Dynamic Trajectory Replanning for Unmanned Aircrafts Supporting Tactical Missions in Urban Environments

Lukáš Chrpa and Peter Novák

Agent Technology Center
Department of Cybernetics, Faculty of Electrical Engineering,
Czech Technical University in Prague
Czech Republic
chrpa@labe.felk.cvut.cz, peter.novak@fel.cvut.cz

Abstract. In the last decade we witnessed an increased demand for employment of unmanned aerial vehicles (UAV) in practise. For instance, there is a growing need to provide surveillance tasks in a given area by a team of cooperating UAVs. In this case, the ability of a single UAV to plan its course of actions (e.g., trajectories that the UAV must fly through) is essential. Trajectory planning algorithm used by UAVs must be able to find trajectories satisfying constraints given by environment (e.g., obstacles) or by UAVs' dynamic models. Besides the planner itself the UAVs must somehow react to changes of high-level tasks or environment. Such a reaction often means to replan the trajectories towards new goals. In this paper, we will discuss the replanning related issues such as swapping the old and new trajectory smoothly respecting the UAV dynamics. We present an idea based on estimating running time of replanning tasks and evaluated its impact to safeness of replanning (e.g., avoiding to get to an inconsistent state).

1 Introduction

With the technological advancements and maturing of hardware in aerial robotic systems in the last decade, we witness a growing demand for employment of unmanned aerial vehicles (UAV) in practise. Scenarios, such as disaster relief management or support of tactical ground military operations often include a need to provide continual situational awareness and/or information collection support by teams of autonomous cooperating aircrafts [6]. The structure of the relevant operational environments for such missions is however often unsuitable for use of conventional-take-off-and-landing aircrafts (CTOL) with fixed-wing design. With their abilities to perform well also in low operational altitudes and velocities, vertical-take-off-and-landing vehicles (VTOL), i.e., helicopter-type aircrafts, provide an agile aerial platform for operations in structured environments, such as urban terrain. Additionally, they are capable of navigation between relatively complex ground structures, such as buildings and mountainous regions.

Along the line of research aiming towards implementation of flexible navigation and collision avoidance algorithms, a set of trajectory planning algorithms for CTOL aircrafts [7] has been developed. Due to their properties and constraints tailored for fixed-wing aircrafts, these are however unsuitable for VTOL trajectory planning. The particular requirements on an ideal trajectory planner resulting from the need to operate VTOL-type aircrafts in complex (urban) environments include the ability to plan through a sequel of way-points (together with their corresponding flight vectors and speeds) with variable speeds respecting the actual aircrafts manoeuvrability according to a model of the aeroplane's physical dynamics.

Recently, in [2] we introduced the details of planning algorithms suitable for trajectory planning for VTOL-type aircrafts. In the next section (Section 2), we briefly describe the considered trajectory planning algorithms. At the core of the presented paper, firstly, in Section 3, we sketch the overall command and control (C2) system for control of tactical ground military missions in urban environments. Subsequently, in Section 4, we discuss issues stemming from the requirement to support dynamic task allocation and replanning of aircraft's trajectory resulting either from a command received from the C2 interface, or emerging from the external dynamics of the environment. An example of the latter is, for instance, the need to adapt the aeroplane's trajectory according to the movements of a mobile target the aircraft is actively tracking and observing with the set of its on-board sensors. In Section 5, we conclude the discourse of the paper with a discussion of the integration of the planner with the C2 interface and provide experimental results comparing the original CTOL-specific planner with the planner proposed to include also support of VTOL-type aircrafts. Finally, we outline the on-going and future work along the discussed line of research.

2 Trajectory planning for unmanned aerial vehicles

Trajectory planning is an essential part of capabilities of an autonomous unmanned aerial vehicle. In the following, we briefly describe two trajectory planning algorithms *Accelerated A** [7] and *Augmented A** [2, 1] suitable for CTOL and VTOL-type aircrafts respectively. The *Augmented A** algorithm is specifically implemented to respect aircraft's speed limit constraints during the planning process.

Trajectories which UAVs fly through can be (for model simplification) combined from primitive manoeuvres. These refer to actions as known from the planning theory. We have three kind of these primitive manoeuvres: Straight, Turn and Pitch. Straight Manoeuvre forms a line segment defined by its position, direction and length. Turn Manoeuvre forms an arc segment defined by its position, radius and angle. Pitch Manoeuvre is used for moving the UAV to upper or lower flight level. Pitch Manoeuvre forms sigmoid-like curve segment defined by its position, height and length.

2.1 Accelerated A*

*Accelerated A** algorithm [7] was introduced as a trajectory (path) planning method based on chaining primitive manoeuvres in continuous space. It is an adaptation and extension of the well known *A** algorithm [4]. The core features of the *Accelerated A** are 1) discretization of the state space by considering a finite set of manoeuvres the particular aircraft is able to perform with respect to the extreme capabilities of the aeroplane, and 2) the use of adaptive sampling of the continuous 3-dimensional space.

The former technique basically boils down to considering only manoeuvres, the aircraft is capable to perform in the full range of speeds it can achieve. Note, that even though a fixed-wing aircraft can slow down to some minimal speed and thus achieve a minimal turning radius, the *Accelerated A** algorithm considers only the maximal turning radius corresponding to the maximal speed of the aircraft. This is a result of the abstraction from the model of the physical dynamics of the aircraft. As a consequence, the UAV is capable to perform every considered manoeuvre regardless of the current speed it has at the point of entering the manoeuvre.

The basic idea behind the adaptive sampling approach rests in changing the density of the generated and expanded states according to the distance of the current state to the nearest obstacle. Roughly speaking, lengths of the generated primitive manoeuvres are decreasing with the decreasing distance of the aircraft's position to the nearest obstacle such as a no-flight zone or a ground terrain feature.

The calculated trajectory is finally smoothed in a post-processing step so that the curves become more realistic with respect to the turning radii of the particular aircraft. Nevertheless, minimal turn and pitch radii must be kept even after the smoothing. In result, the disadvantages of the of the algorithm are that i) due to the abstracting away from the physical dynamics, a real aircraft could perform a much richer range of manoeuvres than the planner is able to consider; and ii) the planner is not able to produce a viable plan for an extremely structured environment, even though a flexible aircraft, such as a VTOL, would be able to fly through it. On the other hand, the advantages include producing realistic looking paths in quite low computation time. In consequence, this algorithm is more suitable for fast aircrafts flying in free space with few obstacles, i.e., CTOL-type vehicles in free airspace.

2.2 Augmented A*

In [2], we introduced a trajectory planning method which, unlike the *Accelerated A** algorithm, consider also speed limit constraints given by the physical dynamics of the particular UAV. The algorithm is based upon state space discretization in terms of stacked hexagonal grids and uses the standard *A** as a main planning procedure. The planning algorithms is augmented so that that in every node, determined by the cell the aircraft is currently positioned within the hexagonal grid and the direction of UAV heading, it considers an interval of the aeroplane's feasible speed values. In [2], we proposed two ways how these intervals can be adjusted.

limitation: when a UAV performs a *turn* or a *pitch* manoeuvre from a node s_1 to a node s_2, the speed limit constraint must be applied. The constraint can be expressed as an interval $\langle v_{lim}^-, v_{lim}^+ \rangle$. Provided an interval of feasible speed values $\langle v_{s_1}^-, v_{s_1}^+ \rangle$ in the node s_1, the resulting interval of feasible speed values in s_2 is calculated as an intersection of these intervals, i.e., $\langle v_{s_1}^-, v_{s_1}^+ \rangle \cap \langle v_{lim}^-, v_{lim}^+ \rangle$. If the intervals are disjoint, then the considered manoeuvre is inapplicable. In order to simplify the model, speed adjustments during *turn* or *pitch* manoeuvres are not considered.

expansion: when a UAV performs a *straight* manoeuvre from a node s_1 to a node s_2, the interval of feasible speed values $\langle v_{s_1}^-, v_{s_1}^+ \rangle$ corresponding to the node s_1 must be adjusted. Let $acc(v, a, s)$ be a function computing speed resulting from uniformly

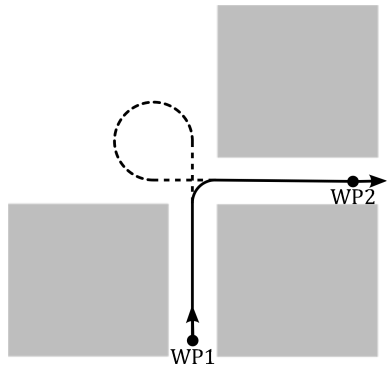

Fig. 1. Illustration of two possible trajectories between WP1 and WP2. Dash line illustrates a situation when minimal possible turn radius is too big and the trajectory forms an 'ear'.

accelerated motion, where v is an initial speed, a is an acceleration (resp. deceleration if negative) and s is a distance. The interval of feasible speed values $\langle v_{s_2}^-, v_{s_2}^+ \rangle$ in the node s_2 is computed as follows:

$$v_{s_2}^- = \max(v_{min}, acc(v_{s_1}^-, a^-, s)) \tag{1}$$
$$v_{s_2}^+ = \min(v_{max}, acc(v_{s_1}^+, a^+, s)) \tag{2}$$

where v_{min}, v_{max} stand for the minimal and maximal speed, a^-, a^+ denote the minimal and maximal acceleration (resp. deceleration if negative), and s is the distance between s_1 and s_2.

The adjustment approach provides intervals of feasible speed values with respect to the trajectory between the given nodes. Observe, that according to the planner, it is possible that the UAV would at certain node reach the speed of 100 km/h and then immediately perform a turn manoeuvre which in fact can be only performed at the speeds less than 50 km/h. Obviously, 'gaps' would occur in the speed intervals on subsequent trajectory segments. These 'gaps', however, can be relatively easily removed by back-propagating the speed intervals, e.g., by instructing the aircraft to slow-down before the *turn* manoeuvre. More involved description of the algorithms for speed interval adjustments and back-propagation can be found in [2].

Advantages of the *Augmented A** planning technique rest in producing feasible trajectories for UAVs respecting the model of their physical dynamics. As depicted in Figure 1 Augmented A* allows the UAV to slow down (if possible) to perform an appropriate Turn manoeuvre contrary to Accelerated A* where the UAV must go slowly all the way or perform an 'ear-like' Turn manoeuvre (dash-lined). The main disadvantage of this planning method, on the other hand, is that due to the state space discretization to stacked hexagonal grids the resulting paths tend to be less realistic paths and the computation time is higher as well. *Augmented A** is further studied and discussed in [1].

Fig. 2. Example snapshots of the C2 interface (left) and an on-going mission simulation in the TAF2 simulator (right).

3 Mission-centric information collection

TACTICAL-AGENTFLY 2 (TAF2) is an experimental multi-agent simulation system we have developed as a part of a larger on-going project initiative aiming at investigation of issues and development of control and planning algorithms for unmanned aerial and ground assets that are engaged in ISTAR (*Intelligence, Surveillance, Target Acquisition, and Reconnaissance*) operations. The TAF2 system facilitates *execution of configurable missions* carried out by a set of aerial and ground assets in an operations theatre. The TAF2 command and control (C2) subsystem allows for a single operation officer to control a number of aerial assets. At any moment, the officer can issue a batch of tasks to the team of cooperating UAVs which subsequently allocate the tasks among themselves and perform them in a cooperative manner, possibly supporting and replacing each other when necessary. In result, the TAF2 system provides a platform for extensive testing and evaluation of various implemented coordination strategies for a team of unmanned aerial vehicles (UAVs) supporting a ground-mission by means of providing common operational picture, area surveillance and on-demand tracking of mobile targets.

From the point of view of the individual robotic aircrafts, on an abstract level, the distributed task allocation mechanism is capable to calculate a sequel of waypoints each of the aircrafts should visit and provide these to the lower level plane's autopilot module. The task allocation mechanism, together with the higher level C2 subsystem is responsible for the *task decomposition*, *task allocation*, *monitoring* of the *task accomplishment* and *result synthesis*. Figure 2 provides a screenshot of the TAF2 C2 interface (left), together with a visual snapshot of an on-going TAF2 mission simulation (right). The C2 interface shows two user-defined surveillance areas and their split between the UAVs. In addition, a user of the interface can see the positions of the individual UAVs as well as other points of interests such as enemy soldiers.

As a consequence of the TAF system's layered architecture, the tasks can change at any time during the mission execution. Additionally, when performing higher level tasks depending on external factors such as tracking of a mobile target, the UAVs must be able to rapidly adapt their flight trajectories accordingly. As a result of these requirements, there is a need for supporting frequent dynamic replanning of the trajectories during the flight. In the remainder of this paper, we focus on the issues of stemming from implementing this feature in the UAV trajectory planner based on the *Accelerated A** and *Augmented A** algorithms.

4 Trajectory replanning

Trajectory replanning is necessary when goals of a particular UAV change either due to the high-level mission control, or because of some external factors. A straightforward strategy in such a case could be simply to drop the current plan, calculate a new one and proceed. However, while a plan is being executed, it might not be possible to easily stop the plan execution and wait until the new plan is provided. Such a nave strategy could be potentially dangerous as the aircraft under no circumstances should lose control over its trajectory as it is impossible to change UAV's location or heading or stop it instantly. Furthermore, a CTOL-type fixed-wing UAV cannot be stopped at all. Therefore a careful replanning of the UAV's trajectory following its physical dynamics must be incorporated into the planner-to-controller interface. The idea we present in following is based on estimation of the time when the current plan is replaced by the new plan. This replacement must be done in such a way that the transition to the new plan proceeds smoothly, i.e., the UAV must avoid to get into an inconsistent state (e.g., must not instantly change its heading direction, speed, or even position).

We use a physical definition of distance as a (partial) function (s) transforming time into Euclidean space and a velocity function (v) as a first derivative of the distance function:

$$s : \Re_0^+ \rightarrow \Re^3, v(t) = \frac{s(t)}{\mathrm{d}t}$$

Obviously, the distance function results from the trajectory planners. Accelerated A* comes with sequences of manoeuvres that can be explicitly described by curves (see Section 2). In this case the speed is constant, thus it is not difficult to add a time component. Augmented A*, on the other hand, produces a list of way-points accommodated with current direction and speed value. Time component can be also easily computed. To provide a continuous distance function, way-points must be interpolated linearly or by using PID[1] controllers to provide more realistic behavior of UAVs [2].

Let us have a distance function s (and its derivative v) representing a trajectory the planner produced and which is currently being executed. If a replanning request is delivered at time t_0, then the planner must provide a new trajectory represented by a distance function s' (and its derivative v') at time t such that $t > t_0$ $s'(t) = s(t)$ and $v'(t) = v(t)$. Informally said, the time t is a 'swap point' between the old and the new plan. However, for the replanning purposes it is necessary to know a position of the 'swap point'. It must be passed to the planner as an input way-point. Let us define *an unchangeable offset* σ as a time shift between the replanning request time t_0 and the 'swap point' time t, i.e., $t = t_0 + \sigma$. For illustration, see Fig. 3.

[1] proportional–integral–derivative

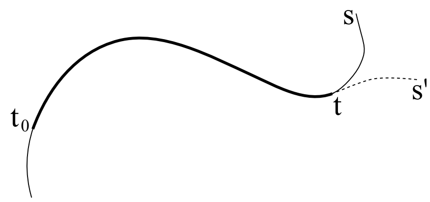

Fig. 3. Example of a replanning request which is delivered at time t_0. The old trajectory s is replaced by a new trajectory s' (dash-lined) at time t.

It is clear that the replanning procedure must be performed within the time defined by the given unchangeable offset σ. If not, it may happen that the UAV will get into an inconsistent state. This can happen for instance if the new plan arrives after the UAV already crossed the 'swap point', i.e., it continues executing of the old plan, the aircraft might be already at a wrong position with respect to the new plan. Let us define *a safeness function* $\Psi : \Re_0^+ \rightarrow \langle 0, 1\rangle$ which assigns every (non-negative) unchangeable offset $\sigma \in \Re_0^+$ a probability value of finishing the replanning task within the time defined by σ. Obviously, the safeness function is a monotonically increasing function, $\lim_{\sigma\rightarrow 0} \Psi(\sigma) = 0$ and $\lim_{\sigma\rightarrow\infty} \Psi(\sigma) = 1$.

For practical reasons it is impossible to set σ to infinity. The approach must balance between two aspects. Firstly, if σ is too low, $\Psi(\sigma)$ would be too low as well and the replanning task would often fail to be completed within the time defined by σ. Secondly, if σ is too high, it may take too long for the UAV to start executing the new plan (e.g., to react to the high-level task change). In result, the main problem is to determine the value of the unchangeable offset σ in such a way that it either maximizes $\Psi(\sigma)$, or minimizes σ. Ideally, we should find σ, such that $\Psi(\sigma) = 1$. This problem is also discussed in [3].

The main issue rests in the fact that we do not know the exact course of the function Ψ. It depends, for example, on the environment, planning method, CPU performance or expected load, etc. Obviously, we cannot compute the exact course of Ψ, but we can at least estimate it. The estimation can be done empirically by evaluating running times of a set of planning problems, i.e., $\Psi(\sigma) = \frac{|\{p \mid p \in problems, \text{ where the running time } t \leq \sigma\}|}{|problems|}$. These problems must be solved with respect to certain environment, by a certain planning method, on certain CPU, etc. that will occur in a certain 'real-world' application.

Besides estimating safeness functions, there exist other potential approaches how to handle replanning issues. One of them is based on the fact that if a replanning request arrives, the UAVs will reactively stop (helicopter-like assets) or perform waiting loops (aircraft-like assets) before adopting the new plan. The approach is safe, assuming the waiting loops are performable or the UAV can stop before the plan ends. On the downside, the UAVs' behaviour might look a bit sloppy, since they must stop or perform at least one waiting loop on every replanning request.

In the case when the planner is running out of time (defined by the given unchangeable offset), an alternative idea would be to provide the UAV at least a partial plan (to gain additional time) and proceed with another replanning request (to fulfill the given goals). However, the partial plan might be too short or might end in a 'dangerous' position (for instance, right before an obstacle), so the approach is also not ideal.

5 System integration and experimental evaluation

5.1 Implementation

The implementation is based on the concept of multi-agent simulation system TAF2 discussed above in Section 3. The high-level tasks define areas of interest where an operator desires to perform surveillance activity, i.e., detect presence and positions of moving ground targets such as enemy soldiers. During the surveillance, the UAVs must avoid potential obstacles, which are known *a priori*. Surveillance tasks are carried out jointly by a team of CTOLs and VTOLs. Task allocation and task decomposition to individual UAVs is done as discussed in Section 3. For illustration see also Figure 2. Every UAV plans its own zig-zag trajectory [5] on the surveillance area assigned to it. In this case, the planes do not calculate whole trajectories but only a sequence of way-points determining the rows of the zig-zag path. Accelerated A* algorithm (cf. Section 2.1) is used as the planning approach for CTOLs, while VTOLs employ the Augmented A* algorithm (cf. Section 2.2). Collision avoidance problems are not taken into account, since the UAVs are assumed to operate on unique flight levels.

Replanning requests arise in two cases. Firstly, the UAV must replan when its task changes (e.g., surveillance area specification changes). For performance reasons, the UAVs plan their trajectories only for next few way-points. Clearly, at some point it is necessary (in fact, when the next way-point on the path is reached) to extend the pre-planned trajectory further. In the implementation, we handle this issue by sending replanning request to the affected UAV.

The replanning problem (discussed in detail in Section 4) is dealt with by estimation of the unchangeable offset, i.e., the time-window in which replanning of plane's tasks must be performed. To set a proper unchangeable offset value we estimate a corresponding safeness function. The estimation was done by experimental evaluation of a set of random test-bed planning problems (see below). In order to provide realistic conditions of the developed application, we had to take into account the relevant physical environment size, positions and sizes of obstacles, the number of CTOLs or VTOLs in the task force, CPU performance, expected load, etc.

For the experimental evaluation we use an urban area environment where two CTOLs and two VTOLs performed random surveillance tasks. We measured running times of planning tasks (more than 1000 in total) that arise as (primitive) subtasks from the high-level surveillance tasks. Because we used different trajectory planning methods for CTOLs and VTOLs we had to distinguish between the planners. Obviously, we had to estimate two safeness functions, one for CTOLs and one for VTOLs. The experiments were performed on Core2Duo 1.86 GHz, 2 GB RAM, Windows 7. The results we obtained are depicted in Fig. 4.

The results show, that Accelerated A* algorithm has a very good performance in environments with relatively low number of obstacles. Additionally, the algorithms is highly optimized for the setting. It provides an estimation of the safeness function that reaches more than 0.95 for the unchangeable offset of 4ms and the maximal value 1.0 for the unchangeable offset of 229ms. It gives us a promising outlook for setting the value of unchangeable offset to 0.5s which is, except the safeness reasons, quite reasonable for realistic CTOLs' behavior. Of course, we must be aware of imperfectness

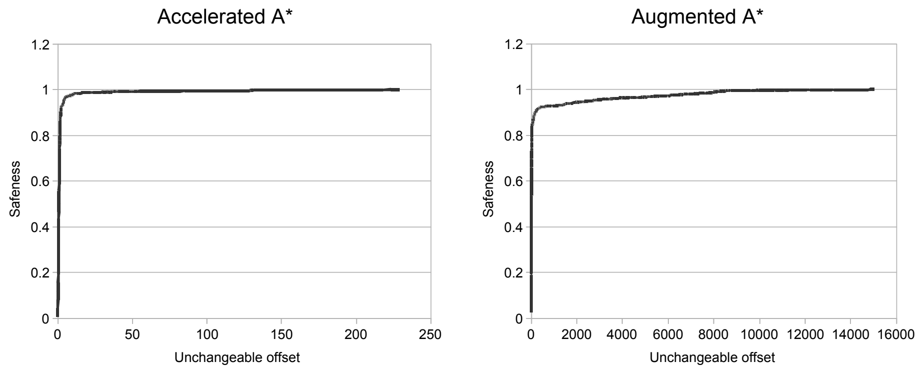

Fig. 4. Experimental results showing estimated courses of safeness functions (CTOLs - left hand side, VTOLs - right hand side)

of estimation of the safeness function, i.e., even though its value is at some point 1.0, it does not guarantee safeness in all possible situations.

Augmented A* handles the speed limit constraints and thus provides higher expressivity than the Accelerated A*. On the other hand, its performance of is worse than that of the Accelerated A* algorithm. Besides its higher expressivity, it is also caused by the implementation which is not optimized for performance. It provides us an estimation of the safeness function that reaches more than 0.90 for the unchangeable offset of 191ms, 0.95 for the unchangeable offset of more than 2.7s and the maximal value 1.0 for the unchangeable offset of more than 15s. Unlike the previous case, here the reasonable value of the unchangeable offset seems to be at least 5s (the estimation of safeness function for this value is about 0.97). However, the higher unchangeable offset is not reasonable for realistic UAVs' (VTOLs') behavior. Additionally, the risk of getting into unsafe (inconsistent) state could not be overlooked.

Handling replanning issues only by setting unchangeable offset might be enough for multi-agent simulations where we do not risk potential losses caused by asset damages. On the other hand if we consider the system working on real hardware, the potential losses might be quite high and unacceptable. It means that we have to ensure (among others) the safeness of replanning. For now, it remains an open problem but we believe that appropriate combination of using unchangeable offsets and the approaches discussed in the last paragraph of Section 4 (e.g., stopping the UAV or performing waiting loops, providing at least partial paths) could work well.

6 Conclusion

In this paper we addressed the replanning issues in trajectory planning that arise in multi-agent systems where the agents are autonomous UAVs jointly performing tasks in a pre-defined area. In particular, we considered surveillance tasks over an urban area performed by a fleet of CTOLs and VTOLs. The arising replanning issues refer to the problem of a smooth transition from an old plan (currently executed) to a new plan (provided by a planner) with respect to duration of planning. In result, the new plan has to be computed before the UAV reaches a so called 'swap point'. We defined an unchangeable offset as a duration between the replanning request and the time of reaching the 'swap point'. We also defined a safeness function assigning unchangeable offsets values expressing the probability of finishing of a replanning task within the time defined by the corresponding unchangeable offset. In our experimental evaluation, we estimated the course of safeness functions for two different planning methods, Accelerated A* and Augmented A*. We found out that in the case of Augmented A* the unchangeable offset has to be either set high, or the risk of getting into an unsafe states becomes too high. Clearly, estimation of the safeness function cannot guarantee the exact course of the safeness function, i.e., the replanning task may not finished in time even thought the value of estimated safeness function suggests it. Finally, we sketched other possible approaches how to deal with replanning issues. For the future investigation it will be necessary to take these approaches into account to provide a (combined) approach emphasizing the safeness and realistic behavior of UAVs. We must also take into account possibilities that some planning task is unsolvable, i.e., the trajectory cannot be found. Because the replanning issues also may affect successful fulfilling of the goals allocated to corresponding UAVs, these issues must be reflected in the more abstract, task allocation layers of the whole multi-agent architecture of the application.

Acknowledgements This work was supported by Czech Ministry of Education, Youth and Sports under Grant MSM6840770038 and by U.S. Army Grant W911NF-81-1-0521 and W15P7T-05-R-P209 under Contract BAA8020902.A.

References

1. L. Chrpa. Trajectory planning on grids: Considering speed limit constraints. In *Proceedings of SCAI*. IOS press, 2011. to appear.
2. L. Chrpa and A. Komenda. Smoothed hex-grid trajectory planning using helicopter dynamics. In *Proceedings of ICAART*, volume 1, pages 629–632, 2011.
3. C. Dinont, P. Mathieu, E. Druon, and P. Taillibert. Artifacts for time-aware agents. In *Proceedings of AAMAS*, pages 593–600, 2006.
4. P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
5. M. Jakob, E. Semsch, D. Pavlíček, and M. Pěchouček. Occlusion-aware multi-uav surveillance of multiple urban areas. In *6th Workshop on Agents in Traffic and Transportation (ATT 2010)*, pages 59–66.
6. N. Nigam and I. Kroo. Persistent surveillance using multiple unmanned air vehicles. In *Proceedings of the IEEE Aerospace Conference*, pages 1–14, 2008.
7. D. Šišlák, P. Volf, and M. Pěchouček. Accelerated a* path planning. In *Proceedings of AAMAS (2)*, pages 1133–1134, 2009.