CEAS Aeronaut J (2017) 8:481–492
DOI 10.1007/s13272-017-0252-5

ORIGINAL PAPER

Automated trajectory generation and airport selection for an emergency landing procedure of a CS23 aircraft

Arno Fallast[1] · **Bernd Messnarz**[1]

Received: 26 November 2015 / Revised: 14 August 2016 / Accepted: 21 June 2017 / Published online: 30 June 2017

Abstract Aircraft in general aviation usually are operated in single pilot mode. Especially, in case of an incapability of the pilot to control the aircraft, an automated emergency procedure is desirable in order to reduce the risk of fatalities. The finding of a solution for an emergency landing maneuver includes preselecting possible landing sites with regard to the available aircraft capabilities and creation of feasible trajectories to these sites. A search tree in four-dimensional search space with an efficient implementation of a rapidly exploring random tree algorithm (RRT*) is created. The algorithm performance is increased by use of basic geometrical sets to construct the final route as a combination of Dubins path segments. To further reduce the route length, a gradient based local optimization routine is added after completion of the RRT* algorithm. At the moment of creation, terrain avoidance is verified and accordance with legal airspace structure is considered. An emergency procedure is created by combining a selected landing site and a flyable trajectory to this site. Each of these combinations is scored, and the most promising emergency landing procedure is chosen and delivered to flight management system of the aircraft. The flight management system controls a full-authority auto-flight system that is capable of performing en-route flight and auto-land procedures as well.

This paper is based on a presentation at the German Aerospace Congress, September 16–18, 2014, Augsburg, Germany.

✉ Arno Fallast
arno.fallast@fh-joanneum.at

Bernd Messnarz
bernd.messnarz@fh-joanneum.at

[1] FH JOANNEUM University of Applied Sciences, Graz, Austria

Keywords Automated trajectory generation · Landing site scoring · RRT* algorithm · Dubins path · Emergency procedures

1 Introduction

Within the project eSAFE—**e**mergency **Safe** Return for CS23 Aircraft—a unique technique for automatic flight guidance including emergency landing in case of a pilot's sudden inability of flight is developed.

The project contains the adoption of a Diamond Aircraft DA42 to Fly-By-Wire capabilities, the implementation of flight control computer as well as sensor and actuator hardware and the development of control and trajectory planning algorithms.

The adoption of the Diamond Aircraft DA42 was performed in the LUFO IV Project FLYSMART simultaneously performed at Diamond Aircraft. To enable full-authority flight and outo-land functions, the DA42 with a redundant flight computer, sensor and actuator system that is designed with a scope on eligibility for certification and thus fulfills the legal required redundancy and reliability demands.

The sensors include triple-redundant system containing inertial measurement units (IMU), Global Positioning System, radar altimeter etc. In addition to that, receivers for commonly used aviation services (ACARS, DGNSS, METAR, NOTAM) are available. By using the available sensors, the execution of the emergency procedure is based only on on-board equipment. The DA42 had two different pilot control sets included: One pilot (the test pilot) controlled the aircraft solely by means of a fly-by-wire side-stick in the cockpit. The other (the safety pilot) maintained control over the conventional, mechanically connected yokes. Several quadruplex redundant safety measures allow the pilots to disconnect the auto-flight system in case of a malfunction. The aircraft used within the scope of the actual project is certified as a an experimental aircraft [11].

The actual work deals with the procedure of the trajectory generation and landing site selection that contributes to a successful emergency landing procedure. As an definition, mechanical failures of the aircraft are not within the scope of the project. Therefore the considered aircraft's capabilities are considered according to the available flight simulation model or handbook of the Diamond Aircraft DA42.

1.1 Possible approaches to the path finding problem

Automatically finding a path from a given initial position to a desired final position generally is not a trivial task. Most of the algorithms are thoroughly investigated in the field of robotics where general formulations of motion planning are introduced. Depending on the complexity (dimensionality of the search space), enormous computational effort may be required to find a valid solution [5].

There exist several approaches to finding a solution of the motion planning task. Some of them are based on the decomposition of the search space into smaller cells where the transfer from one cell to another is a relatively simple task to solve but with the drawback that for larger search spaces (size or dimensionality) the extensive number of cells required may exceed computational limitations by far [6].

Other approaches are based on potential fields, visibility graphs or roadmaps [12], eventually requiring costly precomputing effort. Using precomputed information for navigational purposes includes the drawback of inflexibility for reacting on changes in the search space. Such external influences that may require an update of the navigation solution are e.g. weather, moving obstacles, congestion due to traffic or temporary runway obstruction, just to name a few possibilities.

One field of research in the area of motion planning is focused on the methods of sampling based strategies to explore the search space. When considering computational effort or memory requirements, sampling based approaches do not necessarily scale with the size of the search area as opposed to strategies based on e.g. decomposition with fixed cell size. Given this fact, upper bounds for the computing time are derivable. Since the actual size of the search area is not know beforehand, it is still a requirement to know how the execution time of the implemented algorithm scales with the size of the investigated search area. The execution time of the chosen sample based approach scales with the number of iterations and the dimension of the searched state-space—both numbers known at the moment of design implementation of the algorithm. Therefore a sampling based approach is chosen for the current application and the implementation is described in the following sections.

1.2 Selected approach

The intention of the algorithm is to find a trajectory to a landing site based solely on the available information about the endurance, the terrain and airspace structure to a landing site provided in a database. No costly precomputational effort shall be required for application of the algorithm to ensure the highest flexibility on time dependent external influences.

The chosen algorithm evidently is capable of finding the true optimal solution—if a such a solution exists—given an infinite amount of time [3]. For a practical usage, the actual implementation is designed to search for a solution for a given amount of time or predefined number of iterations respectively. Weakening the necessity of theoretically achievable optimal route, The implementation has to gratify the availability of a *good* solution as the best-effort result that can possibly be achieved within the time and computing effort given. By design of the algorithm, the route returned at any given iteration is the best route found so far.

Furthermore, the created solution shall be delivered in a list of waypoints with a moderately low number of segments. Since the flight plan segments are used in the process of creating the search tree, the result of the route generation can directly be delivered as a list of waypoints to the flight computer.

2 Methods

The creation of a valid solution for the emergency landing in the proposed implementation includes three modules that are explained in the following section.

One module contains the **route generation** from the initial point to the intended landing sites including the creation of a search tree composed of feasible path segments. The second module is responsible for the assessment of the available **landing sites**. Finally, the route search tree is connected to the possible candidate runways and the **scores are combined**.

2.1 Route generation

The route generation is based on an adopted version of a rapid exploring random tree (RRT). The RRT algorithm creates a search tree within the sampling space, starting from the initial position and trying to connect the list of intended terminal positions with this search tree.

A modified version, the RRT* algorithm iteratively reconnects this search tree during the creation process, providing an asymptotical optimum behavior [3].

2.1.1 Algorithm

The RRT* algorithm is a sampling based algorithm capable of handling problems in multidimensional search spaces [7]. The algorithm creates a search tree *T*, containing a set of nodes *N* and the connecting paths edges set *E*.

Each node *N* is defined by its coordinates for the position within the search space. The nodes contain additional information to define the structure of *T*: Information of the parent Node N_{parent}, the unique node that ties the node to the tree. Each node contains a list $[N_{child}]$ of succeeding child nodes attached. That list may be empty or have multiple entries as well. The overall effort to reach the specific node from the starting point is stored in $N_{cost-to-node}$. An edge *E* contains the from- and to-node information, a list of intermediate coordinates for the state transition and the cost assigned to this edge.

At initialization, *T* contains only the starting point $N_{initial}$. Further nodes are then added to the tree by randomized creation of points within the search state space and connecting them in an appropriate way as described below.

The structure is given in the **RRT*** algorithm 1 [7]:

Algorithm 1: The adopted RRT* algorithm with an added procedure for the runway connection

```
$T \leftarrow \{N_{initial}\}$
$R_{Candidates} \leftarrow$ SelectCandidates($endurance$)
for $i = 1, \ldots, maxIteration$ do
    $n_{new} \leftarrow$ Sample($i$)
    $n_{nearest} \leftarrow$ Nearest($T, n_{new}$)
    if Connectable ($n_{New}, n_{nearest}$) then
        $T \leftarrow T \cup \{n_{new}\}$
    foreach $n_{near} \in N_{k-nearest}$ do           /* Rewiring */

        if Connectable($n_{near}, n_{new}$) $\wedge$ Cost($n_{new}$) +
        Cost($n_{near}, n_{new}$) < Cost($n_{near}$)) then
            Parent($n_{near}$) $\leftarrow n_{new}$   /* Update parent
            node */
            Cost($n_{near}$) $\leftarrow$ Cost($n_{new}$) + Cost($n_{near}, n_{new}$)
    foreach $r \in R_{Candidates}$ do
        ConnectRunway($r, T$)
```

The **initial position** is basically given by the actual aircraft state at the moment $t_{activate}$ of the activation of the emergency system, or to compensate for calculation time of the algorithm, a state in the *near* future $t_{activate} + t_{calculation}$, of the aircraft assuming e.g. straight flight, where $t_{calculation} << t_{endurance}$. So the execution time of the algorithm to deliver a valid emergency maneuver solution may be neglected in relation to the total flight time for the emergency maneuver.

To reduce following computational effort, a preliminary **selection of candidate** runways is extracted from the navigational database giving the set $R_{candidates}$. The preliminary chosen candidate runways all have a linear distance within the achievable maximum flight range with the remaining fuel on board at the moment of activation. At this moment, it is not yet clear if a route to each candidate can be found. Each candidate runway is defined by the coordinates of its threshold $[x_{thr}, y_{thr}, alt_{thr}]$ and its heading ϕ.

The maximum endurance is expressed in terms of flight distance. For the case of no wind, this distance defines a circular region centered at the initial point. That is used for the process of the randomized sampling. The process of sampling new states is an issue on itself since the distribution influences the convergence behavior of the search [8]. As an initial approach, an uniform distribution within the search space is realized.

The step of **sampling** creates new nodes that will be connected to the search tree *T*. A new sampled node N_{new} is a state vector containing the coordinates x_{new} and y_{new}, the altitude alt_{new} and the heading ϕ_{new}, giving the vector

$$\mathbf{n_{new}} = \begin{pmatrix} x_{new} \\ y_{new} \\ alt_{new} \\ \phi_{new} \end{pmatrix}. \tag{1}$$

The coordinates x_{new} and y_{new} fulfill the constraint

$$\sqrt{(x_{new} - x_{initial})^2 + (y_{new} - y_{initial})^2} < dist_{endurance}, \tag{2}$$

the altitude alt_{new} is within the range of permissible values such that

$$alt_{new} = [terrain(x_{new}, y_{new}) + \Delta alt_{safe},\ alt_{max}], \tag{3}$$

and the heading ϕ_{new} lies given by

$$\phi_{new} = [0, 360). \tag{4}$$

The newly sampled point n_{new} is not yet connected to *T*, so $parent(n_{new}) = \emptyset$, $child(n_{new}) = \emptyset$ and $cost-to-node(n_{new}) = \infty$, respectively.

This new sampled node n_{new} has to be connected to the existing search tree *T*. The connection is performed not to the node that is closest to n_{new}, but rather to that node n_{parent} that satisfies the precondition of having a feasible connection $n_{parent} \rightarrow n_{new}$ and giving the lowest **overall** cost $cost_{total} = cost_{parent} + cost(n_{parent} \rightarrow n_{new})$. The definition of feasibility will be given in the following sections. The set $N_{parentCandidate}$ ideally contains all nodes of *T*, in a practical approach it may be sufficient (i.e. without loosening the behavior of asymptotically approaching the optimal value) to select a subset $N_{k-nearest} \subset N$ of the k-nearest neighbor nodes to n_{new}, where $k \leq |N|$ obeying a given distance metric [6].

The term $connectable(N_{new} \rightarrow N)$ ensures that there is a feasible connection between the pair of points that the aircraft is capable to fly. It has to be checked if a path generated by the segment creation mentioned before satisfies the following constraints:

- terrain avoidance including safety margin,
- airspace restrictions, e.g. avoidance of prohibited airspace, danger zones depending on their type (e.g. even in case of an emergency, areas of live artillery firing must be avoided),
- aircraft performance capabilities, i.e. minimum turn radii, maximum climb gradient, service ceiling.

Having added n_{new} to T and calculated the overall cost-to-node $cost_{total}$ for n_{new}, the **rewiring** is performed. This step is vital for the optimum behavior of the RRT*-algorithm [7]. Having the set of nodes $N_{k-nearest}$ each node is checked if the newly added node n_{new} gives the possibility of a new connection with lower overall cost-to-node. So if

$$cost(n_{new}) + cost(n_{new} \rightarrow n_{near}) < cost(n_{near}) \tag{5a}$$

and

$$connectable(n_{new} \rightarrow n_{near}) \tag{5b}$$

is true, the node $n_{near} \in N_{nearest}$ has an improved connection via the node n_{new}. The parent information $parent(n_{near})$ is updated to n_{new} and the cost-to-node $cost(n_{near})$ is set to $cost(n_{new}) + cost(n_{new} \rightarrow n_{near})$.

Thus it is ensured that an additional node n_{new} is not only connected to the node n_{parent} of the existing tree T in an optimal way ensuring lowest possible overall cost-to-node, but also the cost-to-node for the k-nearest neighbors benefit from a newly added node n_{new}. Through this active rewiring process the overall cost-to-node values within the tree asymptotically approach the true optimum. Again, for a practical approach and computational efficiency it suffices to regard the subset $N_{k-nearest}$ for the rewiring process for k being sufficiently high, although in an ideal case the rewiring is performed for the set N.

2.1.2 Segment creation

The above mentioned general connecting segment between to nodes has to obey the constraints of the aircraft performance in terms of minimum turn radius. It is proven by Dubins [2] that the connection between two points A and B, defined by their coordinates and heading, having the minimum turn radii from and to the points, r_A and r_B, is always a combination of circle segments, connected by a straight line, (denoted by the segment type S). The circles may be the shape of a left turn or a right turn (segment type L or R respectively), giving the possible combinations LSR, LSL, RSL and RSR as depicted in Fig. 1. For Dubins-Path segments there exist efficient algorithms [14] to determine which of these types is the shortest—an essential prerequisite for the overall performance of the whole trajectory generation.

The minimum turning radius is determined by the capabilities of the aircraft at a given commanded cruise speed referring to the aircraft's pilots handbook [1].

If the distance between the points is smaller than the sum of the two radii such that $|\overline{AB}| < (r_A + r_B)$ and the initial and final heading ϕ_A and ϕ_B are in an adverse relation, no straight line segment can be fitted between the circle segments [14]. Thus the resulting minimum distance path is of the type circle–circle–circle, either in the shape of left–right–left or right–left–right as given in Fig. 2 leading to connection paths of the type RLR and LRL. Due to performance issues and to maintain an generic layout of the finally created route, the rarely occurring cases of circle–circle–circle path types are not taken into consideration at all.

2.1.3 Longitudinal segment component

The previous section described the creation of the lateral components of a path connecting sampling points within the search tree. This path covers the state transition for three dimensions of the four-dimensional search space used

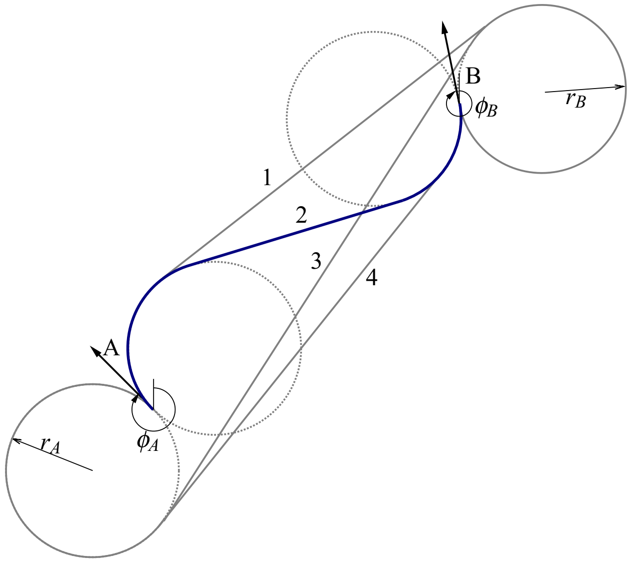


Fig. 1 The four possible circle-line-circle connections $\overline{AB}$: RSR (*1*), RSL (*2*, the shortest in this case), LSL (*3*), LSR (*4*)

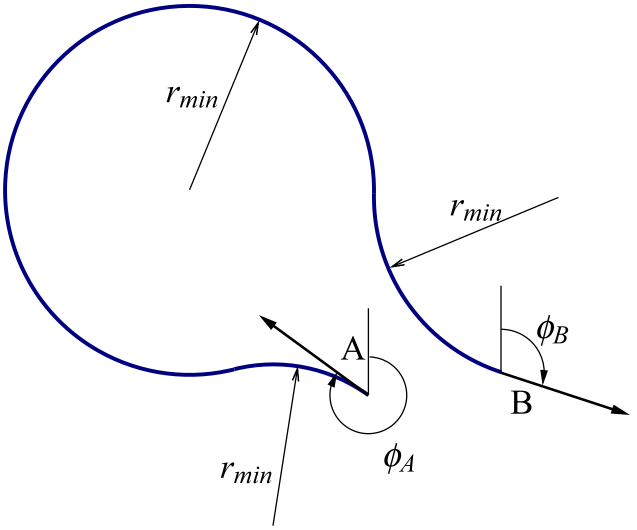


Fig. 2 A possible circle–circle–circle (LRL) path connecting the points *A* and *B* with a minimum turn radius r_{min}

for trajectory generation: The x and y coordinates and the heading ϕ.

For the altitude transition between two points to connect, a continuous climb or descent is assumed. Therefore, a maximum achievable climb or descent gradient must be considered. For a descent maneuver the limiting factors are the maximum allowed speed n_{NE} of the aircraft at a given configuration and maximum applicable "comfortable" descent rate. As a practical approach a descent rate of 1500 ft/min is reported to be feasible by pilots.

For a climb segment, the maximum achievable rate of climb (RoC) and the corresponding distance to climb is extracted from lookup tables of the aircraft's performance charts depending on the actual weight of the aircraft. For performance calculation, the pressure altitude and outside air temperature (OAT) of the target point are considered, ensuring the intended RoC can be maintained over the whole climb segment.

The concept of the reachability is illustrated in Fig. 3.

2.2 Route scoring

In this section the scoring of the routes is further described. While the prior explained route creation process iteratively aimed at reducing the *cost* (i.e. the path length) of a route, the scoring process evaluates the created routes and assesses them in a numerical way.

It is desirable to introduce a normalized scale that represents the length of the route (cost effort) and the differences of cost-to-go to a specific target. The limiting factor for the maximum reachable distance is the given endurance. Consequently, a route exceeding the endurance has to be scored with zero in order to exclude it from further treatment. A score of 1 is defined as the line of sight distance to the nearest runway available out of the list $R_{candidates}$.Thus the scoring S_{route} for the route to a node N with the cost $C_N = cost - to - go(N)$ depends on the endurance C_{max} and distance to the nearest runway $C_{min} = distance(R_{nearest} \in R_{candidates})$ and is defined by:

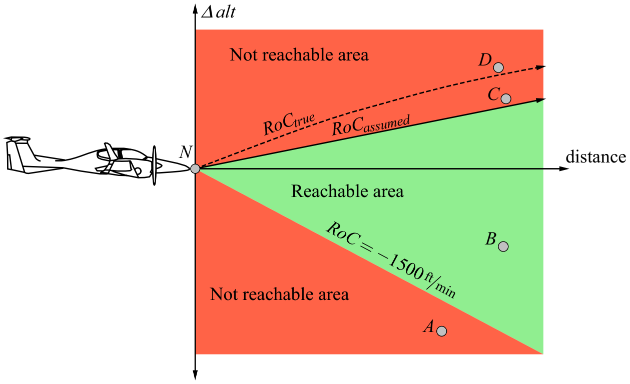


Fig. 3 The reachable areas for the connection of a the point N to the tree. Point *A* is not reachable due to velocity limitations (V_{NE}); Point *B* is reachable; Point *C* is not reachable with the simplified RoC assumption; Point *D* is not reachable

$$S_{route} = max\left(0, 1 - \frac{C_{min} - C_N}{C_{min} - C_{max}}\right), \quad S_{route} \in [0, 1] \tag{6}$$

2.3 Landing site selection

To select the most promising landing site for a safe emergency landing procedure, different factors are considered. These factors are either statically stored in a database such as runway dimensions and orientation and technical facilities on site or time dependently derived from meteorological information such as crosswind component or runway conditions. The resulting ranking is a numerical representation that aims to reproduce a real pilots decision finding [10]. A higher score thus represents an emergency solution that promises a faster medical treatment of the pilot.

Given the available maximum endurance of the aircraft, a preselection of possible landing sites is performed, giving a set $R_{candidates}$ of candidate runways for which the further assessment is performed. To give an insight of the density of available airports, an excerpt of the database for the eastern part of Austria is depicted in Fig. 4.

For the sake of generality, the evaluation process is based on runways rather than on airports. Doing so allows the selection of the optimum landing site by consequently using the same generic scoring system reaching the detail level of even selecting the most appropriate runway in case of differently orientated runways on the same airport.

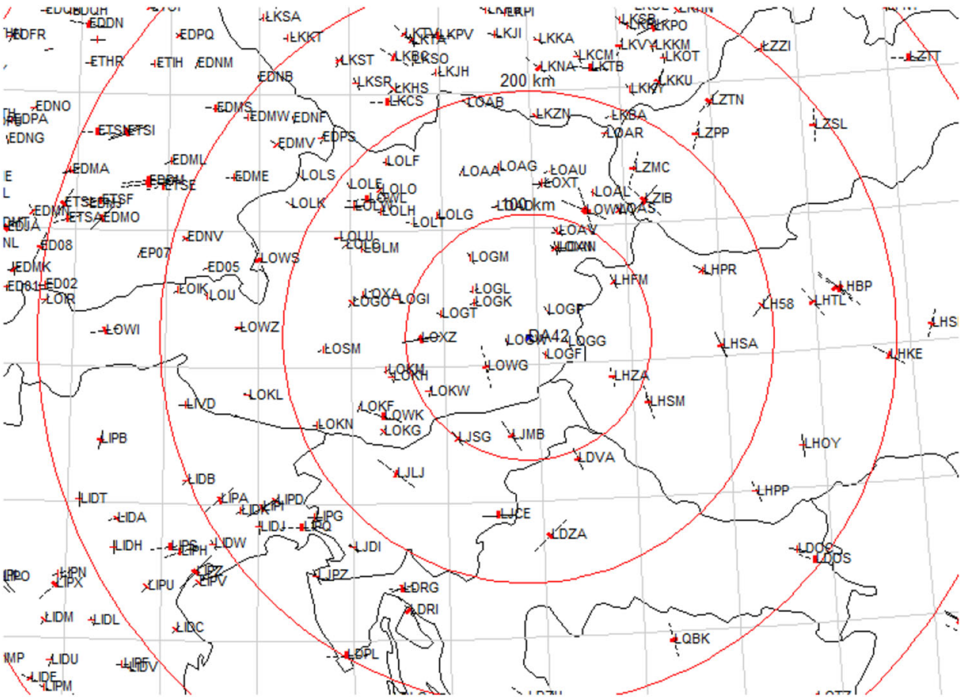


Fig. 4 An example for the database content used for the preselection of runways based on maximum endurance. The airport codes and runway orientations are depicted. The starting point is in the center of the *red circles* that represent an range of 100, 200, 300, and 400 km

Headwind and crosswind components, shifted thresholds or the slope for calculation of the landing distance are automatically taken into account for each landing direction at every available runway.

The set of $R_{candidates}$ runways is evaluated and finally scored.

2.3.1 Runway risk

To assess the risk for a landing site, the determining factors are the runway dimensions, the local weather conditions and the achievable braking action. The required runway length can be computed by use of look up tables contained in the aircraft's performance handbooks by considering weight information from the flight computer. The weight has to be updated by the estimated en-route fuel consumption while executing the flight path calculated by the trajectory planner module.

The risk of the runway length is thus expressed as following:

$$P_{length} = min\left(1, \frac{length_{available}}{length_{required}}\right), \quad P_{length} \in [0, 1]\,. \tag{7}$$

By using the performance charts of the handbook to calculate the required runway length, all conditional factors such as aircraft weight, obstacle height, temperature, pressure altitude, braking action due to surface condition and runway slope are considered. An additional risk factor P_{x-wind} for the crosswind is scaled with regard to the maximum demonstrated crosswind of the specific aircraft type:

$$P_{x-wind} = max\left(0, 1 - \frac{v_{wind}}{v_{x-wind,permissible}}\right), \quad P_{x-wind} \in [0, 1] \tag{8}$$

Here $P_{x-wind} = 0$ means the maximum demonstrated crosswind is exceeded and $P_{x-wind} = 1$ means no negative influence on the safety due to wind. In fact the consideration of local weather assumes access to such data via data link which may become available in the near feature [13].

A similar factor is added for the runway width:

$$P_{width} = min\left(1, \frac{Width_{available}}{width_{minimumpermissible}}\right), \quad P_{width} \in [0, 1] \tag{9}$$

2.3.2 On site facilities

Additional important factors that influence the scoring of the landing site are the on site facilities such as availability of fire brigade or technical emergency services on site, the local quality of medical services (distance to closest emergency care unit) or other arbitrarily chosen scoring preferences. Therefore a scoring factor $S_{facilities}$ is introduced. The values are added to the database manually and chosen to satisfy

$$S_{facilities} \in [0, 1] \tag{10}$$

2.3.3 Meteorological influences

Meteorological circumstances actually influence save landing in multiple ways. If information about the local weather situation at the candidate runways is available, they can be taken into consideration in the previous mentioned calculations for the risk emanating of the runway.

The most influencing factor for sure is the wind direction and speed which is taken into consideration by previously mentioned factors. Severe influence on the braking action after touchdown may be given by runway contamination due to ice, snow or a wet runway.

Additional weather hazards may occur en route. A prevailing wind field may influence the trajectory generation or the segment creation itself and hence these factors are not considered in this section. The other aspect is the influence of temporary thunderstorm areas or contaminated zones due to sand storms or ash clouds. These areas can be treated as *weak obstacles* i.e. a well equipped aircraft may fly through these zones to a certain extent. In case of an emergency, a certain risk may be accepted. A further insight into such strategies is given in [4].

2.3.4 Summarized scoring for the runway

The score S_{runway} for a runway is given by the product of the single risks factors and the additional facilities factor:

$$S_{runway} = S_{facilities} \cdot P_{length} \cdot P_{x-wind} \cdot P_{width} \tag{11a}$$

$$\text{with} \quad S_{runway} \in [0, 1] \tag{11b}$$

2.4 Decision finding process

The final score for the emergency landing maneuver as a whole is consequently given by a weighted combination of the score for the landing site and the score resulting form the en route cost of the found trajectory:

$$S_{total} = k_{runway} \cdot S_{runway} + k_{enroute} \cdot S_{enroute} \tag{12}$$

Weighing the scores for route and runway allows for tuning the system behavior to prefer either an immediate landing at the absolutely closest possible airport available—even with the risk of a hazardous touchdown—or to tend towards a rather safe landing with the drawback of higher en route travel time.

The weighing factors are implemented in the software module that performs the scoring of the runways. Since for the actual implementation of an emergency system it is assumed that the pilot is incapable to decide, this process is performed by the eSAFE system.

The combination of runway and trajectory with the highest overall score is then chosen as the final emergency maneuver solution as given in Table 1. The result is the ranked score for the single runways. If a specific single risk factor is below a certain threshold, additional information is listed. This solution is then delivered to the flight computer and communicated to Air Traffic Control (ATC) in an automated manner. In case of triggering the eSAFE system, it is required that the transponder is set to an according emergency code. The communication of the intended flight route may consist of the transmission of automatically generated voice messages (text-to-speech) or by means of digital data link possibly available in the future [13].

3 Results

To prove the feasibility of the implemented algorithm a series of several thousand simulated algorithm calls where performed. A first scope of investigation was to prove the stability of the program and to gather data for an adaption of the search parameters. These simulation runs where conducted with a fixed starting point and variation of search algorithm parameters such as number of iterations and number of neighbors to consider for the connection and rewiring process mentioned in the algorithm process. A second set of simulation runs where performed with randomly chosen starting points. The starting position as well as the altitude and heading where randomly chosen within a predefined search area, while avoiding states where no solution exists. As an example, such a case without a solution may occur when starting the simulation with the aircraft right in front of a cliff. The simulation setup was chosen to ensure that there exists a solution with regard of the predefined endurance and the available airport in the proximity. The terrain contained the mountainous area of the Austrian alps, thus the navigational solution to the desired solution usually was not trivial. An example of a generated trajectory is given in Fig. 5, the corresponding elevation profile along the route is given in Fig. 6.

Table 1 A sample result score list. Additional risks are enumerated in the list

Runway	Distance (km)	Score (−)	Risks
LOGF 13	20	0.821	
LOGF 31	21	0.780	RWY
LOGP 33	26	0.586	RWY, WET
LOGG 15	32	0.581	
LOGG 33	32	0.573	
LOGP 15	26	0.544	RWY, WET
LOGW 18	18	0.525	RWY
LOGW 36	18	0.474	RWY
LOWG 17C	41	0.463	
LOWG 17L	42	0.454	

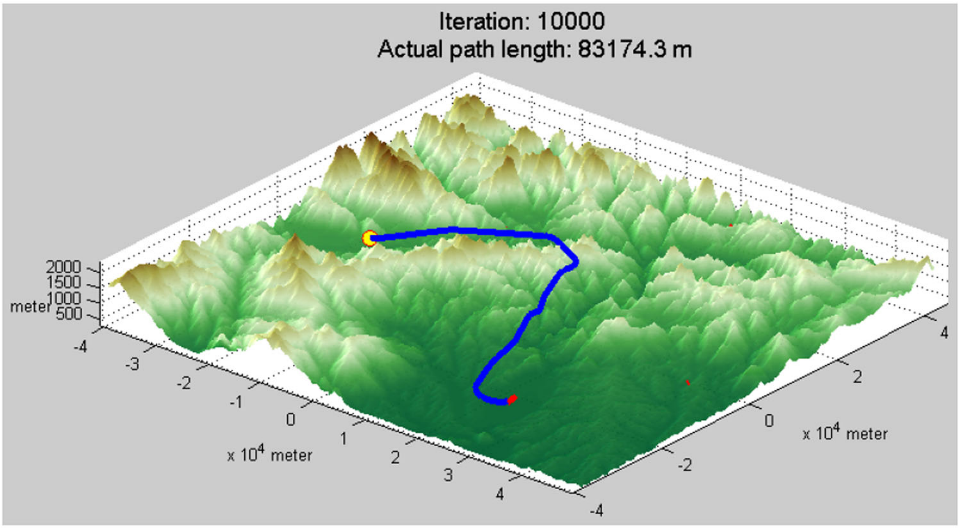


Fig. 5 An example of a created trajectory (*blue line*) by the algorithm in the terrain near Graz, the starting point is the *yellow circle*, the chosen runway is **LOWG 35C** (*red line*)

The conducted simulations of the algorithm have shown the requested behavior, whose main characteristics can be summarized by the following features:

- success rate,
- performance of solution,
- availability of solution,
- no precomputational effort,
- moderate length of waypoint list.

The success rate Within the performed simulation calls only few cases where recorded where no solution was found. Such were the case when an arbitrarily chosen starting point was too close to the terrain for a feasible evade trajectory—a circumstance the algorithm can not be accounted for.

The performance of the algorithm The evolution of the length for the emergency maneuver enroute part shows the expected behavior concerning the approach to an (actually unknown) true optimum. An example for the development of the length of the found emergency maneuver route (Fig. 7) is given in Fig. 8.

Availability of solution As depicted in Fig. 8, it is evident that a first solution was provided by the algorithm after only a few iteration steps and rapid improvements path length could be achieved within the subsequent iterations. Only minor improvements could be achieved later on, even after a high number of iterations where performed. This also implies, that the required *good* solution is available quickly, although the true optimum is only reached with infinite iterations.

No precomputational effort was required to perform the simulations. The path generation algorithm as proposed only relies on the—statically stored—database of the terrain and the available runways. This ensures high flexibility for future implementation regarding temporal influences such as bad weather zones or avoidance of high traffic areas.

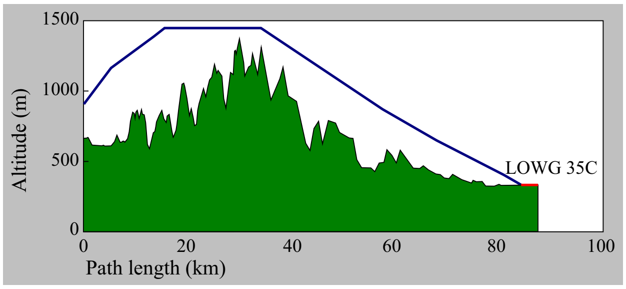


Fig. 6 The elevation profile for the solution in Fig. 5

The resulting waypoint lists created by the algorithm calls barely consisted of more than a few waypoints – an important key in efficiently communicating the intended emergency route to the flight control computer of the aircraft and to ATC.

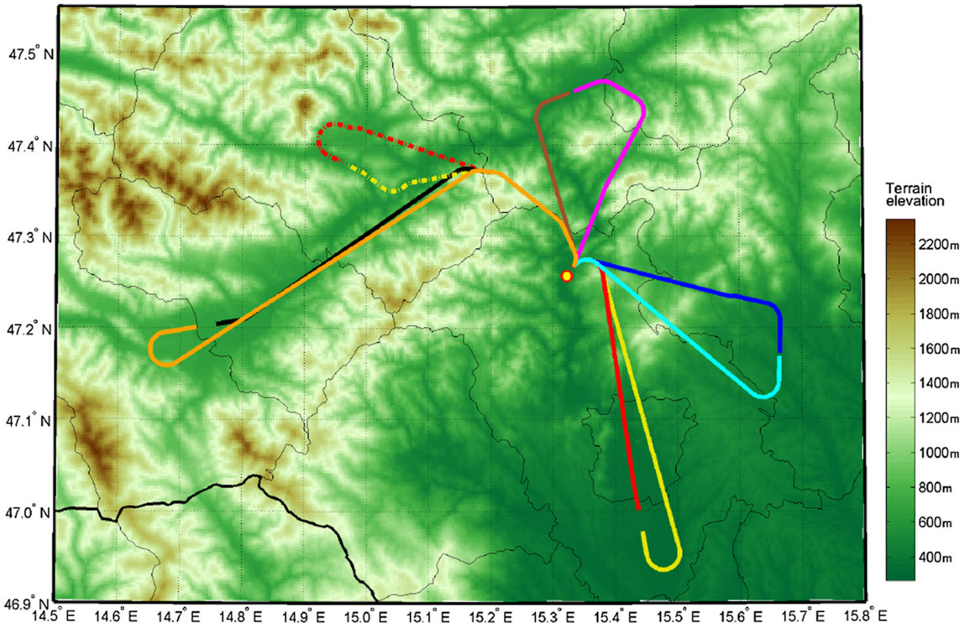
4 Flight Tests

First tests of the eSAFE software where conducted at the FH JOANNEM research simulator (**JFS**^{2}) depicted in Fig. 9. The trajectory generation module was connected to the JFS2 simulator. For the flight dynamics, the simulation model of the DA42 as used within the certified FNPT-II simulator was implemented.

The final version of the eSAFE software finally shall be tested in real flight conditions in the area of Wiener Neustadt Ost (ICAO-Code: LOAN), home base of *Diamond Aircraft Industries GesmbH*.

Prior to the flight tests, the software was tesed at the ironbird of the DA42 at the University of Stuttgart. There a complete copy of the hardware (including flight computers,

Fig. 7 An example route set created with 10,000 iterations. The maximum flight altitudewas set to 1500 m so the trajectory has to circumfly the summit at N 47.2° E 15.1° in the north. The development of the route length is given in Fig. 8



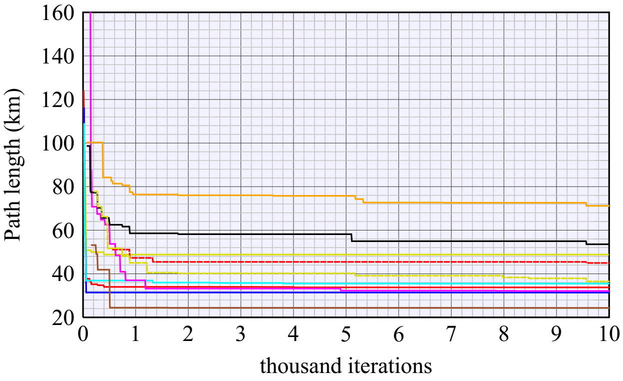


Fig. 8 The development of the path length plotted over the number of iterations for several candidate runways. The cost-to-goal shows a rapid improvement over the first few iterations and minor subsequent improvements while asymptotically approaching the (actually unknown) true optimum. The colors of the paths are set according to Fig. 7

actuators and sensors) is situated in an hardware-in-the-loop setup and the data communication between the trajectory generation module and the DA42 flight computer could be tested. After successfully testing the collaboration of the systems, final flight tests could be conducted on the real DA42 in flight. There the computer hosting the trajectory generation is connected to the flight computer over a cabled data link provided by the aircraft. Since the software for the trajectory generation for the flight testing was not a safety critical part for this prototype—in fact, the pilot always could manually override the system—the execution could be run at a commercially available laptop computer without any special concerns on safety relevant software.

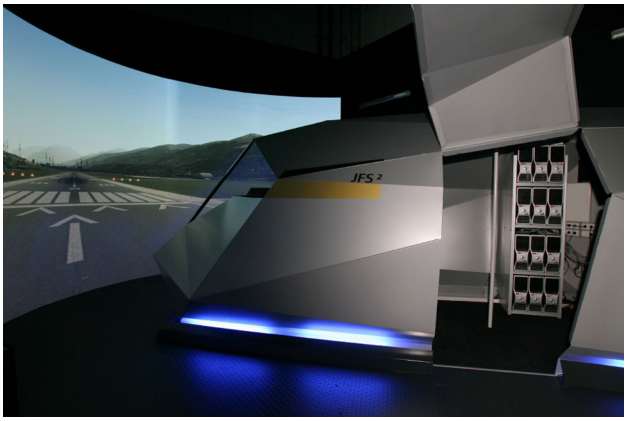


Fig. 9 The FH JOANNEUM research simulator (**J**oanneum **F**orschungs-**S**imuator—JFS^2)

Due to legal aspects the permit to fly for the experimental aircraft and the conducted flight tests included several restrictions:

- Auto-flight tests must be performed within a restricted area in the surroundings of LOAN.
- Flying over the urban area of Wiener Neustadt and nearby villages is prohibited in auto-flight.
- Approaches are allowed only for the runway **LOAN27**.
- No automated activation of auto-flight mode. An additional manual confirmation of the pilot has to be implemented.
- No direct execution of auto-land. At least one turn in a designated holding pattern has to be performed. Additional confirmation of the final approach and the auto-land has to be executed by the pilot on board.

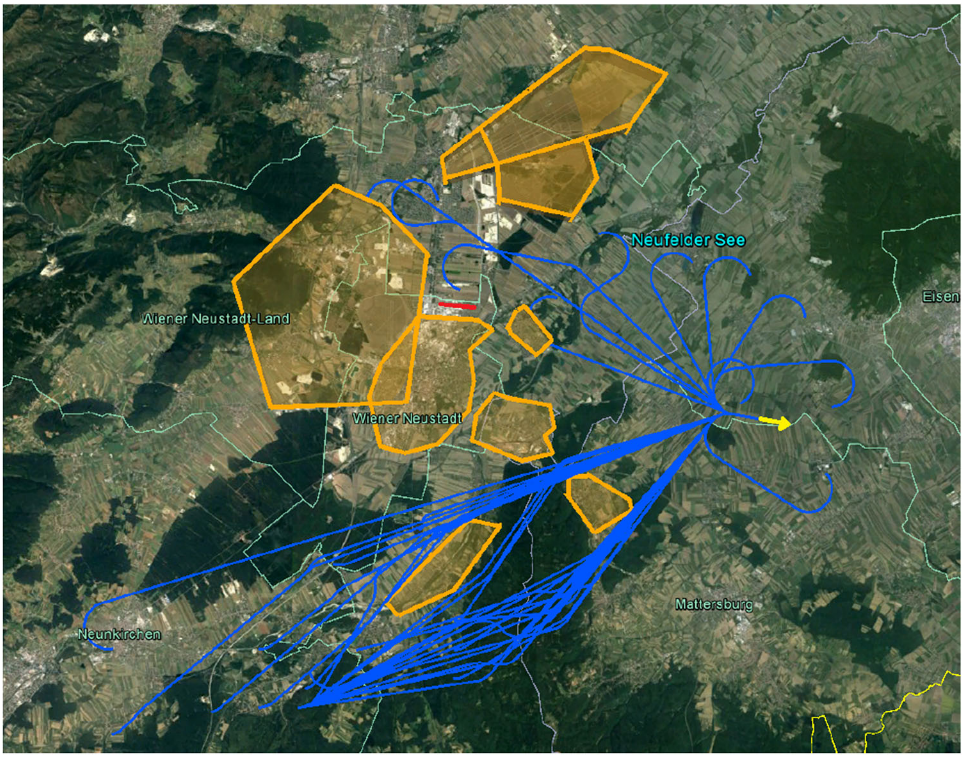


Fig. 10 The trajectories created during the test flight test mission (*blue lines*). The entry to the holding pattern is marked with a *yellow arrow*. Also depicted are the runway LOAN27 (*red*) and the no-fly zones (*orange polygons*)

While the first restriction reduced the possibility to test the algorithm in real flight under the influence of complex or mountainous terrain (In fact the vicinity of Wiener Neustadt does not feature any major elevations), the second restriction in turn added several no-fly zones that have to be obeyed.

The third restriction prevented the final runway scoring algorithm to be tested under real conditions. Since this is a numerical assessment of created trajectories and runway parameters, the results of this module are not influenced by any flight dynamic aspects (speeds, weight, turn radii). Thus not testing the runway scoring module only represented a minor flaw of the flight test campaign.

The fourth restriction may influence the performance of the trajectory following. An additional visual inspection of the created route by the pilot and a subsequent confirmation is a time consuming process. Thus it may occur that starting point of the created route is situated behind the actual aircraft's position the flight path controller of the flight computer has to establish on the trajectory.

The last restriction does not influence the function of the trajectory generation algorithm itself at all. Since the algorithm is implemented in a general formulation, the database actually consisted not the data for the runway **LOAN27** but the data to the inbound leg of the holding pattern. Thus it presented the capability to obey given approach procedures at any arbitrarely runway. Since the data for the runway consists of a point in space represented by latitude, longitude, elevation and heading, the same structure can also be used for a *virtual runway*—the final leg that leads into the predefined holding pattern or a final approach fix where an approach procedure starts. Compared to an predefined approach procedure, the holding pattern and the final approach are hard coded into the database of the runway.

While the aricraft was alread in flight and in preparation for the actual triggering of the eSAFE-system, the trajectory generation module was already delivering emergency trajectory solutions. The bunch of trajectories is depicted in Fig. 10. It is obvious, that all trajectories returned by the trajectory generation system avoid the no fly zones and lead to the entry point of the holding pattern. The starting point of the trajectories varies according the actual aircraft position. Owed to the fact that intense debugging information was recorded during this test flights, the route creation interval is tremendously reduced at this test flight. Also a visual output of the claculation progress was given to the pilot. Hence the usaually achievable update rate of five seconds was artificially reduced.

Only a very limited time slot was available for the conduction of the flight tests, hence only one approach could be performed. The flight path of the actually flown trajectory is depicted in Fig. 11.

As it can be seen, the manual activation of the auto-flight system has been delayed and the calculated starting point has been missed. This is owed to the additional visual inspection of the generated trajectory and subsequent activation of the auto-flight mode. However valuable data could be collected and the system integration has been proven to collaborate well.

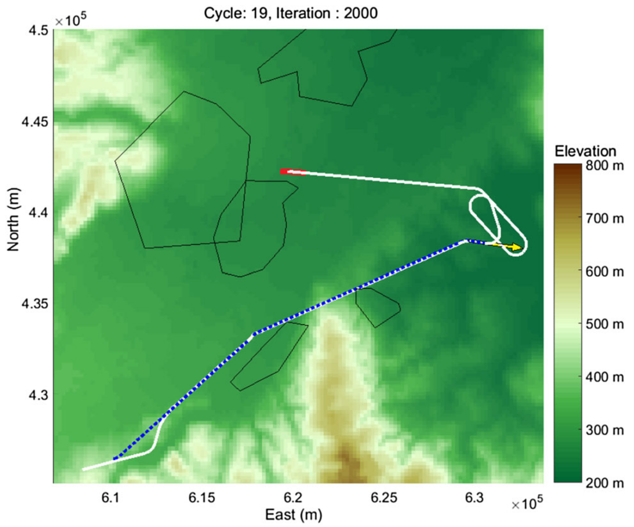


Fig. 11 The recorded flight path of the flight test (*white line*) and the created flight trajectory (*blue dashed line*). The entry to the holding pattern is marked with a *yellow arrow*. Also depicted are the runway LOAN27 (*red*) and the no-fly zones (*black polygons*)

5 Conclusion

The **RRT*** algorithm that is chosen for the actual implementation is proven to yield an optimal solution if such an solution exists [3]. Even with the use of simplified models for the flight performance of the aircraft the simulations have shown to still deliver a feasible solution for an emergency landing route that is expected to be *close* to the actually unknown true optimum with respect to the defined cost function (the path length in the current implementation). In the context of the trajectory generation in case of an emergency, immediate action is required, since the aircraft inevitably has to proceed its flight which may be in an disadvantageous direction. Therefore it is required to accept the best route available that could be found within the time given. The convergence behaviour towards the optimum has been investigated for trivial cases (such as flying meandering S-shaped trajectories through artificial terrain) where the true optimum could be determined analytically. These testcases confirmed the property of the **RRT***—as stated in literature–to assymptotically approach a true optimum, also in the current implemetation with Dubins-path segments and the preformance of the DA42 considered.

The representation of the aircraft's performance limitation by simplified geometrical considerations such as minimum turn radii and climb rates yields the desired flyable trajectory with a length that is expected to be close to the optimum and provides this solution within a reasonable amount of time. In fact, an update interval of 10 seconds could be demonstrated during the simulations.

The shown method of scoring the landing site candidates represent a comprehensible representation of the factors a real pilot would take into consideration. The scoring delivers a clearly stated preference of the available landing sites.

Combining the scores for the en-route part and the landing site itself ensures the execution of the emergency landing procedure promising the least risk for further fatalities on board or on ground.

6 Outlook

The implementation of the proposed algorithm into the existing Fly-By-Wire platform and the validation by flight test with the Diamond DA42 are actual work in progress.

The first conducted real-flight tests have shown that the implementation of the algorithm interacts as expected with the flight management system and the created emergency routes can be followed by the autopilot in a perfect manner.

Meteorological influences are taken into consideration for the critical phase of landing where headwind or tailwind directly influence the landing capabilities of the aircraft. For the flight planning itself, a detailed consideration of a wind field is not considered yet, hence the stated observations hold for conditions with weak wind. While there exist possibilities to account for a prevailing wind field at the process of path planning [9], these calculations add further computational effort. Here the tradeoff between the additional gain in the validity a found trajectory and the availability within a short amount has to be thoroughly investigated in the future.

Even though the flight tests have shown promising results, it has to be mentioned that—due to legal restrictions—these tests were performed in a limited area and a single runway was given as a landing option. Thus the full functionality of the runway scoring procedure could not be tested in real flight environment. Although the simulation studies have shown desirable results and the scoring procedure seems to be capable of performing well, further flight testing and tuning of the scoring parameters is inevitable.

Future investigations also have to deal with the concept of how the system is activated in flight. Obviously the system for the prototype was activated manually. After the Creation of an emergency maneuver solution, the pilot had to additionally clear the execution of the auto flight phase due to legal constraints in the prototype. The authorities requested an additional check by the pilot in command for liability reasons. In a future implementation different activation possibilities are thinkable: A first possible activation could be realized by an emergency button within the cockpit. Such a button could easily be pressed even by a handicapped pilot or by a passenger instructed to do so.

An other possible approach may be an implementation of a dead-man's vigilance device. Since the trajectory generation algorithm is permanently activated in the background, the flight computer can be aware of an upcoming dangerous situation. Such a situation may be the risk of running out of fuel without the possibility to return to the next airport when flying over sea at an search and rescue mission or approaching the vicinity of mountains in a narrow valley. Also a sudden steep descent—maybe as a result of a passed out pilot—or no input at all over a certain period may possibly activate the system. At all times, it shall be possible for a pilot to regain manual control in case a incapability was temporary caused by e.g. oxygen deficiency in high altitude.

A second future implementation may target the task of flight planning without an emergency situation solely for convenience. The algorithm in its general form as implemented within the project can also be used as a planning tool for an regular flight. By restricting the candidate runways to those on a desired destination airport, the algorithm plans a feasible and comfortable route with consideration of the terrain and airspace structure. Obviously, in such an implementation, the list of waypoints is given to the pilot to manually fly the route as desired without declaring emergency. This activation may be started manually at any time—be it prior to takeoff or in flight when optionally diverging to a different airport.

While the above-mentioned implementation for an automated flight execution including auto-land requires a tight coupling to the aircraft's flight computer and rigorous treatment of certification aspects, an intermediate implementation may be realized as an assistance system—only displaying a possible route to the pilot that has to manually fly the route. Thus the system has not to be integrated into the aircraft and may be available as a standalone solution for a wider range of aircraft.

Acknowledgements Open access funding provided by FH Joanneum—University of Applied Sciences. This work represents results obtained within the project eSAFE, sponsored by the Austrian Aeronautics Research and Technology Programme (TAKE OFF) of the Austrian Research Promotion Agency (FFG)

References

1. Diamond Aircraft Industries GMBH: Airplane flight Manual DA42 NG. Diamond Aircraft Industries GMBH, Wiener Neustadt, Austria (2014). Rev. 6 01-Apr-2014 Doc. # 7.01.15-E
2. Dubins, L.E.: On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. Am. J. Math. **79**, 497–516 (1957)
3. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. Int. J. Robot. Res. **30**(7), 846–894 (2011)
4. Kuenz, A.: The 5th dimension in conflict management—xyzt+-capability. In: 2015 IEEE/AIAA 34th digital avionics systems conference (DASC), pp. 1A5–1–1A5–9 (2015). doi:10.1109/DASC.2015.7311331
5. LaValle, S.M.: Planning algorithms. Cambridge University Press, New York (2006)
6. LaValle, S.M., Branicky, M.S., Lindemann, S.R.: On the relationship between classical grid search and probabilistic roadmaps (2004). http://msl.cs.uiuc.edu/~lavalle/papers/LavBraLin04.pdf. Accessed 15 Nov 2013
7. LaValle, S.M., Kuffner Jr., J.J.: Randomized kinodynamic planning. Int. J. Robot. Res. **20**(5), 378–400 (2001). http://dblp.uni-trier.de/db/journals/ijrr/ijrr20.html#LaValleK01. Accessed 14 May 2013
8. Lindemann, S.R., LaValle, S.M.: Current issues in sampling-based motion planning. In: P. Dario, R. Chatila (eds.) Robotics Research. The Eleventh International Symposium, *Springer Tracts in Advanced Robotics*, vol. 15, pp. 36–54. Springer Berlin Heidelberg (2005). doi:10.1007/11008941_5
9. Martin Seleck Petr Vana, M.R.T.M.: Wind Corrections in Flight Path Planning. Int. J. Adv. Robot. Syst. (2013). doi:10.5772/56455
10. Meuleau, N., Plaunt, C., Smith, D.E., Smith, T.B.: An emergency landing planner for damaged aircraft. In: K.Z. Haigh, N. Rychtyckyj (eds.) Proceedings of the Twenty-First Conference on Innovative Applications of Artificial Intelligence. AAAI (2009). http://aaai.org/ocs/index.php/IAAI/IAAI09/paper/view/268. Accessed 2 May 2013
11. Nguewo, D.: Technische und operative Anforderungen an ein emergency-safe-return-System. FFG: Internal Technical Report D1.1 (2015)
12. Schøler, F.: 3d path planning for autonomous aerial vehicles in constrained spaces. Ph.D Thesis, Section of Automation & Control, Department of Electronic Systems, Aalborg University (2012)
13. SESAR Joint Undertaking (ed.): The Roadmap for delivering High Performing Aviation in Europe—European ATM Master Plan. Publications Office of the European Union, Luxembourg (2015). Executive View
14. Shkel, A.M., Lumelsky, V.: Classification of the dubins set. Robot. Autonom. Syst. **34**(4), 179–202 (2001). http://www.sciencedirect.com/science/article/pii/S0921889000001275. Accessed 14 Nov 2014