operating system.

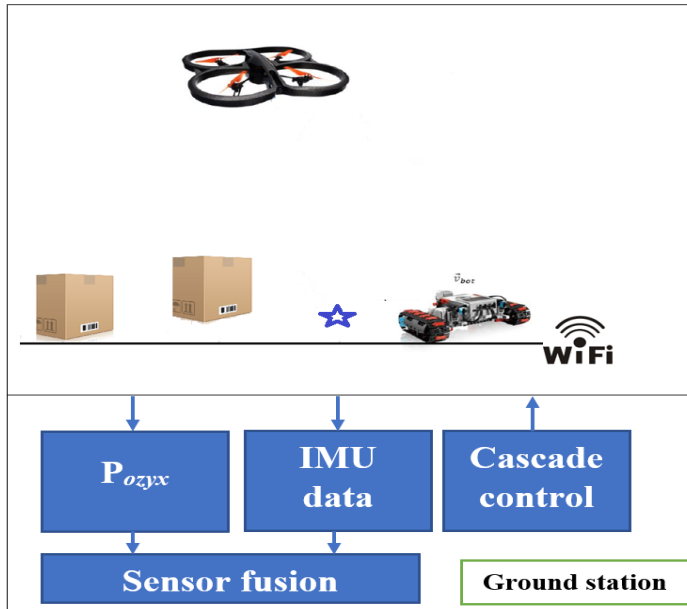


Figure 1

The proposed approach

In outdoor applications, GPS usually provides a reasonable estimation however, it is not an effective tool for indoor applications. There are several options for indoor position estimation such as visual navigation, the onboard IMU data, or extra sensors. This section concerns the state estimation of the UAV in an indoor environment using IMU, Pozyx, and sensor fusion techniques.

2.1 IMU Data Processing

The IMU measures the roll, pitch, and yaw angles. The velocity estimations are obtained by fusing the information from a 3-axis gyroscope with the information from a 3-axis accelerometer and a magnetometer. Position estimations can then be obtained by integrating the velocities. The x position of the drone is calculated in Equation 1.

$$x_k = x_{k-1} + \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} V \Delta t \quad (1)$$

where: V is the velocity of the drone; x_k, x_{k-1} are position of the drone in x -direction at sample $k, k-1$; Δt is sample time.

The y position is calculated in a similar approach. A better estimation of the height can be obtained by fusing the velocity in z -direction with the information from the ultrasonic sensor.

2.2 Pozyx

In this study, the drone is equipped with *Pozyx* as shown in Figure 2. The position estimation is obtained accurately by using ultra-wideband technology. Please refer to [27].



Figure 2

Pozyx on AR.Drone 2.0

Three anchors are placed in a room. The anchors should not be placed on one line or one plane. The absolute position of these anchors needs to be known in advance. By measuring the distance between the *Pozyx* tag and each anchor, the position of the *Pozyx* tag (UAV position) can be estimated. These were filtered out by defining a maximum allowed deviation of the current position compared to the previous one. When the deviation exceeds the threshold value, the current position is set equal to the previous one. The *Pozyx* offers a robust position estimation. However, this signal is noisy so that it is not recommended to use the *Pozyx* as the only source of information. The results obtained by unfiltered and filtered *Pozyx* are shown in Figure 3.

2.3 Sensor Fusion

To combine the advantage of IMU and *Pozyx*, there is opted to use a Kalman filter to fuse these signals. The Kalman filter combines the information of different sensors and the expected state from the physical model to estimate the state, $(x; y; z; v_x; v_y; v_z)^T$

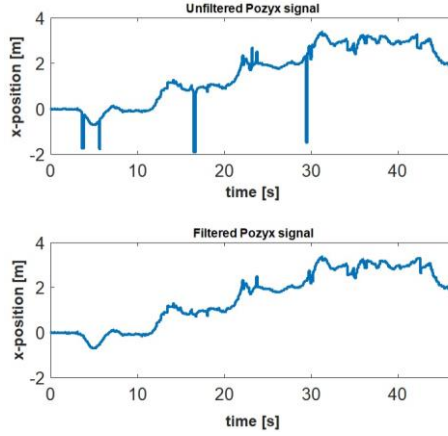


Figure 3

Unfiltered compared to Filtered *Pozyx* signal

There are two phases in the Kalman filter: the prediction phase and the update phase. In the first phase, the Kalman filter estimates the position and its uncertainty based on the physical model.

$$\hat{x}_k = A\hat{x}_{k-1} \quad (2)$$

$$P_k = AP_{k-1}A^T + Q \quad (3)$$

where

$\hat{x}_k, \hat{x}_{k-1}$ are estimated current state and previous state respectively;

A is the state-transition model;

Q is the covariance of process noise;

P_k, P_{k-1} are current and previous estimate co-variance.

In the second phase, the observed measurements are implemented to find the new best estimate as the following equations.

$$K = P_k C^T (C P_k C^T + R)^{-1} \quad (4)$$

$$\hat{x}_k = \hat{x}_k + K(z_k - C\hat{x}_k) \quad (5)$$

$$P_k = P_{k-1} - K C P_k \quad (6)$$

Where

K is the Kalman gain;

C is the observation model which converts the true state space into the observed space;

R is the co-variance of observation noise;

z_k is the measurement signal.

The transition matrix A was chosen as in Equation 7. The variable dt is equal to the mean execution time of the loop. With the used sensors, it is possible to directly measure the states; therefore, the measurement matrix C is equal to a unity matrix.

$$A = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

In order to design the Kalman filter, the process noise covariance Q and the measurement noise covariance R are added. The co-variances of the *Pozyx* were then tuned manually until the desired result was obtained.

$$Q = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 \end{bmatrix} \quad (9)$$

$$R = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.05 \end{bmatrix} \quad (10)$$

Figure 4 shows an estimation of the position with the Kalman filter, *Pozyx*, and the IMU. In this experiment, the drone starts at $x \approx 0$ m, then goes around 21 seconds to $x \approx 1$ m, and finally goes around 30 seconds to $x \approx 2$ m. As the IMU estimation drift very quickly, it is not possible to use it as the pure reference

resource. The disadvantage of *Pozyx* is its noise. The proposed sensor fusion method which combines the advantages of both signals is a better option with robust estimation without noise.

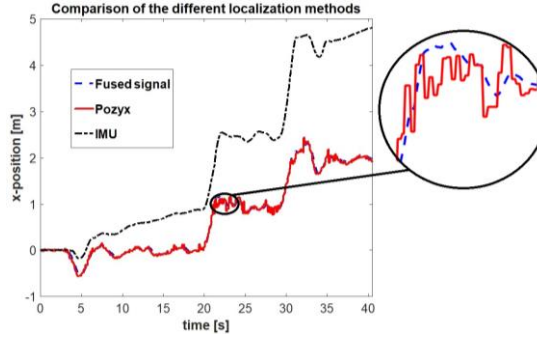


Figure 4

The comparison of the different localization methods

3 Obstacle Detection and Landing

Obstacle detection is one of the key functionalities of a fully autonomous drone. For autonomous landing applications, both static obstacles and moving obstacles should be detected.

3.1 The Static Obstacle Detection

In order to detect static obstacles, a solution based on distance measurement is proposed. The algorithm uses the difference between absolute height (the distance to the ground level h_a) and relative height (the distance to the nearest obstacle h_r) to detect an obstacle. A graphical representation of the heights can be found in Figure 5.

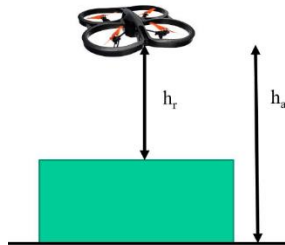


Figure 5

The static obstacle detection based on the absolute and relative heights

3.2 The Moving Obstacle Detection

The idea behind the detection of moving obstacles is to interpret complex situations and to make appropriate decisions of the UAV based on this additional information. In the experiment, a Lego Mindstorms EV3 was used to represent a moving obstacle. The moving obstacle is marked with red paper. Based on color detection, it is possible to extract the coordinates of the obstacle in the image frame. These coordinates need to be converted to real-world coordinates to obtain the obstacle's position/velocity information. Figure 6 presents moving obstacle detection using color segmentation. The images were acquired by using the bottom camera of the drone.

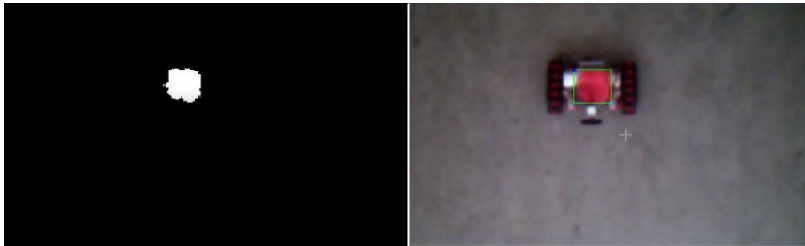


Figure 6

Color-based tracking moving obstacle

In order to convert obstacle positions from an image coordinate to a world coordinate, the calculation is implemented based on the diagram as shown in Figure 7. In which, M is the center point of the received image. P is the position of the Lego robot in the image frame. AOV_x is the angle of field of view and D presents the bottom camera of the AR.Drone 2.0. O is the projection point of D in the horizontal plane. W is the intersection point between DP and the horizontal plane. The following formulas are applied.

$$|DM| = \frac{|KM|}{\tan(AOV_x/2)} \quad (11)$$

$$\alpha = \frac{|PM|}{|DM|} \quad (12)$$

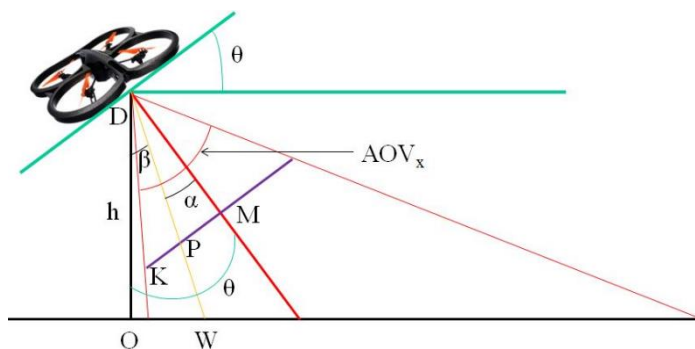


Figure 7

The conversion from the image coordinates to the real-world coordinates

$$\beta = \begin{cases} \theta + \alpha & \text{if } X_P > X_M \\ \theta - \alpha & \text{if } X_P < X_M \end{cases} \quad (13)$$

We define α' as:

$$\alpha' = \text{artan}(\frac{X_P - X_M}{|DM|}) \quad (14)$$

We have:

$$\beta = \theta + \alpha' \quad (15)$$

$$x_W = \tan(\beta) h \quad (16)$$

With $DO = h$

The absolute world x -coordinate of the object is:

$$x_{obj} = x_{drone} + \cos(\gamma) \cdot x_W - \sin(\gamma) y_W \quad (17)$$

Where:

x_W, y_W : the world coordinates in x and y directions of point W;

x_{obj} : the absolute world x-coordinate of the object;

x_{drone} : the absolute world x -coordinate of the drone;

γ : the yaw angle of the drone.

In a similar approach, the y -coordinate of the object (the moving obstacle) is:

$$y_{obj} = y_{drone} + \cos(\gamma) \cdot y_W + \sin(\gamma) x_W \quad (18)$$

where:

y_{obj} : the absolute world y -coordinate of the object;

y_{drone} : the absolute world y-coordinate of the drone;

The moving obstacle's velocity is estimated as follows:

$$v_{obj}(t) = \sqrt{\left(\frac{x_{obj}(t) - x_{obj}(t-1)}{\Delta t}\right)^2 + \left(\frac{y_{obj}(t) - y_{obj}(t-1)}{\Delta t}\right)^2} \quad (19)$$

3.3 Landing Algorithm

In our approach, the landing algorithm based on the above obstacle detection method is presented in Figure 8. In which, depending on specific situations, the drone decides the best location for the landing task.

Figure 9 presents an example of a UAV landing application for victim rescue purposes [28]. Assume the map of the horizontal landing plane is presented as in Figure 9a with the purple area occupied by obstacles and the green area free for landing. Take into account the dimension of the drone, the area that is marked as "obstacle" should be extended by the drone's (rounded up) radius. Therefore, the real possible landing spot is the green locations as shown in Figure 9b. As the landing map is a binary matrix, the optimal landing spot can quickly be found by checking for every possible landing point the distance to the target location. The optimal landing spot is then the one with the smallest distance to the victim's location. Figure 9c presented an optimal landing spot in terms of the distance to the target location (victim's position).

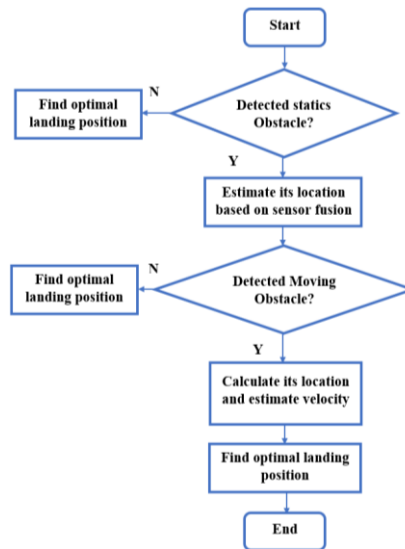


Figure 8
Landing algorithm flowchart

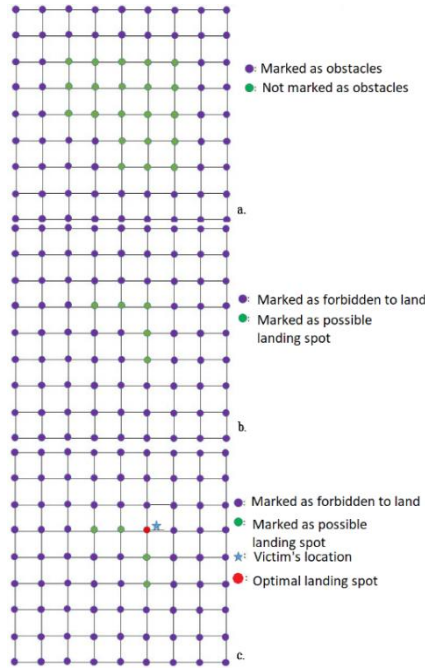


Figure 9

Landing mapped protocol

4 Control Design Based on an Accelerated Particle Swarm Optimization Algorithm

In this section, the control approach for the drone is presented. The cascade control includes an outer loop and an inner loop. For more details about the main characteristics of the Ar.Drone 2.0, please refer [29]. The outer loop transforms the landing commands into the AR.Drone 2.0. The high layer consists of a C++ application in Visual Studio, which allows accessing all drone's communication channels, therefore, enabling functions to send commands or set configurations, receive and store data from sensors and videostream. The inner loop is considered a black-box. The identification method is used to identify the transfer functions of this black-box as shown in Figure 10.

As the drone is complicated to control as it is a multiple-inputs and multiple-outputs system (MIMO). However, it can be considered as a Linear Time-Invariant (LTI) System. Therefore, it is decomposed into multiple single input and single output (SISO) systems.

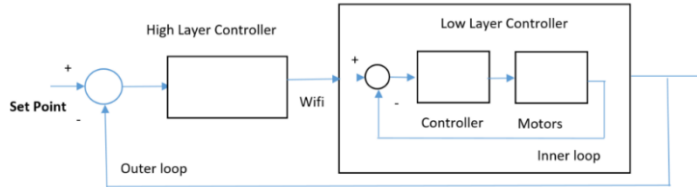


Figure 10

Quadrotor layers: the low layer represents the electronic assistance and the embedded operative system on the AR.Drone; the high layer represents the pilot

The parametric identification using the prediction error method (PEM) and Pseudo-Random Binary Signal (PRBS) input signals is used to identify the system. The results are:

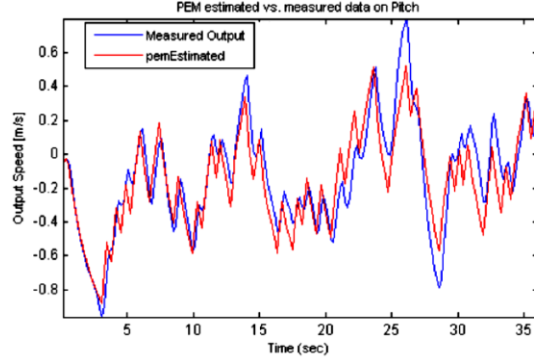
$$\begin{aligned}
 H_x(s) &= \frac{V_{out}^x(s)}{V_{in}^x(s)} = \frac{7.27}{(1.05s + 1)} \\
 H_x(s) &= \frac{V_{out}^x(s)}{V_{in}^x(s)} = \frac{7.27}{(1.05s + 1)} \\
 H_{altitude}(s) &= \frac{\dot{\zeta}_{out}(s)}{\dot{\zeta}_{in}(s)} = \frac{0.72}{(0.23s + 1)}
 \end{aligned} \tag{8}$$

For the outer loop controller, the inputs are the setpoints for speed ($V_{in}^x(s)$, $V_{in}^y(s)$, $\dot{\zeta}_{in}(s)$) and the outputs are the response of the internal control to follow those setpoints ($V_{out}^x(s)$, $V_{out}^y(s)$, $\dot{\zeta}_{out}(s)$). The drone provides an estimation by using its equipped sensors, making it also possible to obtain the position by integrating the relative speeds.

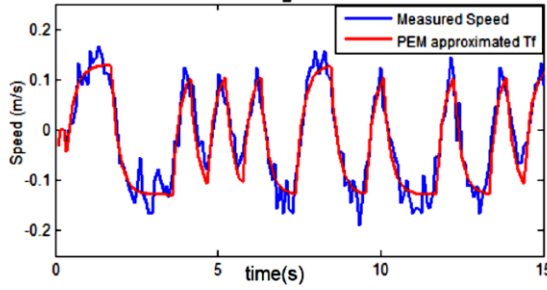
For each drone, specific transfer functions are archived. The validation of the AR.Drone 2.0's transfer functions is shown in Figure 11. The results prove that drone movements are properly approximated.

In this study, the controllers of the AR.Drone 2.0 are designed based on a multi-objective particle swarm optimization (MOPSO) [33]. This method is implemented to facilitate convergence to an optimal set of PID. In the traditional PSO, the particle's positions are found based on two values: (1) the current global best Gb and (2) the personal best Pb_i [32].

In our approach, only the global best is used to accelerate the convergence of the algorithm. The diversity of each particle is simulated by using randomness.



(a) The validation of the pitch/roll transfer function response and measurement (speed in m/s).



(b) The validation of the altitude transfer function response and the measurement(speed in m/s).

Figure 11

The comparison of pitch/roll (a), altitude (b) transfer function

The velocity and position vectors are designed as:

$$V_i(t+1) = V_i(t) + c_1 r + c_2 (Gb(t) - X_i(t)) \quad (10)$$

$$X_i(t+1) = X_i(t) + V_i(t+1)\Delta t \quad (11)$$

where:

$$c_1 \in [0.1 \ 0.5] * (UB-LB);$$

$$c_2 \in [0.1 \ 0.7];$$

r : a random number in $[0,1]$.

$Gb(t)$: the global best position of iteration t ;

$V_i(t)$; $V_i(t+1)$: the velocities of the particles i of iteration t , $t+1$;

$X_i(t)$; $X_i(t+1)$: the positions of the particles i of iteration t , $t+1$;

LB , UB : the lower and upper bounds of X .

The value of c_1 can be designed as the following formula to reduce the randomness.

$$c_1 = c_0 \xi^t (UB - LB) \quad (12)$$

Where:

$c_1 \in [0.1 \ 0.5]$ is the initial value of the randomness parameter while t is the number of the iterations and $\xi \in (0 \ 1)$ is a control parameter. For more detail, please refer [33] for more detail.

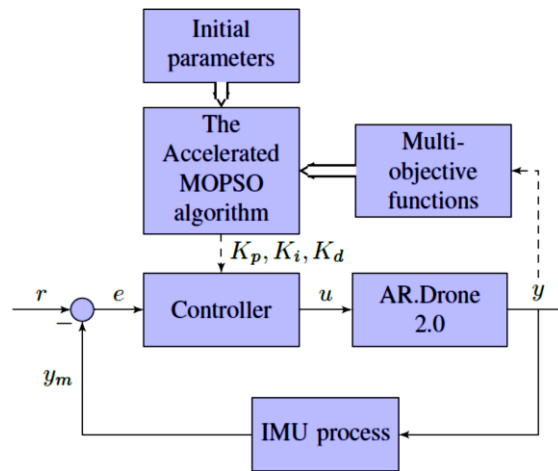


Figure 12

The block diagram of the accelerated MPSO-based PID controller

In this study, the controller parameters K_p ; K_i ; K_d are considered as an optimal set of MPSO algorithm. They are chosen to satisfy prescribed performance criteria regarding the settling times (T_s), the overshoot (OS), and the steady-state error (SSE). The objective functions are designed as:

$$\begin{aligned} J_1(X) &= |SSE| \\ J_2(X) &= OS \\ J_3(X) &= T_s \end{aligned} \quad (13)$$

where:

X is a set of parameters that is optimized, $X = (K_p; K_i; K_d)$.

Figure 12 depicts the block diagram of the controller approach based on MPSO. In this study, the dimension of the particle is 3. In the beginning, the algorithm generates arbitrary values of $K_p; K_i; K_d$ and calculates the objectives function. It continuously updates the set of parameters until it reaches the optimal set.

The composited objective optimization is designed as follows.

$$J(X) = \beta_1 J_1(X) + \beta_2 J_2(X) + \beta_3 J_3(X) \quad (14)$$

Where: $\beta_1, \beta_2, \beta_3$ are positive values, in this study, those values are set as $\beta_1 = 0.3$, $\beta_2 = 0.25$, $\beta_3 = 0.45$. The swarm size $N = 50$. The maximum number of iterations $T_{max} = 50$; $c_0 = 0.2$; $c_2 = 0.7$; $\xi = 0.97$.

The results of the x, y, z position controllers for the drone are presented in Figure 13, Figure 14, Figure 15 respectively. It can be observed that the measured position accurately follows the simulated one which correctly tracks the reference without overshoot. The control efforts are reasonable. The position errors between real and desired positions in $x; y; z$ coordinates are presented in Figure 16.

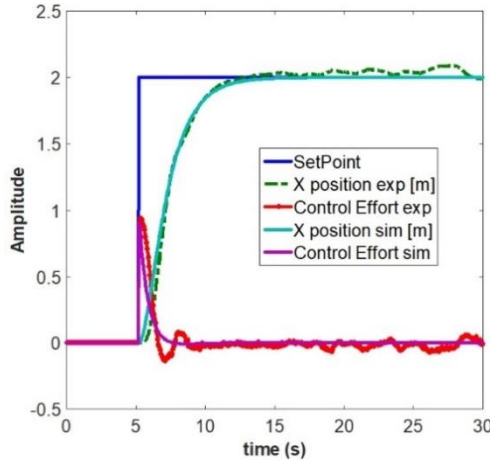


Figure 13
PID controlled X position (m)

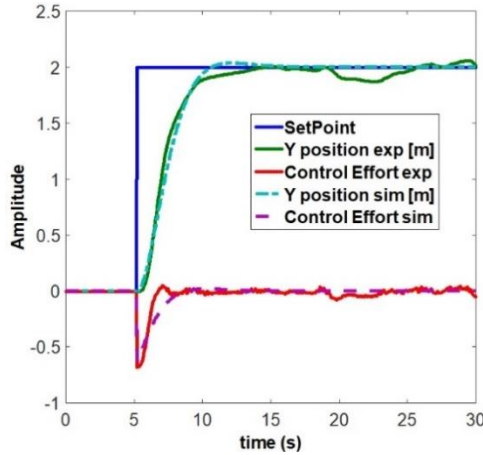


Figure 14
PID controlled Y position (m)

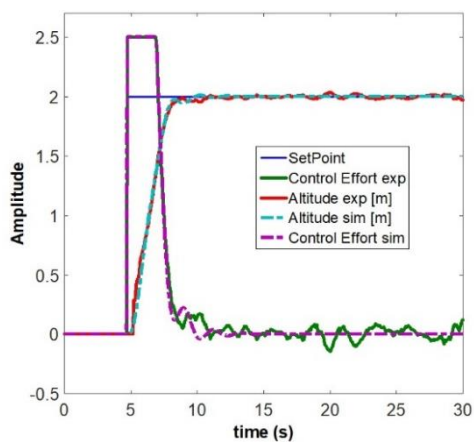


Figure 15

PID controlled Altitude (Z) position (m)

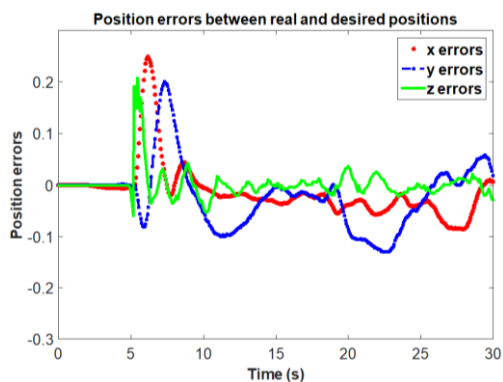


Figure 16

Position errors (m) between real and desired positions in x; y; z coordinates

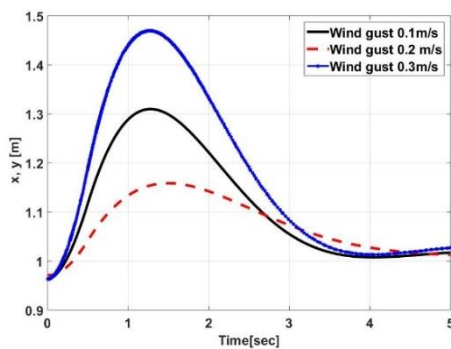


Figure 17

Wind disturbance X(Y) response comparison

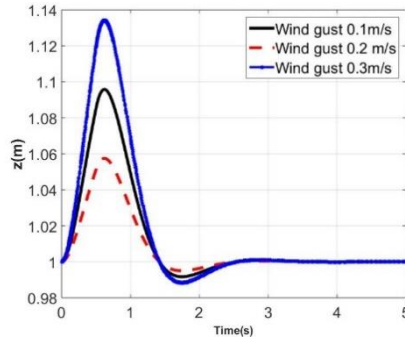


Figure 18

Wind disturbance Z response comparison

In this study, the stabilization of the drone underwind gust conditions is considered. The UAV's controller should handle the disturbance rejection problem approximately. Figure 17 and Figure 18 illustrate the wind disturbance to the step response where wind gusts are 0.1m/s; 0.2 m/s; 0.3 m/s during 5s respectively. The results show that the designed controllers are stable with wind gust disturbance.

5 Experiment Results

In this section, the results of moving obstacle velocity estimation of the AR.Drone 2.0 are illustrated. In addition, the landing processes in environments with the presence of unknown static/dynamic obstacles are validated.

5.1 Velocity Estimation of Moving Obstacles Results

For safe landing in a dynamic environment, an essential requirement is the perception of the moving object's velocity. To validate our approach, the velocity estimation system setup consists of a Lego Mindstorm that is driving with a constant speed of 0.2 m/s and an AR.Drone 2.0 that estimates the velocity of the robot using its bottom camera. To obtain a proper velocity estimation, a cumulative moving average filter is applied in this application. The result shows that the measurement is reasonable at the first stage as shown in Figure 19. After that, the velocity estimation is quite accurate as shown in Figure 20.

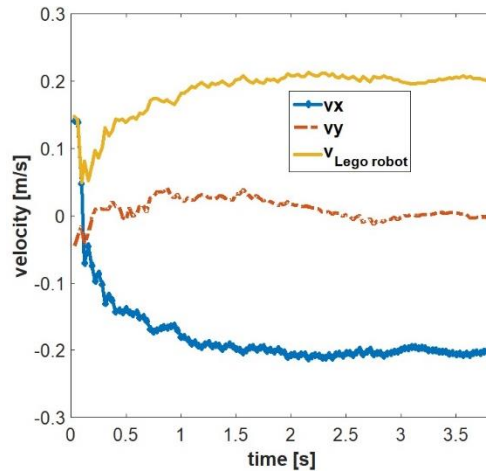


Figure 19

The cumulative moving average filter that is applied to the velocity measurements of the moving obstacle (first stage)

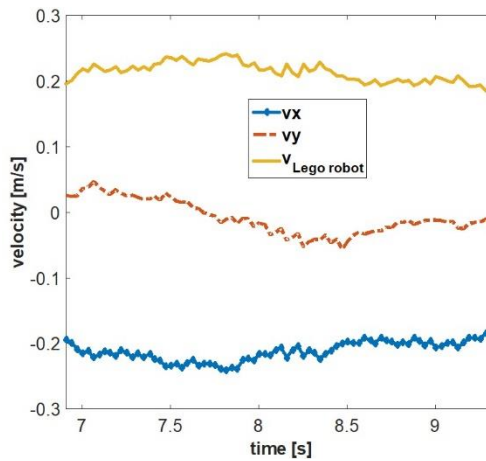


Figure 20

The cumulative moving average filter that is applied to the velocity measurements of the moving obstacle (second stage)

The experiment shows that velocity estimation based on a vision system is feasible from a drone. Even considering delays, accurate velocity estimations of moving obstacles are possible. This speed information can help autonomous UAV to interpret complex situations to find a suitable position for its landing.

5.2 Autonomous Landing Results

The idea for autonomous landing task of the drone is to find the position in the environment at which there are no static/potential moving obstacles and has the shortest distance to the target. As soon as the drone detects a static obstacle by its ultrasonic sensor, the position of that obstacle is provided by the sensor fusion method. The drone's velocity information is provided as stated in the previous subsection.

Case I: Landing in presence of static obstacles

The experiment for the landing process in presence of static obstacles is conducted in a room with full furniture such as sofa, chair, table, etc.; the target (victim) location is marked as a blue rectangle on the ground (see Figure 21). The outcomes prove that the drone is able to avoid all obstacles on its path and autonomous landing on the optimal position (victim's position).

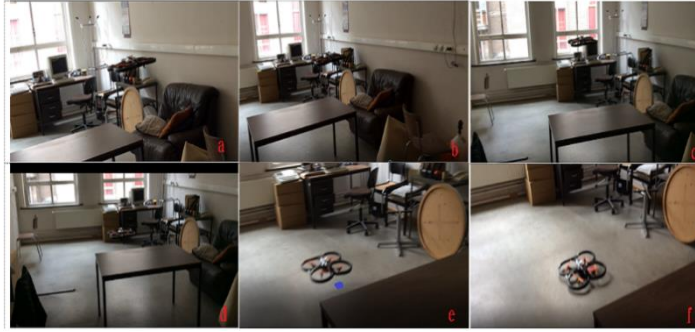


Figure 21

Landing in presence of static obstacles results in the sequence of time. a) fly over a static obstacle (sofa); b) fly over a static obstacle (table); c) fly to the target position; d) closer to the target; e) prepare to land; f) landing

Case II: Landing in presence of moving obstacles

In this experiment, a Lego Mindstorms EV3 was used to represent a moving obstacle. The moving obstacle is marked with a red marker and the target (victim's position) location is marked as the blue area on the ground as shown in Figure 22. Based on the color detection algorithm, it is possible to extract the obstacle's position in the image frame. In Figure 22a, the drone detects the Lego robot as a moving obstacle. As the target position on the robot path, the UAV stills follow the robot and flies to the position near the target. However, it did not decide to land immediately. The drone hovers and waits until the robot passes the target position then it lands safely.

The conducted experiments prove that the proposed approach allows the UAV to land on the optimal position safely in GPS denied environments with the presence of uncertain obstacles.

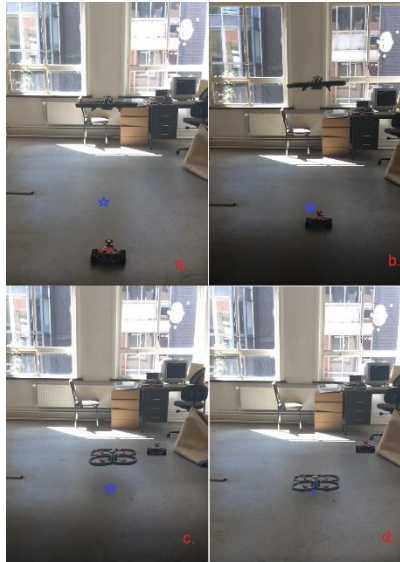


Figure 22

Landing in presence of dynamics obstacle results in the sequence of time

Conclusions

This paper presents a novel autonomous landing approach of a commercial AR.Drone 2.0 in presence of uncertain obstacles with GPS denied environment. The main achievements are: (i) the development of an autonomous navigation system to assistant landing for UAV in term of avoiding unknown and dynamic obstacles; (ii) the proposed unknown static obstacle and dynamic obstacle detection approach; (iii) the effective estimation of moving obstacle's velocity based on the drone's vision system; (iv) the development of cascade control based on an accelerated particle swarm optimization algorithm which allows the drone to land safely. The results show that the obtained model fits well with the measured data. Furthermore, the designed control strategy is capable of controlling the drone properly. In addition, the proposed autonomous landing strategy can guide the UAV to avoid static/dynamic obstacles and land on the optimal position. However, only considering moving obstacles with constant speed is the limitation of this work. Future work includes an extension of the proposed approach to multiple UAVs and a combination of multiple ground vehicles moving with variable speeds.

Acknowledgment

This work is granted by the National Foundation for Science and Technology Development of Vietnam NAFOSTED, Vietnam under project number 107.01-2018.331.

The authors would like to gratefully thank Arne Maenhaut for his assistance in conducting the experiments in which some results were part of his master thesis guided at our lab, Ghent University, Belgium.

References

- [1] S. Minaeian, J. Liu, and Y.-J. Son, Vision-based target detection and localization via a team of cooperative UAV and UGVs, *IEEE Transactions on System, Man, and Cybernetics: Systems*, Vol. 46 (7), 2016, pp. 1005-1016
- [2] T. T. Mac, C. Copot, D. T. Tran, R. De Keyser, Heuristic approaches in robot path planning: A survey, *Robotics and Autonomous Systems*, Vol. 86, 2016, pp. 13-28
- [3] T. T. Mac, C. Copot, D. T. Tran, R. De Keyser, A Hierarchical Global Path Planning for Mobile Robot based on Multi-Objective Particle Swarm Optimization, *Applied Soft Computing*, Vol. 59, 2017, pp. 68-76
- [4] D. Zheng, H. Wang, J. Wang, S. Chen, W. Chen, X. Liang, Image-Based Visual Servoing of a Quadrotor using Virtual Camera Approach, *IEEE/ASME Transactions on Mechatronics*, Vol. 22 (2), 2017, pp. 972-982
- [5] F. Kendoul, Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems, *Journal of Field Robotics*, Vol. 29 (2), 2012, pp. 315-378
- [6] J. Sprinkle, J. Eklund, S. Sastry, Deciding to land a UAV safely in real time, *Proceedings of the American Control Conference*, 2005, pp. 3506-3511
- [7] S. Saripalli, G. Sukhatme, Landing a helicopter on a moving target, in: *Proceedings IEEE International Conference on Robotics and Automation*, 2007, pp. 2030-2035
- [8] F. Alarcon, D. Santamaria, A. Daniel, UAV helicopter relative state estimation for autonomous landing on moving platforms in a GPS-denied scenario, *IFAC-PapersOnLine*, Vol. 48 (9), 2015, pp. 37-42
- [9] W. Bagen, J. Hu, Y. Xu, A vision-based unmanned helicopter ship board landing system, *2nd International Congress on Image and Signal Processing*, 2009, pp. 1-5
- [10] C. De Wagter, J. Mulder, Towards vision-based UAV situation awareness, in: *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, 2005, pp. 1-16
- [11] K. H. Hsia, S. F. Lien, J. P. Su, Height estimation via stereo vision system for unmanned helicopter autonomous landing, *2010 IEEE International Computer Communication Control and Automation Symposium*, 2010, pp. 257-260

- [12] Y. H. Shin, S. Lee, J. Seo, Autonomous safe landing-area determination for rotorcraft UAVs using multiple IR-UWB radars, *Aerospace Science and Technology*, Vol. 69, 2017, pp. 617-624
- [13] S. Scherer, L. Chamberlain, S. Singh, Autonomous landing at unprepared sites by a full-scale helicopter, *Robotics and Autonomous Systems*, Vol. 60 (12), 2012, pp. 1545-1562
- [14] Y. Chong, Y. Cai, Q. Chen, Multi-resolution visual fiducial and assistant navigation system for unmanned aerial vehicle landing, *Aerospace Science and Technology*, Vol. 67, 2017, pp. 249-256
- [15] Y. F. Shen, Z. U. Rahman, D. Krusienski, J. Li, A vision-based automatic safe landing-site detection system, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49 (1), 2013, pp. 294-311
- [16] J. Seo, T. Walter, Future dual-frequency GPS navigation system for intelligent air transportation under strong ionospheric scintillation, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15 (5), 2014, pp. 2224-2236
- [17] S. Gao, L. Xue, Y. Zhong, C. Gu, Random weighting method for estimation of error characteristics in SINS/GPS/SAR integrated navigation system, *Aerospace Science and Technology*, Vol. 46, 2015, pp. 22-29
- [18] J. Seo, T. Walter, P. Enge, Availability impact on GPS aviation due to strong ionospheric scintillation, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 47 (3), 2011, pp. 1963-1973
- [19] J. A. G. Pulido, G. Pajares, S. Dormido, J. M. De la Cruz, Recognition of a landing platform for unmanned aerial vehicles by using computer vision-based techniques, *Expert Systems With Applications*, Vol. 76, 2017, pp. 152-165
- [20] R. A. Pradeep, P. Radhakant, Robust auto-landing of fixed-wing UAVs using neuro-adaptive design, *Control Engineering Practice*, Vol. 60, 2017 pp. 218-232
- [21] C. Luo, X. Li, Y. Li, Q. Dai, Biomimetic Design for Unmanned Aerial Vehicle Safe Landing in Hazardous Terrain, *IEEE/ASME Transactions on Mechatronics*, Vol. 21 (1), 2016, pp. 531-541
- [22] X. Yang, M. Garratt, H. Pota, Nonlinear position control for hover and automatic landing of uavs, *IET Control Theory and Application*, Vol. 9, 2012, pp. 911-920
- [23] B. Herisse, T. Hamel, R. Mahony, F. X. Russotto, Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow, *IEEE Transactions on Robotics*, Vol. 28 (1), 2012, pp. 77-89

- [24] Alexander C. Woods and Hung M. La, A Novel Potential Field Controller for Use on Aerial Robots, *IEEE Transactions on System, Man, and Cybernetics: Systems*, Vol. 49 (4), 2019, pp. 665-676
- [25] M. F. Lee, S. F. Su, J. W. Yeah, H. M. Huang, J. Chen, Autonomous landing system for aerial mobile robot cooperation, *IEEE International Symposium on Soft Computing and Intelligent Systems*, 2014, pp. 1306-1311
- [26] A. Benini, M. J. Rutherford, K. P. Valavanis, Real-time, GPU-based pose estimation of a UAV for autonomous takeoff and landing, *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3463-3470
- [27] <https://www.pozyx.io/>
- [28] P. Van de Voorde, S. Gautama, A. Momont, C. M. Ionescu, P. De Paepe, N. Fraeyman, The drone ambulance [A-UAS]: golden bullet or just a blank?, *Resuscitation*, Vol. 116, 2017, pp. 46-48
- [29] T. T. Mac, C. Copot, A. Hernandez, R. De Keyser, Improved Potential Field Method for Unknown Obstacle Avoidance Using UAV in Indoor Environment, *IEEE 14th International Symposium on Applied Machine Intelligence and Informatics*, 2016, pp. 345-350
- [30] R. De Keyser, C. M. Ionescu, FRTool: a frequency response tool for CACSD in Matlab, in *Proc. of the IEEE Conf. on Computer Aided Control Systems Design*, Munich, Germany, 2006, pp. 2275-2280
- [31] A. Hernandez, C. Copot, J. Cerquera, H. Murcia, R. De Keyser, Model Predictive Path-Following Control of an AR.Drone Quadrotor, *XVI Latin American Control Conference The International Federation of Automatic Control*, Cancun, Mexico, 2014, pp. 618-623
- [32] X. S. Yang, S. Deb, S. Fong, Accelerated Particle Swarm Optimization and Support Vector Machine for Business Optimization and Applications, *Networked Digital Technologies, Communications in Computer and Information Science*, Vol. 136, Springer, 201, pp. 53-66
- [33] T. T. Mac, C. Copot, R. De Keyser, M. Ionescu, The Development of an Autonomous Navigation System with Optimal Control of an UAV in Partly Unknown Indoor Environment, *Mechatronics*, Vol. 49, 2018, pp. 187-196