

Approach to Swarming Behavior and Deterministic Planning Considering Motion Models

J. Holsten, S. Loechelt, T. Ostermann, D. Moormann,

Department of Flight Dynamics, RWTH Aachen University,

Wuellnerstrasse 7, 52062 Aachen, Germany, holsten@dynamik.rwth-aachen.de

Main aspects in research at the Department of Flight Dynamics are the design, modeling, simulation and controlling of autonomous UAVs (Unmanned Aerial Vehicles). Current research includes developing UAVs operating in a swarm at different scenarios, such as image acquisition or ad-hoc cellular networks, where teaming and mission planning strategies constitute an important aspect. In this contribution the influence of simple motion constraints of the simulated individuals on teaming strategies as well as global trajectory planning is investigated. The behavioral rules of swarming, such as separation, alignment and cohesion, combined with obstacle avoidance behavior and targeting, influence the motion of simple individuals, characterized by mass, extension and attitude. The motion models of real UAVs are approached by constraining the acceleration ability of the individuals in size and orientation depending on the attitude and velocity of the individuals. The influence of these constraints on the motion behavior as well as different weighting of the steering rules and obstacle avoidance parameters is investigated and discussed. Furthermore possible deterministic planning modules constrained by simplified motion models are investigated by using appropriate cost functions and a following filtering.

Introduction

In the past swarming algorithms have been investigated by computer scientists and behavioral biologists. For animation swarming was used to provide a realistic model of bird and fish swarms or flocks, whereas biologists use swarming to understand the behavior of colonies (ants, bees) in their performance of complex tasks. Results of conducted studies show, that collective complex behaviors are based on simple behavioral rules for the individual, reacting to its perceived environment. Computer scientist also use this knowledge of natural behavior to solve optimization problems, an example is the particle swarm optimization algorithm. MONTES (Ref. 1) gives an introduction to particle swarm optimization and begins by describing the origins of swarming simulation. The first and today widely known results of swarming were published by REYNOLDS (Ref. 2), whose main contribution was the introduction of three simple rules which lead to swarming. His motivation was to model swarming behavior of birds realistically in animation. Besides particle swarm optimization and the swarming rules of REYNOLDS, ant colony optimization and bee behavior inspired algorithms are used by computer scientist and engineers. GROSAN (Ref. 3) gives a summary of computational techniques that were inspired by nature for optimization problems. His paper concentrates on ant colony optimization, bees behavior inspired algorithms, particle swarm optimization and bacterial foraging algorithm. An extension of REYNOLDS' rules was done by PETERSON

(Ref. 4) who simulated a swarm of animals moving in an artificial world. Besides the three swarming rules of REYNOLDS the movement of PETERSON'S animals was constrained through a minimal velocity, maximal velocity and maximal change of velocity in absolute value and orientation. Furthermore the field of vision of the animals was limited to 180°. These additions were also used within our simulation as a comparison to behavior with more complex motion constraints. A practical study implementing the rules from REYNOLDS was done by KELLY (Ref. 5). Small robots were equipped with visual sensors to detect other robots and obstacles. The movement of the robots was guaranteed through a wandering instruction, in case no other robots evoked movement.

A different approach to steering UAVs and swarms is the deterministic steering of all individuals, by implementing trajectories to fulfill global objectives. This approach can be used to control the macroscopic motion of a swarm. Basis for global planning is a discretization of the operation and environment to allow discrete path planning and routing algorithms, e.g., Dijkstra (Ref. 6) or A*. Most planning algorithms are based on minimization of costs. Any kind of influence parameter can be accounted for by the inclusion of appropriate costs. The resultant trajectories have to be smoothed into flyable paths or motion models have to be included in the planning. Different concepts of deterministic planning is introduced in the following section. One implementation of deterministic planning for UAV swarms for different scenarios was conducted by CHENG (Ref. 7), where two main problems, the task allocation with mini-

IMAV 2010, Braunschweig Germany, 06-09 July

mal communication and ad-hoc organization as well as the complete coverage of urban structures with a limited field of view, were addressed.

For hybrid controlling each UAV needs to possess a set of simple rules to steer and to avoid obstacles autonomously. Furthermore, rules for each mission objective have to be conducted. To find effective rules and global mission planning strategies for different kinds of UAVs, the influence of the UAVs motion on the stability, efficiency and design of the algorithms has to be investigated. Later a generic flight simulation will be combined with different route planning algorithms as well as swarming behaviors to find effective behavioral rules. The resulting rules will be implemented on a swarm for a hardware in the loop simulation.

In this contribution the first steps towards combining motion models with planning and swarming behavior are introduced. First possible deterministic planning algorithms are described and their use within the area of UAVs and for multiple UAVs acting as a swarm is analyzed. Then the potential of autonomous swarming behavior and the influence of motion constraints is introduced. At last, examples of our swarming simulation with and without motion constraints are given and compared.

Deterministic Planning

In contrast to non-deterministic behavior based on biologically inspired swarming algorithms and rules as described later stands the class of deterministic path planning. Deterministic planning has the characteristic of replicable system behavior which is an advantage in safety and certification issues of UAVs. However, deterministic planning in complex environments can be demanding regarding computational resources and thus be critical for autonomously operating UAVs. There are basically two main approaches to deterministic path planning utilizing algorithms originally employed in operations research:

- graph based planning
- linear programming

Graph Based Planning

The hereby employed algorithms are taken from Graph Theory and have found a wide application in routing and operational research applications. Basis of this theory is a representation of the problem in term of vertices and edges connecting the vertices. These are associated with costs and the objective is to minimize these costs. There are informed algorithms as A* taking additional information into account and non-informed algorithms, as Dijkstra, which performs an exhaustive search guaranteeing the determination of the cost minimum path, though at a high computational effort. More detailed information on this is given by (Ref. 6).

Linear Programming

A second prominent approach is linear programming (Ref. 8), where the problems constraints and mission objectives are formulated as linear inequalities. There are different types of linear programming depending on the class of variables used. While there are real, integer and logical type linear programming, mixed integer linear programming has found the widest application in UAVs. Depending on the problem's complexity the formulation of a linear programming problem can be quite demanding and quickly reach a magnitude, where the resulting expressions are difficult to handle. Furthermore, the formulation of inequalities for each aspect within the problem can be accompanied with a high effort. By the approach of linear programming the inclusion of flight mechanics is directly possible through the definition of additional inequalities. For example in case of curve flight bank angles as well as bank rates can be limited by this approach. As linear programming is less flexible compared to graph based approaches, only graph based approaches will be focused on in the following. This is also to be seen in regard to continuously growing computational resources allowing for a wider usage of graph based methods.

Prerequisites for Deterministic Path Planning

For deterministic path planning based on graphs the environment has to be presented in discrete form. As UAVs operate in the 3D domain, a Cartesian grid or a hexagonal map with relevant height levels are suitable approaches. Maps can be a geographic representation of the environment or connected to other aspects of the problem. From these maps relevant graphs have to be deducted. Especially Cartesian grids can be easily generated with matrices.

In case of linear programming, the mission constraints, the vehicles dynamics as well as obstacles and the environment have to be formulated as linear inequalities with the corresponding variables and their limits. The solution can then be calculated by solving the system of inequalities. This system can be solved for example by the Simplex Algorithm (Ref. 8). As the problem can comprise a high number of variables to be evaluated the solutions are constructed by hyper-planes in higher dimensional domains.

If the results of the discrete search are polygon trajectories a smoothing has to be applied to allow for the consideration of flight mechanics. This is possible by the use of filters or splines.

Voronoi Graphs

Voronoi graphs are a comparably easy and intuitive formulation to avoid obstacles within an area. They are themselves no routing algorithm but allow for the generation of trajectories with maximum distance to adjacent obstacles.

Obstacles are connected to their nearest neighbors to form polygons. In the centroids of these polygons, waypoints can be set which apply for the maximum distance to the obstacles. Therefore, Voronoi graphs can be seen as a geometric generation of trajectory waypoints in a straight-forward manner (Ref. 9).

Simultaneous Localization and Mapping

The below presented methods are based on the fact, that the environment is known. Nevertheless, many UAV scenarios are located in unknown terrain and the determination of the environment itself becomes a main aspect of the planning approach. In this context the simultaneous location and mapping can be mentioned. The UAV hereby generates a map while at the same time uses reference points out of this map to locate itself. It is clear that in changing environments, this approach has its limitations. On the other hand, reference points can be used to cope with GPS out-takes and non-availability, which is not unlikely in urban environments.

Graph Based Planning Algorithms

Graph based planning and routing algorithms have gained an enormous distribution within the field of operation research and robotics, also due to continuously increasing computational power. As they are based on the abstract concept of graphs, they offer a maximum of flexibility for solving complex problems. The core of graph based algorithms, which belong to dynamic programming, are simple. Each graph consists of vertices which are connected to each other by edges. Starting from a start vertex, the objective is to reach a destination vertex under minimum costs. The concept of cost is fundamental to graph theory and therefore objectives have to be expressed as costs. Edges represent the cost for transition from one vertex to another. They can be dependent of the direction. In the easiest way they are non-directional and the graph remains unchanged and thus can be perceived to be state and time independent. This approach is especially valid in the context to operation research, where normally no dynamics have to be accounted for.

A more control based view of graphs with their three main elements conclude in the following analogies:

- vertices correspond to the system's states
- edges represent the transition between these states
- costs symbolize the effort/inertia for transition between states

In the following basic algorithms within graph theory are presented:

Dijkstra Algorithm

The Dijkstra algorithm belongs to the most prominent search and routing algorithms and was invented by the Dutch mathematician E.W. DIJKSTRA (Ref. 6). It is an exhaustive search algorithm guaranteeing to find the optimal solution. Basic idea is that for all neighbors of the investigated vertex it is evaluated whether the summed up cost to this vertex is larger than the cost that would occur by approaching it from the evaluated vertex. If this is the case, the accumulated cost for this vertex will be updated and the new best predecessor is saved. Dijkstra can only be performed with all non-negative costs. There are three data-sets for the search algorithm: The group of all vertices, the group of visited vertices for which no minimum costs have been found yet and the group of vertices, for which the minimum cost has been found. If a trajectory to a point is found, which offers lower costs than a previous trajectory, then this is taken and the relevant predecessor for the investigated vertex is updated. To find the optimal route to a destination, the predecessors are followed to the start point. As all possible combinations are searched regardless of incorporation of further information the computational runtime is higher than in the A* case. Due to this exhaustive search, the finding of the minimum cost path can be guaranteed.

A* Algorithm

The A* algorithm belongs to the group of informed search routines. A heuristic is used to search in the direction which promises minimum costs. In the case of routing problems, a simple but meaningful heuristic is to perform the search into the direction of the desired destination while evolving from the vertex with minimum cost. Due to the heuristic, not all possible routing combinations are investigated but only those applying to the heuristic. Due to this awareness the graph's magnitude decreases and the computational effort can be reduced compared to a fully exhaustive search.

D* Algorithm

While Dijkstra and A* algorithms perform on non-changing graphs where the structure of vertices and edges and the connected costs are maintained, the D* algorithm as derivative of the former two ones can be used in changing environments, which applies much more in the sense of autonomous UAVs. Changed conditions within the graphs are recognized and used for an update of the cost computation. Result is a cost optimal path under consideration of the heuristic leading from the start to the destination vertex in a changing environment directly related to a changing graph.

Receding Horizon Search

Receding horizon search is an approach where the search is performed for different subsets of the problem dimension for optimization of an objective function. Furthermore, the search can be performed for different scales of detail for the problem. It can be well illustrated by going from an address in City A to an address in City B. In context to routing, the first step would be to get from city A to B. When arriving there, the correct district of the city has to be found. When this is done, the correct street has to be found and there the right number. As consequence the planning can be divided into 4 segments with increasing level of detail. Receding Horizon Search is an optimization approach which is also sometimes denoted as Model Predictive Control in context to control theory. By adjusting the input sequence to a system, the system's output can be optimized (Ref. 10). Here the planning horizon comprises only a subset of the overall problem set. This subset can be time related or related to the resolution of different detail levels. Due to this limitation, a global optimum is normally not found.

Planning for Single UAV

The above presented algorithms have been widely investigated in context of robotics and unmanned vehicles and thereby in particular for UAVs (Ref. 11). A main problem is the introduction of motion models accounting for the dynamic properties which influence flyable trajectories. Especially in case of fixed-wing UAVs minimum speed and curve radii are important considerations. A further consideration is the ability of routing algorithms to run autonomously in real time on the UAV's processing hardware. Furthermore static scenarios leading to fixed vertices and edges are a simplification of in general time variant and dynamically changing scenarios. The problem is to find robust planning strategies that can cope with changing environments and situations and update in the case that changes have influence on the so far generated paths. For motion models the selection of neighboring vertices is dependent on the previous actions and states. The graphs become thus time and state dependent and their complexity increases according to the number of considered states (Ref. 12), (Ref. 13). A good example in connection with UAVs is the change from a left curve into a right curve and the associated delay due to bank dynamics. This delay has to be accounted for in the planning algorithm by reducing the set of reachable neighbors between two discrete steps. Furthermore, discretization has to be adapted to allow for meaningful consideration of flight maneuvers covering multiple discrete planning steps.

Aspects for Deterministic Team Planning

In addition to the above mentioned aspects for single UAVs multiple UAV operations impose further strong implications

on the planning process and the systems needed to allow for it in a real world environment. Coordination in spatial and temporal domain imposes the need for information exchange between the operating agents within the swarm (Ref. 14), (Ref. 15), (Ref. 16). In general there are centralized approaches with a leading element within the swarm or distributed approaches for coordination between equal team members. Latter approach offers better robustness to failure of single members but also requires a higher coordination effort. For graph based planning, the graphs become even more complex and time/state-dependent by the inclusion of additional participants. Currently, there is a strong research for this kind of highly complex optimization and routing issues. In respect to control theory, this multi UAV coordination puts another control loop into action with consequences for controllability and stability (Ref. 12).

Swarming Behavior

Swarming behavior in contrast to deterministic planning has more random elements. To investigate the swarming behavior especially for the use in UAVs a simulation based on the rules of REYNOLDS was build. The three main rules implemented are the

- separation
- alignment
- cohesion

of each individual. Each individual was furthermore equipped with a field of view of 120° relative to its direction of velocity. In the simulation each velocity component can be scaled with its weighing factor $w_{s,a,c}$.

The separation aims to avoid collisions of the individuals and to assure a minimal distance within the swarm. The influence of the separation rule should decline with increasing distance and have a maximum at zero distance. To avoid problems due to singularities the hyperbolic secant was chosen over a hyperbola. The distance between the individuals was scaled to achieve benign behavior. For a weighing factor $w_s = 1$ the resulting speed was constrained to the maximal allowed velocity $v_{max} = 40$. The complete calculation is given in Equation 1, with n being the number of individuals and $\vec{x}$ the position.

$$\vec{v}_s = \sum_{i=1, i \neq k}^n \left(-v_{max} \cdot \text{sech}(0.1 \cdot \|\vec{x}_i - \vec{x}_k\|) \cdot \frac{\vec{x}_i - \vec{x}_k}{\|\vec{x}_i - \vec{x}_k\|} \right) \cdot w_s. \quad (1)$$

The alignment rule provides collective swarm behavior. The individuals match their speed to their neighbors, leading to a almost homogeneous movement of the complete swarm in one direction. In this case only the neighboring individuals within the own field of view are considered.

Equation 2 gives the relation of the alignment velocity. In this case n represents the number of individuals within ones field of view.

$$\vec{v}_a = \sum_{i=1, i \neq k}^n \left(\frac{\vec{v}_i}{n} \right) \cdot w_a. \quad (2)$$

The cohesion rule assures the movement towards a swarm and keeps the individuals within the swarm. A force is exerted on the individuals to move towards the mass center of it's immediate neighbors. The larger the distance towards the relative mass center the larger the induced velocity. To avoid over proportional rates of the cohesion rule compared with the other rules, the arctangent function was chosen. It was scaled to v_{max} for a distance of infinity in case of $w_c = 1$.

$$\vec{v}_c = \frac{v_{max}}{\pi/2} \cdot \arctan(\|\vec{x}_{mc} - \vec{x}_k\|) \cdot \frac{\vec{x}_{mc} - \vec{x}_k}{\|\vec{x}_{mc} - \vec{x}_k\|} \cdot w_c. \quad (3)$$

The weighing of the rules can be changed through adaption of the parameters w_s , w_a and w_c . For each step the calculated velocities are added to the old velocity and the result is scaled to the mean length of all components. Afterwards constraining factors such as maximal or minimal velocity and maximal acceleration are considered. These constraints are described and compared to constraints resulting from basic motion models in the following sections.

Obstacles and Targets

The described basic swarming simulation was extended by obstacle avoidance and target following algorithms. A target can be set optionally and exerts a velocity towards itself on all individuals. The velocity induced by the target is calculated analogue to $\vec{v}_c$ using the arctangent function.

$$\vec{v}_t = \frac{v_{max}}{\pi/2} \cdot \arctan(\|\vec{x}_t - \vec{x}_k\|) \cdot \frac{\vec{x}_t - \vec{x}_k}{\|\vec{x}_t - \vec{x}_k\|} \cdot w_t. \quad (4)$$

For obstacle avoidance different methods were analyzed. Most common, and also introduced by REYNOLDS (Ref. 17), is a distance proportional force field. Around each obstacle a force field induces a velocity away from the obstacle. This methods has difficulties with perpendicular approaches, since the approach velocity is only reduced but not redirected. Therefore a different method was chosen. If the through swarming and targeting rules calculated velocity points towards the obstacle and the distance is small, the calculated velocity is projected into the surface plane of the obstacle. Afterwards the projected velocity is scaled using the hyperbolic secant function and a weighing factor w_o . The induced velocity tangent to the

obstacle leads to a redirection of the flight path around the obstacle.

$$\vec{v}_{proj} = ((\vec{x}_o - \vec{x}_k) \times \vec{v}_k) \times (\vec{x}_o - \vec{x}_k) \quad (5)$$

$$\vec{v}_{o,0} = \frac{\vec{v}_k \cdot \vec{v}_{proj}}{\|\vec{v}_{proj}\|^2} \cdot \vec{v}_{proj} \quad (6)$$

$$\vec{v}_o = \vec{v}_{o,0} \cdot v_{max} \cdot \text{sech}(0.1 \|\vec{x}_o - \vec{x}_k\|) \cdot w_o. \quad (7)$$

The induced velocities due to a target and obstacles are added to the velocities induced by the swarming rules and scaled together.

Motion Constraints

Two different models to constrain the calculated velocities were implemented and tested. For both the new velocity for each step was calculated and then adapted according to the constraints. The first set of constraints consists of the following parameters:

- maximal velocity
- minimal velocity
- maximal acceleration

In this model only the absolute value of the velocity and its change is considered and constrained. The resulting trajectories are smooth for the most parts, since the previous velocity is considered in each step. For small velocities abrupt changes of direction can occur. The results will be discussed in more detail in the following section.

The second model uses the velocity to approximate an attitude akin to a real plane and derives constraints for the next velocity out of this attitude. Since this is only the first step towards a real planes motion model many assumptions were presumed. The attitude of a plane can be described by the three Eulerian angles

- bank angle Φ ,
- pitch angle Θ ,
- azimuth angle Ψ .

Only stationary states were considered and a complete sideslip free flight as well as coordinated curves were assumed. With these assumptions the path angles χ and γ can be derived out of the given velocity components. Due to the assumed sideslip free flight the azimuth angle equals the path azimuth $\Psi_j = \chi_j$. The bank angle for each step j results from the previous rate of turn and velocity according to Equation 8. Furthermore it is constraint to $\Phi \in [-20^\circ, 20^\circ]$.

$$\Phi_j = \arctan\left(\frac{\dot{\Psi}_j \cdot \|\vec{v}_j\|}{g}\right). \quad (8)$$

The pitch angle depends on the velocity as well as the rate of climb. In a first approximation the pitch angle can be assumed to equal the sum of the angle of attack α and the path angle,

$$\Theta_j = \gamma_j + \alpha_j, \quad (9)$$

with α dependent of the velocity for stationary flight. For increasing velocities the α decreases rapidly. For stationary flight states and small angles of attack the needed lift L approximately equals the weight G and is constant ($L \approx G = \text{const.}$). The lift is proportional to the angle of attack and proportional to the square of the velocity ($L \propto \alpha$, $L \propto v^2$). To maintain a constant lift a second degree hyperbolic relation between α and v can be assumed. To adapt the relation to values found in numeric simulations of the planned wing, the following terms were included.

$$\alpha_j = \frac{50}{(\|\vec{v}_j\| + 0.75)^2} [^\circ] \quad (10)$$

The feedback of the attitude on the possible next velocity occurs in three different ways. A change of thrust can be assumed to only lead to a change in horizontal velocity ($\vec{v}_{xy} \propto T$). Since the thrust $T \in [T_{min}, T_{max}]$ ranges within limited values, the change of horizontal velocity is constrained as well.

The second constraint can be derived from coordinated curve flight, where the rate of turn and thus the azimuth rate $\dot{\Psi}$ is directly proportional to the bank angle ($\dot{\Psi} = \dot{\chi} \propto \tan \Phi$), where $\Phi \in [-20^\circ, 20^\circ]$ is limited and its rate of change $\dot{\Phi}$ is limited as well.

The third constraint is the sink/climb rate, which depends on the pitch angle and can be assumed to be proportional to it ($\dot{H} \propto \Theta$). Hereby the pitch angle needs a finite time to change itself, which leads to a constraint in $\dot{\Theta}$. Furthermore the pitch angle is constrained in magnitude depending on the aircraft type, $\Theta \in [\Theta_{min}, \Theta_{max}]$.

The velocity for each step is first calculated analogue the basic swarming rules without constraints due to motion models. Afterwards the change in horizontal magnitude, the turn rate and vertical speed are checked to be consistent with the previous attitude. Small deviations are allowed to compensate for the unsteady flight states and thus to allow changes of the stationary flight states. The results of simulated scenarios with a basic motion model in form of constrained maximum velocity values and the slightly extended motion model are described in the next section.

Results

The simulation was run several times with different weighing of the induced velocities. An overview of the described cases is given in Table 1. The cases are grouped into two categories, *Boids* when using the simple velocity constraints and *Planes* when using the attitude model. Figures 1 to 5 show the observed behavior. Each colored line

represents one individual. The dotted lines show the projection into the horizontal plane.

Type	Cohesion w_c	Separation w_s	Alignment w_a	Target w_t
Boid	0.1	1.5	0.3	0.1
Boid	0.75	1.5	0.3	0.1
Boid	0.2	1.3	0.1	0.3
Plane	0.2	1.3	0.1	0.3
Plane	0.2	1.3	0.1	0.3

Table 1. Overview of the discussed cases. The obstacle weight is constant in all cases ($w_o = 0.6$).

The first three cases analyzed use the simple motion constraint. In the first case, illustrated in Figure 1, the rules inducing swarming behavior were weighed low compared to keeping the distance and flying towards the target. This results in the fact, that as the first Boid reaches the target, the velocity component to keep the distance towards this Boid outweighs the targetting of the other Boids.

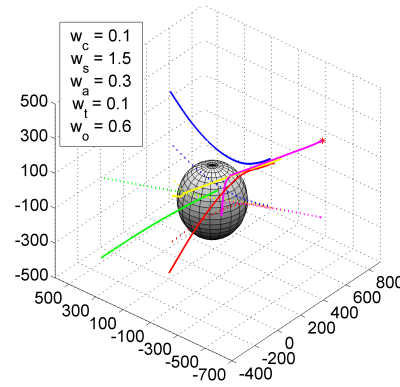
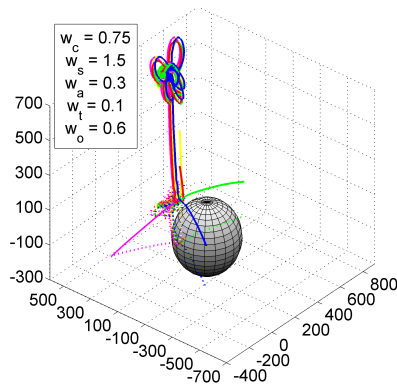
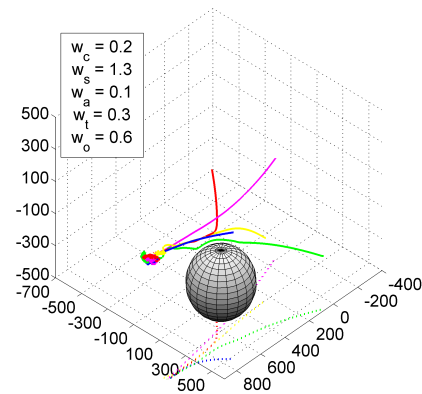
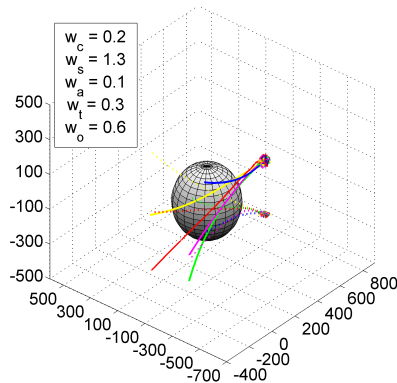


Fig. 1. Low weighing of the swarming rules.

Case 2 depicts the opposite behavior. In this case the swarming rule, especially the cohesion towards the mass center outweighs the other rules. The Boids first fly towards each other to build a swarm and then, as a collective, fly towards the target where they continue circling, as illustrated in Figure 2.

The third case illustrates a more leveled behavior, where all Boids orient themselves towards the target and circle there as a collective, Figure 3.

The last two cases, which are depicted in Figure 4 and 5, use the extended motion model. Especially at the end of the simulation, when reaching the target it can be seen, that the Planes start to circle with a defined minimal radius, whereas in case of the Boids sharp turns are observed. In the fourth case a flight around the obstacle occurs and as can be seen

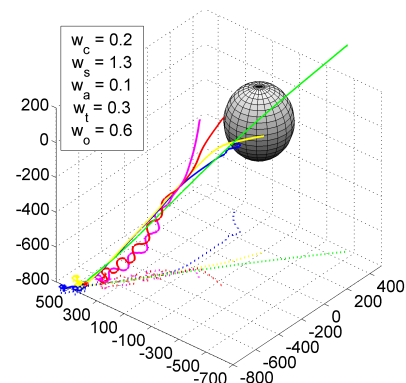
**Fig. 2. High weighing of the swarming rules.****Fig. 4. Advanced motion model.****Fig. 3. Balanced weighing of the swarming rules.**

in Figure 4 the trajectory is redirected parallel to the surface of the sphere. In the fifth case a slight oscillation, due to the feedback of the attitude can be seen. Two planes start circling each other with a defined radius while approaching the target.

For all five cases the smallest distance between two individuals and the distance towards the obstacle was controlled. The distance between two individuals is depicted in Figure 6. In all cases a minimal distance stabilizes itself, which mainly depends on the weighing of cohesion and separation. The time until this distance is reached depends on w_c . In the first case $w_c = 0.1$ and the minimal distance only occurs at the end of the simulation. In the second case for $w_c = 0.75$, the time towards minimal distance is shortest, whereas the other three cases with $w_c = 0.2$ lie in between but close to each other. The distance towards the obstacle is in all cases large enough.

Conclusion

An overview of different deterministic approaches as well as swarming behavior was given and analyzed according its

**Fig. 5. Advanced motion model showing an oscillation between two planes.**

usability in UAV swarms. The structure of a swarming simulation using different motion models was introduced and results were depicted. The results show a slight difference between motion models of different complexity and different weighing of the parameters, especially the constraint on the azimuth rate supports flyable curves. Future investigation will include the analysis of the stability of the feedback of the attitude as well as the behavior close to the target. Next steps will adapt the simplified motion model to the generic simulation to include unsteady flight states in the planning. An objective is to allow investigation of the influence of different known airplane motion models on swarming behavior. Additionally the hybrid combination of deterministic planning with swarming behavior will be investigated.

Nomenclature

n	Number of individuals
T	Thrust
$\vec{v}$	Velocity

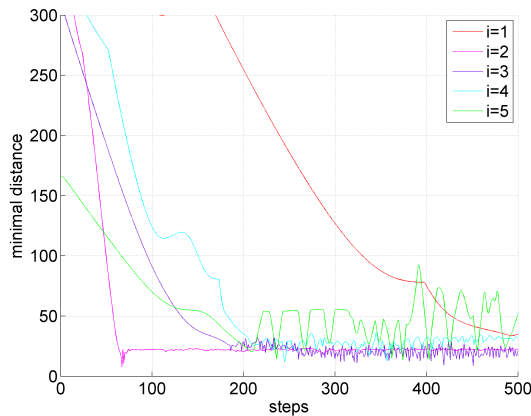


Fig. 6. Minimal distance between two individuals.

$\vec{v}_s$	Velocity due to separation
$\vec{v}_a$	Velocity due to alignment
$\vec{v}_c$	Velocity due to cohesion
w_a	Alignment weight
w_c	Cohesion to mass center weight
w_o	Avoid obstacle weight
w_s	Separation weight
w_t	Target weight
$\vec{x}$	Position

i	Other individual, case
j	Step
k	Current individual
mc	Mass center
obs	Obstacle
$proj$	Projected
t	Target

Φ	Bank angle
Θ	Pitch angle
Ψ	Azimuth angle

References

- ¹M. A. M. de Oca, "Particle swarm optimization," May 7, 2007.
- ²C. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," 1987.
- ³A. Grosan, "Stigmergic optimization: Inspiration, technologies and perspectives."
- ⁴A. R. M. G. D. Peterson, "Animal aggregation: Experimental simulation using vision-based behavioural rules," *Lectures in Complex Systems*, pp. 623–630, 1993.

⁵D. Kelly and A. Keating, "Flocking by the fusion of sonar and active infrared sensors on physical autonomous mobile robots."

⁶J. L. Gross and J. Yellen, "Graph theory and its applications," in *CRC Press, Boca Raton, FL*, 1999.

⁷P. Cheng, J. Keller, and V. Kumar, "Time-optimal uav trajectory planning for 3d urban structure coverage," in *ICRA Workshop*, 2008.

⁸F. Hillier and G. J. Lieberman, "Introduction to operation research," in *McGraw-Hill, New York*, 8th edition, 2009.

⁹S. A. Bortoff, "Path planning for uavs," in *Proceedings of the American Control Conference, Chicago, Illinois*, June 2000.

¹⁰C. G. Cassandras and W. Li, "A receding horizon approach for dynamic uav mission management," in *Enabling technologies for simulation science. Conference No 7, Orlando, FL*, 2003.

¹¹K. M. A. Chaudhry and R. D'Andrea, "Low observability path planning for unmanned air vehicle using mixed integer linear programming," in *43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.

¹²M. Mesbahi, "On state-dependent dynamic graphs and their controllability properties," *IEEE Transactions on automatic control*, Vol. 50 No. 3, 2005.

¹³A. Khera and A. Abate, "Dynamic graph analysis and optimal control -dynamic programming approaches to safety," 2004.

¹⁴A. P. I. Kaminer, O. Yakimenko and R. Ghabcheloo, "Path generation, path following and coordinated control for time-critical missions of multiple uavs," in *Proceedings of the 2006 American Control Conference Minneapolis, Minnesota, USA*, 2006.

¹⁵T. S. J. Sousa and P. Varaiya, "Task planning and execution for uav teams," in *43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.

¹⁶J. T. D. G. L. Marsh, G. Calbert and H. Kwok, "Multi-agent uav path planning," in *MODSIM05 Conference, Melbourne*, 2005.

¹⁷C. Reynolds, "Not bumping into things," Notes on "obstacle avoidance" for the course on Physically Based Modeling at SIGGRAPH 88, August 1988.