A Comparative Study of Collision Avoidance Techniques for Unmanned Aerial Vehicles

Alexander Alexopoulos, Amr Kandil, Piotr Orzechowski and Essameddin Badreddin
Automation Laboratory, Institute for Computer Engineering, University of Heidelberg
Mannheim Germany
alexander.alexopoulos@ziti.uni-heidelberg.de

Abstract— **In this paper three collision avoidance methods for an unmanned aerial vehicle (UAV) are tested and compared to one another. The quadrocopter dynamic model with attitude and velocity controller, a trajectory generator and a selection of collision avoidance approaches were implemented. The first collision avoidance method is based on a geometric approach which computes a direction of avoidance from the flight direction and simple geometric equations. The second technique uses virtual repulsive force fields causing the UAV to be repelled by obstacles. The last method is a grid-based online path re-planning algorithm with A* search that finds a collision free path during flight. Various flight scenarios were defined including static and dynamic obstacles.**

Keywords- ***unmanned aerial vehicles; geometric collision avoidance; repulsion force; collision avoidance with A* search***

I. Introduction

In the last two decades, autonomous unmanned aerial vehicles (UAVs), and especially unmanned helicopters or quadrocopters have won a great importance not only because of the variety of their application fields, but also because they serve as very interesting research platforms. Their ability to take-off and land vertically as well as the hovering flight qualifies them for a wide range of both military and civil domains under changing flight and environment conditions. To name just a few applications, UAVs are being increasingly used for the surveillance of power plants, inspection of power lines, rescue missions and mapping of cultural heritage sites [1]. The use of safety critical systems such as UAVs requires the consideration and implementation of some safety measures to avoid accidents during flight, which could have catastrophic results. One of the main flight risks is the collision with moving or stationary objects. Therefore, a collision avoidance mechanism for UAVs is considered to be indispensable. The UAV must be capable of detecting potential collisions with previously unknown stationary or moving objects and to perform suitable evasive manoeuvres to avoid collision with the detected objects. In the following some fundamental methods tackling this problem are briefly described. One of the simplest methods for collision avoidance is that of the geometric approach. In [3] a geometric approach for conflict detection and resolution is described. By calculating the "point of closest approach" (PCA), the worst case condition between two UAVs is evaluated. If a collision danger is detected, the PCA is used to adapt the reference velocity vectors of both aircrafts, such that the UAVs assure a safe distance between each other. Therefore cooperation between the aircrafts is assumed here, which is in most cases not given. In [4] a collision avoidance method for an unmanned helicopter based on potential fields is proposed, which is a modification of the method developed in [5]. In [5] each obstacle is assigned with a repulsive force field and the goal is assigned with an attracting force field, hence a robot can follow a collision-free path by computing a motion vector from the superposed force fields. In [4] the assignment of repulsive forces is performed to obstacles only. Thence this potential field approach is limited to collision avoidance. Some other collision avoidance approaches require a grid model of the environment. As described in [6] a graph search algorithm like A*([7]) is used to find the shortest way around obstacles through this grid representation of the environment. Because of the possibility of facing unknown or/and moving obstacles, an on-line re-planning is to be applied. In [8] an evolutionary approach for collision avoidance is has been utilized. If an obstacle is detected, the reference trajectory is distributed in segments. After a multiple random mutation of those segments, the best mutations are chosen for the new collision-free reference trajectory.

In this work three different collision avoidance methods were chosen to be tested and compared to one another for the quadrocopter platform. The chosen methods are the geometric approach, the repulsion force approach and finally the local path planning with A* search approach. The first two approaches were chosen because they provide a reflexive behavior of the system and enable real-time collision avoidance implementation. The last one was chosen to have a counterpart to the reflexive behavior-based approaches. To simulate and to compare these approaches a dynamic model of a quadrocopter with attitude and velocity controller was selected.

In the next section the problem formulation is stated and the corresponding solution approach is presented. In section 3 a brief system description of the controlled quadrocopter model [9] is given. The chosen collision avoidance techniques are introduced in section 4. In Section 5 the simulation results are presented and discussed. Finally, some interesting aspects and remarks are mentioned in the conclusions.

978-1-4799-0652-9/13 $31.00 © 2013 IEEE
DOI 10.1109/SMC.2013.338

II. Problem Statement and Solution Approach

The tasks were to choose a suitable UAV model and controller enabling cruise flight simulation, the implementation of a trajectory generator, selection, adaption and implementation of several collision avoidance methods for the given UAV Model and evaluation and comparison of the collision avoidance methods.

Therefore, a non-linear state space model of a quadrocopter based on the work in [9] was used and a non-linear attitude and velocity controller based on the back-stepping approach were implemented within the simulation environment. Also a spline-based trajectory generator was programmed. For Navigation a P-Controller was used for Position Control. Integration of the geometric, the repulsion force and the A* search collision avoidance approach into the system structure was then performed. Finally, several scenarios for mission flight with unknown obstacles (static and dynamic) were defined. The different collision avoidance approaches were evaluated with respect to safety distance violations and collision prevention against static and dynamic obstacles.

III. System Description

A. Dynamic Model

For modeling the quadrocopter dynamics the mechanical configuration depicted in Fig. 1 was assumed. The body fixed frame and the inertial frame are denoted by $\mathbf{e^B}$ and $\mathbf{e^I}$, respectively. The UAV is defined as a point mass. To derive the equations of motions, the following notations are necessary. $\mathbf{P^I} = (x, y, z)^T$ is the position vector of the quadrocopters' centre in the inertial frame, $\mathbf{P^B} = (x_B, y_B, z_B)^T$ is the position vector of the helicopters' centre in the body fixed frame, $\mathbf{v} = (u, v, w)^T$ are the linear velocities in body fixed frame, $\boldsymbol{\omega} = (p, q, r)^T$ are the angular rates for roll, pitch and yaw in body fixed frame and $\boldsymbol{\Theta} = (\phi, \theta, \psi)^T$ is the vector of Euler angles. A key component of the quadrocopter model is the transformation between inertial and body frames. Rigid body dynamics are derived with respect to the body frame that is fixed in the centre of gravity of the quadrocopter. However, to simulate the motion of the quadrocopter in the inertial frame a transformation of the coordinates is needed.

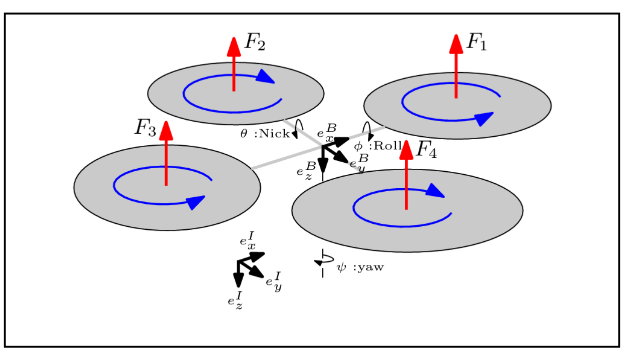

Figure 1. Mechanical Configuration of a quadrocopter with body fixed and inertial frame

If the quadrocopter attitude is parameterized in terms of Euler angles, the transformation can be performed using the rotation matrix $\mathbf{R}(\boldsymbol{\Theta})$, which is a function of roll, pitch and yaw angles. Using s and c as abbreviations for sin(.) and cos(.) respectively, the linear velocities defined in the inertial frame $(v_x, v_y, v_z)^T$ can be obtained as follows:

$$\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi - c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{1}$$

The transformation of positions defined in the body frame into the corresponding positions in the inertial frame can be obtained by:

$$\begin{bmatrix} \mathbf{P^I} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}(\boldsymbol{\Theta}) & \mathbf{P^I_{B,org}} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P^B} \\ 1 \end{bmatrix} \tag{2}$$

The equations of motion are derived from the first principles (Newton–Euler laws [10]) to describe both the translational and rotational motion of the quadrocopter, leading to the following non-linear state space model, with the state vector $\dot{\mathbf{x}}$:

$$\dot{\mathbf{x}} = (u, v, w, \phi, \theta, \psi, p, q, r)^T = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)^T,$$

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ p \\ q \\ r \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} -(\cos x_4 \sin x_5 \cos x_6 + \sin x_4 \sin x_6)\dfrac{u_1}{m} \\ -(\cos x_4 \sin x_5 \sin x_6 - \sin x_4 \cos x_6)\dfrac{u_1}{m} \\ g - (\cos x_4 \cos x_5)\dfrac{u_1}{m} \\ x_7 \\ x_8 \\ x_9 \\ \dfrac{I_y - I_z}{I_x} x_8 x_9 + \dfrac{L}{I_x} u_2 - \dfrac{I_R}{I_x} x_8 g(u) \\ \dfrac{I_z - I_x}{I_y} x_7 x_9 + \dfrac{L}{I_y} u_3 + \dfrac{I_R}{I_y} x_7 g(u) \\ \dfrac{I_x - I_y}{I_z} x_7 x_8 + \dfrac{1}{I_z} u_4 \end{pmatrix}, \tag{3}$$

$\mathbf{u}^T = (u_1, u_2, u_3, u_4)$ are the inputs for altitude, roll, nick and yaw. I_x, I_y, I_z are the inertia about the x, y, z-axes of the body fixed frame, I_R is the rotor moment of inertia, L is the length between the centre of gravity of the quadrocopter and the centre of one rotor, g is the gravitation constant and $g(u)$ is a function of u depending on the rotors' angular velocities. The derivation of the model is rather sophisticated and it cannot be handled here in details. For more details on quadrocopter modelling [11] can be consulted. A closer look at the state space model reveals that on the one hand the angular accelerations depend only on the angular rates and the input vector $\mathbf{u}$. On the other hand the linear accelerations depend on the Euler angles and $\mathbf{u}$. Hence the state space model can be divided into two interlinked sub-models M_1 and M_2 . Table 1 lists the chosen parameters based on [9].

B. Attitude and Velocity Control

The model structure is suitable for a cascaded attitude and velocity controller. The attitude controller, controlling subsystem M_1 is ordered in the (faster) inner loop and the velocity controller, controlling M_2 in the (slower) outer loop. The control of attitude and velocity of quadrocopters are not part of this work. Therefore consult [9] and [12] for more details about the present controller. With the given control structure very sufficient reference reactions were derived in simulations.

C. Trajectory generation and Navigation

A P-Controller was used for the position control. Smooth trajectories from the given waypoints can be constructed by using splines to obtain appropriate reference trajectories, which the quadrocopter shall track while facing obstacles. Fig. 2 shows satisfactory navigation behaviour with a desired average speed of 7.5m/s (simulation result: 5.496m/s), having an average position error of 7.61cm and a maximum position error of 38.24cm. It is now possible to integrate several collision avoidance methods within this setup.

TABLE I. MODEL PARAMETERS

Parameter	Value
m	$0.5[kg]$
L	$0.2[m]$
$I_x = I_y$	$4.85 \cdot 10^{-3}[kg \cdot m^2]$
I_z	$8.81 \cdot 10^{-3}[kg \cdot m^2]$
I_R	$3.36 \cdot 10^{-5}[kg \cdot m^2]$
thrust factor	$2.92 \cdot 10^{-6}[kg \cdot m]$
air drag factor	$1.12 \cdot 10^{-7}[kg \cdot m^2]$

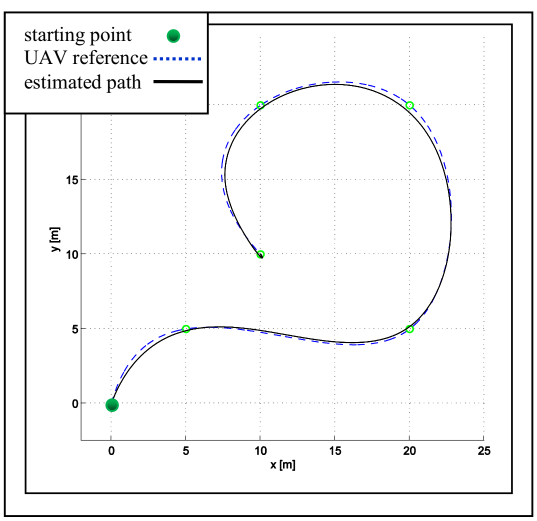

Figure 2. Navigation with a desired average speed of 7.5m/s

IV. COLLISION AVOIDANCE METHODS

A. Geometric Approach

In this algorithm, the agents involved are typically considered as a single point mass with a constant velocity vector denoting speed and direction. This approach originally determines a cooperative collision avoidance solution between two agents, which are in conflict and are able to communicate with each other. If another agent is detected, it is checked if a collision risk exists. Therefore the PCA and the resultant miss distance $\mathbf{r_m}$ are calculated. Thereby, a linear extrapolation of the agents' trajectories is performed [3]. The miss distance is defined as:

$$\mathbf{r_m} = \hat{\mathbf{c}} \times (\mathbf{r}_i \times \hat{\mathbf{c}}) \, , \tag{5}$$

where $\mathbf{r_i}$ is the relative distance vector and $\hat{\mathbf{c}}$ denotes the unit vector of the relative velocity $\mathbf{c}$ (Fig. 3). The miss distance vector $\mathbf{r_m}$ and the relative velocity $\mathbf{c}$ are obviously orthogonal:

$$\mathbf{r_m} \cdot \mathbf{c} = 0 \, . \tag{6}$$

The miss distance vector can also be expressed, as the relation between $\mathbf{r_i}$, $\mathbf{c}$ and the time τ needed to reach the PCA:

$$\mathbf{r_m} = \mathbf{r_i} + \mathbf{c} \cdot \tau \, , \tag{7}$$

(6) and (7) yield:

$$\tau = -\frac{\mathbf{r_i} \cdot \mathbf{c}}{\mathbf{c} \cdot \mathbf{c}} \, . \tag{8}$$

If the agents approach towards each other, then $\tau > 0$ will hold and respectively $\tau < 0$, if they veer away from each other. Hence for $\tau > 0$ it must be checked, if a pre-defined safety distance $\mathbf{r_s}$ is maintained towards the PCA. Therefore the rest distance $\mathbf{r_f}$ is obtained:

$$\mathbf{r_f} = \mathbf{r_s} - \|\mathbf{r_m}\| \tag{9}$$

Thus the conflict condition is $\mathbf{r_f} > 0$ and $\tau > 0$. So if there is a collision risk, the algorithm tries to maximize the miss distance, by assigning every involved agent an evasive maneuver, thus every agent has a hand in maximizing $\mathbf{r_m}$, called "vector sharing". This is depicted in Fig. 4, where Agent A has to compensate following part of the miss distance:

$$\mathbf{r}_A^{vs} = \frac{|\mathbf{v_B}|}{|\mathbf{v_A}| + |\mathbf{v_B}|} \frac{r_f}{|r_m|} (-r_m) \, . \tag{10}$$

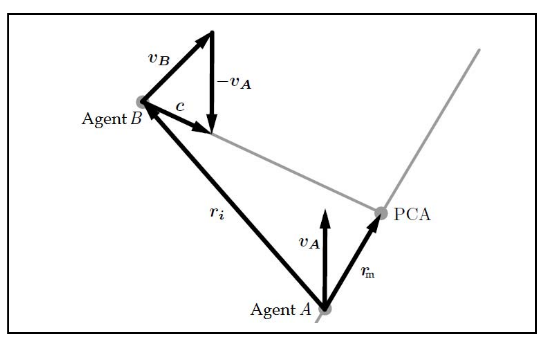

Figure 3. Two agents in conflict

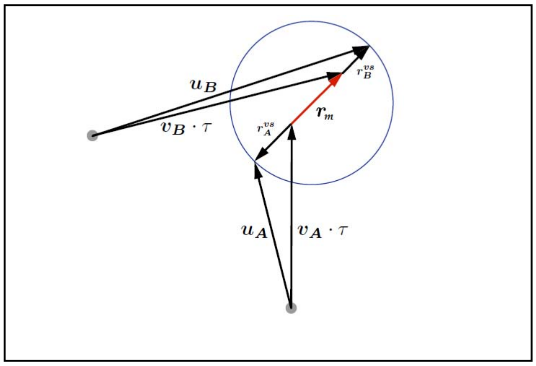

Figure 4. Evasive manoeuvre based on the miss distance

To do that, the direction of its velocities must be adapted according to:

$$\hat{u}_A = \frac{\mathbf{v_A} \cdot \tau + r_A^{vs}}{\left|\mathbf{v_A} \cdot \tau + r_A^{vs}\right|} . \tag{11}$$

Analogous equations hold for agent B so that the safety distance $\mathbf{r_s}$ can be maintained:

$$r_s = \left|r_A^{vs}\right| + \left|r_B^{vs}\right| + \left|r_m\right| . \tag{12}$$

If two objects are on frontal collision course, r_m would be zero according to equation (5). Therefore [3] suggests countering that with variation of r_m for one agent.

In this work no communication and cooperation between the agents is assumed. In other words, no "vector sharing" is performed and the UAV has to compensate $\mathbf{r_m}$ all by itself. Thus some simple modifications to the algorithm were done.

B. Repulsion Force Approach

In [5] a method for preventing a mobile robot from collisions based on artificial potential field concept was presented. The geometry and motion of robot and obstacles are considered, thus the shape and position of obstacles in the environment must be known. Since the information about obstacles is usually not known in advance, it would be favorably to place the potential field forces around the mobile robot itself than on the obstacles. This idea was presented in [13], while applying the potential field forces on a mobile robot. In addition to the previously mentioned advantage, this approach takes also the velocity of the robot into consideration. The repulsion force is adapted according to the robot velocity. When the velocity of the robot moving towards an obstacle increases, the repulsive force on the robot will be increased. The approach for collision avoidance introduced in [13] was modified in [4] so that it can be utilized for a helicopter application.

Based on the work of [4], a formula for the calculation of the repulsive forces in taking into account the relative velocity to the obstacles, and not only the UAVs' velocity, was developed. The algorithm computes the repulsion force to each obstacle i according to the formula:

$$F_i = sat\left(\left(\cos\alpha_i \cdot \frac{c_1 + c_2 \dfrac{\mathbf{v^r}}{\mathbf{v^q_{max}}}}{r_i - r_s}\right)^2\right), \tag{13}$$

where sat(.) constraints to the maximal possible repulsion force. The angle to an obstacle i is denoted by α_i, $\mathbf{v^r}$ is the relative velocity vector to the obstacle, $\mathbf{v^q_{max}}$ is the maximum velocity vector of the quadrocopter, r_i is the distance to the obstacle and r_s is the safety distance in which the repulsion force is at its maximum. The two parameters c_1 and c_2 are used for repulsion force calibration at zero and maximum velocity, respectively. Very satisfying results could be achieved taking $c_1 = c_2 = 30$. The sum of all repulsion forces for n obstacle is given by:

$$\mathbf{F_{sum}} = \sum_{i=1}^{n} F_i(r_i, \alpha_i, \mathbf{v^r}) . \tag{14}$$

C. Path Re-Planning with A* search

In this approach, the environment is discretized by dividing the map in a grid. The grid is represented by a weighted graph, whereby the cells of the grid correspond to the vertices of the graph. The edges connect adjacent vertices orthogonally and diagonally. The weights of the edges correspond to the distance between the centre points of each vertex. The grid enables finding a collision free paths, with help of a graph search algorithm like A* [7]. Therefore, if an obstacle is detected, the edges that connect cells on the obstacle and the cells within the safety distance have to be removed from the graph. Then a path re-planning can be performed, by calculating the shortest path from the actual position to the goal in the modified graph. This path re-planning must be performed in every time step, for a moving obstacle. Fig. 5 depicts an example for the latter mentioned problem. There are many variations of the A* search algorithm with different heuristics for optimizing the time and space complexity. An overwiew can be found in [14]. In our case the classical A* with the Euclidean distance as a heuristic estimate and a grid resolution of one meter for acceptable execution times were used.

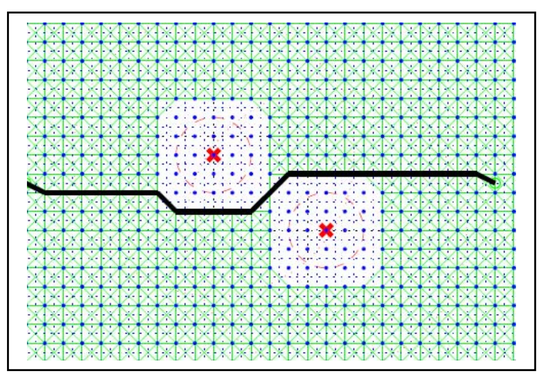

Figure 5. Obstacle avoidance by path re-planning with A* search

V. TESTS AND RESULTS

For testing and evaluating the collision avoidance methods, 13 missions, which are meant to cover most of possible scenarios regarding obstacle appearance during navigation, were defined. Three crucial scenarios, depicted in Fig. 6, are presented for a detailed evaluation in the following subscetions. For the geometric approach the collision avoidance space had to be limited to the plane. Indeed, satisfying results were achieved with the geometric approach in 3D, but the altitude controller of the quadrocopter caused problems because of the severe oscilations of the reference velocities at this approach.

A. Missions

Three out of the 13 missions were selected for the presentation here. In mission (a), the quadrocopters' goal position coincides with a static obstacles' centre position. In mission (b), the quadrocopter and a moving obstacle (5m/s) are on frontal collision course and in mission (c) the quadrocopter faces three moving obstacles (5m/s) on collision course during navigation. The quadrocopters' desired average velocity, while tracking the reference trajectory was set to 7.5m/s, while the maximum velocity was limited to 10m/s. The safety distance r_s to the obstacles was set to 3m, in order not to enter this spherical safety zone around the obstacles that were also assumed to have a spherical shape. A collision is considered to be happened If a distance to the obstacles' centre of 1m or less was reached. Finally it is assumed that the quadrocopter has sensors to detect obstacles within a radius of 10m with a focus of 180° in front direction and is able to estimate the obstacles' position and velocity.

B. Simulation Results

Regarding those three missions, all three approaches managed to prevent a collision. But the geometric approach led to several safety distance violations (Fig. 6) just as the path re-planning approach did once (Fig. 8 (b)). In the particular case of mission (a), where the goal coincides with the obstacles' centre position the geometric approach leads to a marginal stable system state, in which the quadrocopter circles around the obstacle. The repulsion force approach, delivers a similar behavior, while librating in a constant distance to the obstacle (Fig. 7 (a)). Only the path re-planning approach, provided a stable state, because no shortest path to this goal could be found, a cancelation of the navigation followed (Fig. 8 (a)). In the second mission, only the geometric approach caused a violation of the safety distance (Fig. 8 (b)). However it has to be mentioned that if the obstacles' speed would have been adequately larger then 5m/s, all three methods could not prevent a safety distance violation or even a collision with the obstacle due to the specified constraints, most notably the detection radius. A sensing system, enabling an improved farsightedness (approx. 20-30m), would be needed to get better results with faster obstacles, while the constraints concerning the motion of the quadrocopter remain. This could be attractive for outdoor applications in relation to indoor applications, where a much lower motion speed of objects can be assumed. Finally, in mission (c) all three techniques delivered satisfying results concerning the collision avoidance, while the path re-planning with A* generated a collision free path, which was closest to the reference trajectory (Fig. 8 (c)). On the one hand, this may be caused by the reflexive nature of the two other approaches. They provide an immediate reaction on the attraction yielded by the obstacles. On the other hand the path re-planning with A*, searches for an optimal solution and may cause the quadrocopter to stand still until a free path becomes available. If no optimal solution can be found, navigation will be stopped (see mission (a)). Indeed this is not the best solution for time critical applications, while for applications requiring an as effective as possible tracking of the reference trajectory the path re-planning with A* would be suitable.

Tab. 2 provides the results of all 13 missions. It can be observed, that only the geometric approach could not prevent all collisions. This approach had serious problems mainly with moving obstacles and groups of adjacent static obstacles.

TABLE II. EVALUATION OF THE 13 MISSIONS

criterion	Geometric A.	Repulsion F.	A* search
$\mathbf{r_s}$ violations (static obstacles)*	2	0	0
Collisions (static obstacles)*	1	0	0
$\mathbf{r_s}$ violations (moving obstacles)*	4	2	1
Collisions (moving obstacles)*	3	0	0

*with a total of 18 static and 10 dynamic obstacles

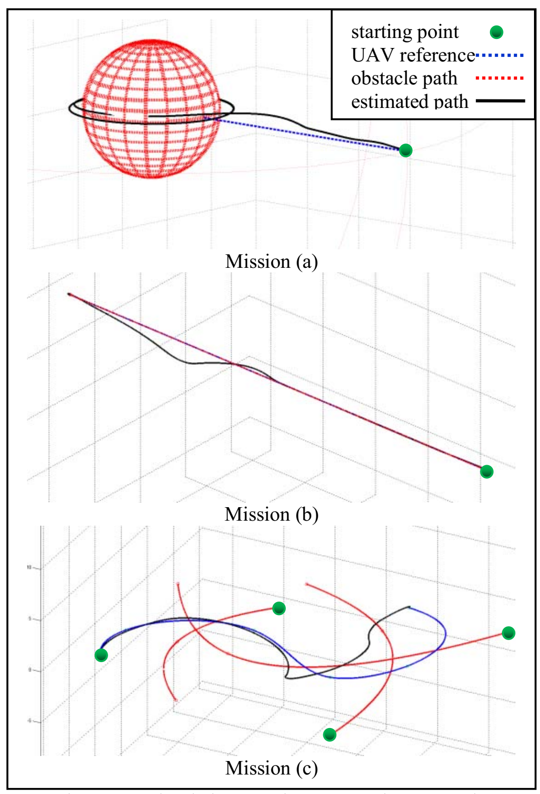

Figure 6. Simulation Results: Geometric Approach

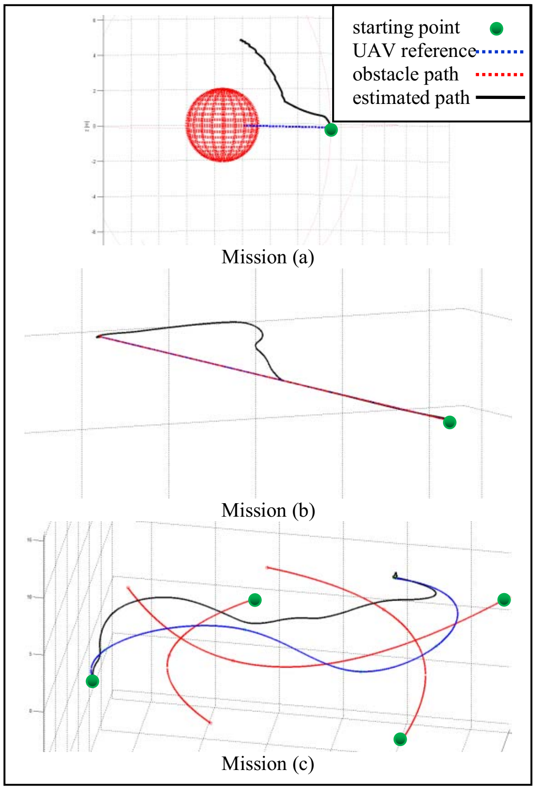

Figure 7. Simulation Results: Repulsion Force Approach

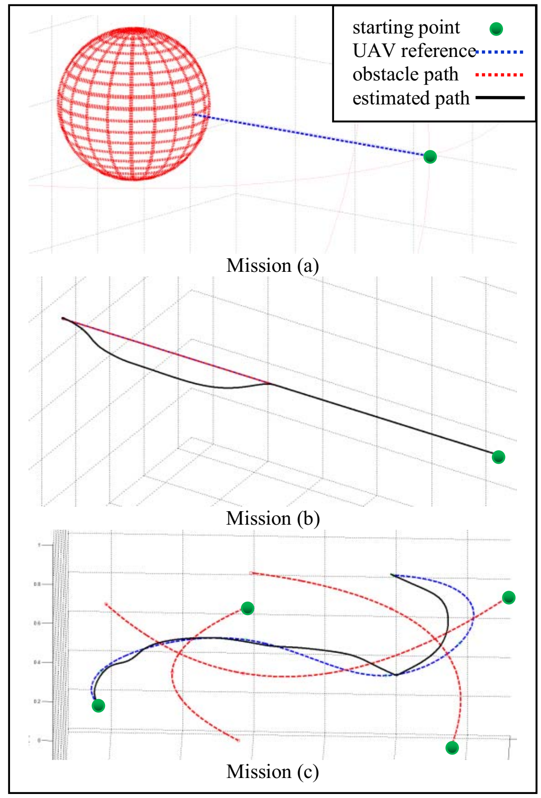

Figure 8. Simulation Results: Path re-planning with A* search

VI. CONCLUSION

For the defined missions, the repulsion force and the path re-planning with A* search proved to be suitable collision avoidance methods handling static and especially dynamic obstacles. The geometric approach may suffer because of the lacking cooperativeness (like in the original approach) with dynamical obstacles. The greatest advantage of the repulsive force approach against the path re-planning approach is the smaller time complexity of the algorithm. While the former has a time complexity of $O(n_o)$, with n_o being the number of obstacles, the latter has $O(n_o + n_x n_y \log(n_x n_y))$ in 2D, and $O(n_o + n_x n_y n_z \log(n_x n_y n_z))$ in 3D, while n_x are the number of grid cells in x-direction, n_y and n_z respectively. Hence real-time applicability has to be investigated for 2D, particularly with other heuristic estimates for the A* search [14], while for 3D it seems to be inapplicable. In conclusion the advantages for the repulsion force method prevail, thus in a future work we intend to implement this approach on the quadrocopter "*Mikrocopter L4-ME"*. More information concerning this type of quadrocopter can be found in [15].

REFERENCES

[1] P. Marker, The 2010-2015 World Outlook for Unmanned Aerial Vehicles (UAV) and Systems, ICON Group International Inc., 2009.

[2] Lozano, R., Unmanned Aerial Vehicles Embedded Control, Wiley-ISTE, 2010.

[3] J. Park, H. Oh, and M. Tahk,"UAV Collision Avoidance based on Geometric Approach," in *Int. Conf. of Instrumentation , Control and formation Technology*, Tokyo, 2008, pp. 2122-2126.

[4] A. A. Kandil, A. Wagner, A. Gotta, and E. Badreddin, "Collision Avoidance in a Recursive Nested Behaviour Control Structure for Unmanned Aerial Vehicles," in *Int. Conf. on Systems, Man, and Cybernatics*, Instanbul, 2010

[5] O. Khatib, "Real-time obstacle avoidance manipulators and mobile robots," *IEEE Conf. on Robotics and Automation, 1985, pp. 500-505*.

[6] B. Gardiner, W. Ahmad, T. Cooper, M. Haveard, J. Holt, and S. Biaz, "Collision Avoidance Techniques for Unmanned Aerial Vehicles," Auburn University, 2011.

[7] P. Hart, N. Nilsson, and B. Raphael: "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," in *IEEE Transactions on Systems Science and Cybernetics 4*, 1968, Nr. 2, S. 100–107

[8] D. Rathbun, S. Kragelund, A. Pongpunwattana, B. Capozzi, "An Evolution based Path Planning Algorithm for Autonomous Motion of a UAV through Uncertain Environments," in *Digital Avionics Systems Conference*, 2002.

[9] H. Voos: "Entwurf eines Flugreglers für ein vierrotoriges unbemanntes Fluggerät (Control Systems Design for a Quadrotor UAV),“ in *Automatisierungstechnik*, 57, Oldenbourg Verlag, 2009, pp. 423-431.

[10] M. F. Beatty, "Principles of Engineering Mechanics, Volume 2: Dynamics," The Analysis of Motion in Mathematical Concepts and Methods in Science and Engineering, 33, Springer, 2006.

[11] S. Bouabdallah, P. Murrieri, and R. Siegwart (2004), "Design and Control of an Indoor Micro Quadrotor," in Proc. of the IEEE Int. Conf. on Robotics and Automation, New Orleans, LA, 2004, pp. 4393-4398.

[12] M. Krstic, I. Kanellakopoulos, P. V. Kokotovic, "Nonlinear and Adaptive Control Design," John Wiley & Sons, 1995.

[13] A. Holenstein, and E. Badreddin, "Collision avoidance in behaviorbased mobile robot design," *IEEE Conf. on Robotics and Automation, Sacramento, California, April* 1991, pp. 898–903.

[14] A. Patel (2012, November 15), *Heuristics* [Online]. Available: http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html

[15] MikroKopter Wiki (2013, February 20), *MikroKopter* [Online]. Available: http://www.mikrokopter.de/ucwiki/