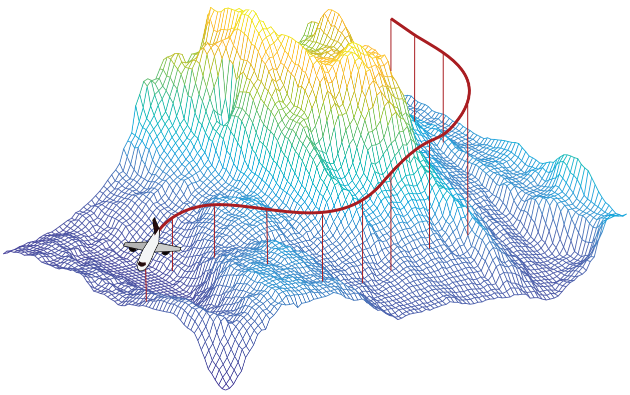
Technische Universität München

Institute of Flight System Dynamics

Controlled Flight Into Terrain Analyses In Flight Data Monitoring

Semester Thesis



Author: Niclas Bähr

Matriculation Number: 03639018

Examiner: Prof. Dr.-Ing. Florian Holzapfel

Supervisors: Lukas Höhndorf, Javensius Sembiring, Phillip Koppitz

November 8, 2017

This Thesis has been modified from its original version for publication!

Statutory Declaration

I, Niclas Bähr, declare on oath towards the Institute of Flight System Dynamics of Technische Universität München, that I have prepared the present Semester Thesis. independently and with the aid of nothing but the resources listed in the bibliography. This thesis has neither as-is nor similarly been submitted to any other university.

Garching, November 8, 2017

Niclas Bähr

ASTER GDEM is a product of METI and NASA.

Kurzfassung

Das Ziel dieser Arbeit ist die Entwicklung eines Algorithmus zur Untersuchung des Unfallszenarios *Controlled Flight Into Terrain*. In der Praxis stützen sich die Analysen hauptsächlich auf Warnungen der sogenannten *TAWS*, Terrain Awareness and Warning Systems. In dieser Arbeit soll ein alternativer Ansatz zu den gängigen Untersuchungen erarbeitet werden. Um die Grundlage für die Analysen zu schaffen, wurden vier Geländedatenbanken miteinander verglichen: *ASTER GDEM*, *GTOPO30*, *GMTED2010* und *SRTM*. Mit der höchsten Genauigkeit, der geringsten Zahl an Fehlstellen und einer vergleichsweise großen Geländeabdeckung von Nord nach Süd konnte sich in den Testläufen die *ASTER* Datenbank durchsetzen. Mit Hilfe dieses Geländemodells überprüft der implementierte *MATLAB* - Algorithmus für jedes Sample im aufgezeichneten Flugpfad, nach welcher Zeit und an welcher Stelle das Flugzeug mit dem Terrain oder einem Hinderniss kollidieren würde, gesetzt dem Fall der ursprüngliche Kurs bliebe erhalten. Damit ergibt sich für jede aufgezeichnete Position des Flugzeugs eine Tangente an dessen Pfad, die wir *CFIT - Trajectory* nennen. Auf jeder dieser Geraden wird stetig überprüft, an welcher Stelle keine Ausweichmanöver mehr möglich sind, um einen Crash zu vemeiden, womit der *Point of no Return* erreicht ist. Der Abstand der originalen Flugposition vom *Point of no Return* bestimmt die Risikobewertung für die analysierte Stelle, wobei später die Einzelanalysen der Positionen zu einer einzigen Kennzahl für den gesamten Flug, der *CFIT - Number*, zusammengefasst werden. In den ersten Anwendungen auf Anflüge verschiedener Flughäfen können wir zeigen, dass der Algorithmus die Abweichungen von einem im Sinne des *Instrument Landing Systems* perfekten *approach* deutlich sichtbar macht und damit die Qualität der Landemanöver kategorisiert. Nach der Implementierung und Lösung der im Ausblick formulierten Herausforderungen sollten sich in Zukunft auch Flughäfen, Runways und vieles mehr hinsichtlich der neu eingeführten Kennzahlen miteinander vergleichen lassen.

Abstract

The objective of this semester thesis is the development of an algorithm in order to analyze the accident scenario *Controlled Flight Into Terrain*. In practical applications these analyses are based on the *TAWS* warnings (Terrain Awareness and Warning Systems), whereas in this thesis an alternative approach to the common investigations shall be developed. Four Terrain Elevation Databases have been compared to establish the basis of the analysis: *ASTER GDEM*, *GTOPO30*, *GMTED2010* and *SRTM*. With the highest accuracy, the least void areas and a comparatively high surface coverage of the earth, *ASTER GDEM* was found to be the best and most suitable for the current analysis. By means of this terrain model the implemented *MATLAB* algorithm calculates for every sample of the recorded flight path, where the airplane would collide with terrain or any obstacle in case the original flight path was not corrected. That's how a tangential straight line to the aircraft's path is obtained, which we call the *CFIT - Trajectory*. On the latter one, it is continuously examined, where all evasion possibilities for avoiding a crash are blocked, which marks the *Point of no Return*. The distance from the original plane position to the calculated *Point of no Return* determines the risk assessment for the analyzed point, whereby later those single evaluations of the samples are summed up to a single value for the whole flight, which we call the *CFIT - Number*. In first applications on approaches to different airports we were able to show, that the algorithm is capable of detecting significant deviations from a perfect *approach* (perfect in accordance to the *Instrument Landing System*), with which the quality of landing maneuvers can be categorized. After solving and implementing the challenges mentioned in the outlook section it shall be possible to implement the developed algorithm on different airports and runways and to compare them in terms of *CFIT - risk* using the functionalities developed and presented in this thesis.

Table of Contents

List of Figures

Table of Symbols

Underscored variables indicate non - scalar quantities.

Latin Letters		
a	deg	coordinate of $\underline{RRP}$
$\underline{c}$		center of evasion curve
C	m	circumference
$\underline{d}$		horizontal direction vector of *CFIT - Trajectory*
$\underline{DIR}$		direction vector of the runway
f_S	Hz	sampling frequency
$\underline{\mathcal{F}}$		set of final approach samples
g	m/s^2	gravity
g	m	distance projected to the ground
h	m	altitude
l	m	length
$\mathcal{L}$		Variable Length Operator
m	kg	mass
$\underline{m}$		map
$\underline{M}$		set of maps
n		load factor
$\underline{n}$		direction towards curve center
N		number of samples defining the flight path
Q		number of maps covering the flight path
Q		number of control points
r	m	radius
R		number of flight variables
$\underline{\underline{R}}$		rotation matrix
$\underline{\mathcal{R}}$		runway area
$\underline{RRP}$		Runway Reference Point
$\underline{REP}$		Runway End Point
$\underline{S}$		search area
t	s	time
T	N	thrust
$\mathfrak{T}$		Meters to Degrees Transformation Operator
V	m/s	speed
$\underline{V}$	m/s	speed vector
w	m	width
x	deg	position coordinate
x		coordinate of the $(xy)-$ frame

$\underline{X}$		flight path
y		coordinate of the $(xy)-$ frame
$\underline{z}$		flight variable
$\underline{Z}$		set of flight variables
Greek Letters		
α	deg	auxiliary angle
β	deg	bearing of the runway
γ	deg	flight path angle
δ	s	difference to optimal approach time
Δ	s	60 s (constant)
ϵ		glide ratio
ε	deg	difference angle
ζ		CFIT - Number
θ	deg	position angle inside the $(XYZ)-$frame
$\underline{\kappa}$		evasion curve
λ		length shift parameter
μ		Elevation Data Read Out Function
ξ		curve parameter on ϕ
Π		function, which calculates the time to the *PONR*
σ		standard deviation
$\underline{\tau}$		CFIT - Trajectory
φ	deg	position angle inside the $(XYZ)-$frame
$\underline{\phi}$		flare out curve
Φ		formalism to extract maps from flight path
ψ	deg	track angle
ω		curve parameter on $\underline{\kappa}$

Indices	
adj	adjacent
c	variable related to evasion curve
CP	control point
$CFIT$	Controlled Flight Into Terrain
E	longitudinal variable
H	horizontal
I	impact
l	left
M	variable related to a single map
$\underline{M}$	variable related to all maps
N	latitudinal variable
opt	optimal
PU	pull up

$PONR$	point of no return
r	right
ref	reference value
RA	radio altitude
RW	runway
S	sample
S	start point
T	trimmed vector
(xy)	$(xy)-$ frame
(XYZ)	$(XYZ)-$ inertial frame
(GPS)	WGS84 reference system
τ^*	variable related to point under investigation on $\underline{\tau}$
ϕ	variable related to flare out curve
Operator Symbols	
$\dot{x}$	time derivative of x
$\partial x/\partial y$	derivation of x with respect to y
$\vert \cdot \vert$	cardinality
$\vert \cdot \vert$	absolute value
$\Vert \cdot \Vert$	Euclidian vector norm
$\lfloor \cdot \rfloor$	rounding off operator
$\langle \cdot, \cdot \rangle$	vectorial scalar product
$\varnothing(\cdot)$	mean value
Other Symbols	
$\sqcup$	Conversion Parameter
$\underline{\oplus}$	Set containing all CFIT - incidents

1 Introduction

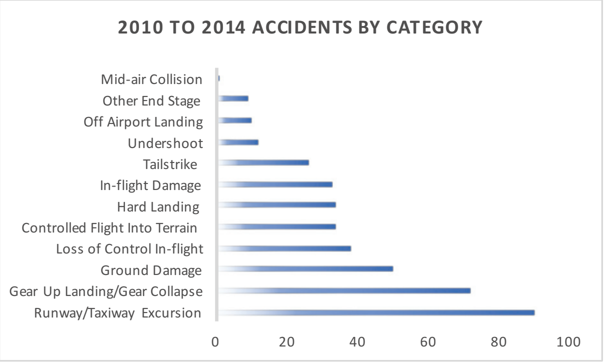
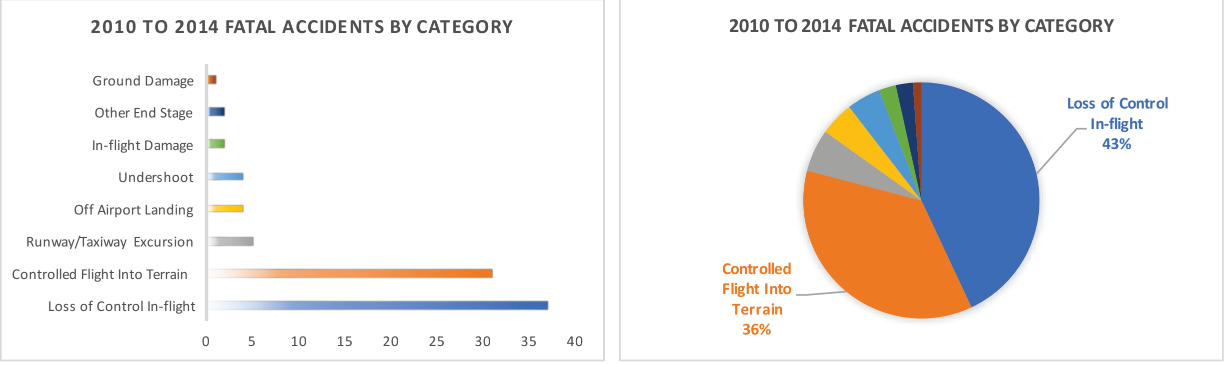


Figure 1.1: Accidents by category [12].

The technical improvements of modern aviation made the aircraft a very reliable and safe mean of transport [15]. Nevertheless, the accident risk can never be ruled out completely. In Figure 1.1 the accidents, which happened between January 2010 and December 2014, are broken down into their categories. Only hull loss or substantial damage to all jet and turboprop aircrafts with a maximum take-off weight above 5.400 kg, which were furthermore engaged in commercial aviation, are taken into account. In total 415 accidents are listed, of which 88 were fatal and resulted in 2.541 fatalities. Figure 1.2 sheds light in detail on the latter ones. One can see, that the two most dangerous categories to human life are *Loss of Control in Flight*, covering 43% of all fatal accidents, and *Controlled Flight Into Terrain* with 36 %. In the course of this semester thesis *Controlled Flight Into Terrain* (*CFIT*) shall be the subject of investigation. The *International Civil Aviation Organization* defines in [13] *CFIT* as

"In-flight collision or near collision with terrain, water, or obstacle without indication of loss of control."



(a) (b)

Figure 1.2: Fatal accidents by category [12].

Though there are only few of those *CFIT* causalities each year, 91% of them end deadly. Since there is no loss of control, typically the aircraft does not indicate any malfunction. The contributing factors are rather flight crew errors, undesired aircraft

states, environmental threats (such as meteorology) or deficiencies in technology and equipment [12].
Some airlines already examine the contributing factors to *CFIT*. In practical applications these analyses are based on the *TAWS* warnings (Terrain Awareness and Warning Systems). They can be incorrect due to wrong settings or obsolete terrain elevation information. It is for this reason that an alternative approach to the common investigations shall be developed in this thesis. The objective is to use recorded flight data to estimate the *CFIT - Risk* of a particular flight, in other words tackling the problem from an *FDM* prespective [7]. The *Flight Data Monitoring* (*FDM*) software collects huge amounts of data. The parameters of interest are recorded on board the aircraft during flights with a device called *Quick Access Recorder*. Throughout maintenance they are downloaded and stored in the airlines' databases for later investigations of threshold exceedances and trends [6].
The basis of this thesis is established in chapter 2, where a terrain database is chosen. Subsequently, the developed algorithm is explained in chapter 3 followed by the *Results and Discussion* section. In closing, chapter 5 gives an outlook to future improvements and sums up the the work in a brief paragraph. Note, that the numbers and facts of this introduction were taken from [12].

2 Terrain Elevation Databases

In this chapter the evaluation basis is built, before the algorithm, which has been developed for the analysis of the *CFIT - Risk*, is explained in detail. In order to check whether an aircraft came close to terrain or any obstacle, information about the surrounding topography is needed. In the scope of this thesis the following four, freely available databases have been subject of investigation: *ASTER GDEM*, *GMTED 2010*, *GTOPO30* and *SRTM*. Access to all of them is given via [5], the so called *Earth Explorer*. The following first section of this chapter shall give a brief introduction with some useful background information to the mentioned elevation models.

2.1 Overview

Every single database provides elevation data of the earth's surface in different resolution and accuracy. The user can read out the terrain elevation at certain points (given *latitude* and *longitude*) on the earth with the help of additional programs. We decided to use *MATLAB's Mapping Toolbox* to read and visualize the terrain data as this program provides an easier interface to deal with such data type. The data is delivered in different formats, but here we restricted the analysis to *GeoTIFF* files, which contain not only the terrain height at every raster point, but also information about the map projection and the spatial reference system [16].

ASTER GDEM

ASTER is the name of a Japanese instrument on NASA's *Terra* - Satellite (cf. Figure 2.1), which was launched in 1999. It stands for **A**dvanced **S**paceborne **T**hermal **E**mission and **R**eflection Radiometer. Since 2000, stereo pictures of the earth's surface are taken with a near infrared camera, which build the basis for the creation of a digital elevation model. Soon after the release of the first **G**lobal **D**igital **E**levation **M**odel (GDEM) in June 2009, NASA's scientists brought up the second version in October 2011, which contains improvements in accuracy, void - area patching and which was used in this thesis. It covers the surface of the earth from $83^o N$ to $83^o S$ with a resolution of 1 arc - second in 1^o - by - 1^o tiles [2]. Based on information received from NASA during the development of this thesis, the third version of ASTER is planned to be published in later 2017 or early 2018. Note:

"A digital terrain model (DTM) depicts the topographical surface of the ground. This can be contrasted to a digital elevation model (DEM), which also includes all the objects on that surface, such as vegetation and buildings." [11]

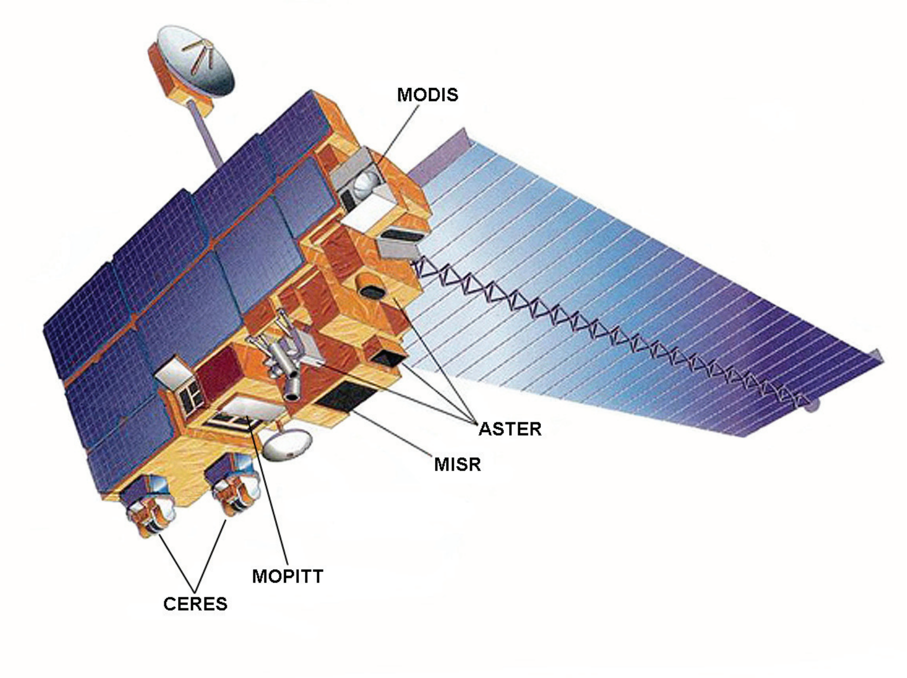


Figure 2.1: NASA's *Terra* Satellite [19].

GTOPO30

GTOPO30 is the oldest of the investigated databases, since it was already finished in 1996 by the U.S. Geological Survey's (USGS) EROS Data Center in Sioux Falls, South Dakota. The elevation model was created from different "raster and vector sources of topographic information" within a period of three years. When it comes to resolution, *GTOPO30* has nothing in common with modern satellite topography databases: It is meshed at 30 arc - seconds, which amount to approximately 1 kilometer. One could see an advantage in the complete coverage of the earth from $90^o N$ to $90^o S$. The main part of the earth's surface (excluding Antarctica) is delivered in 27 tiles of 50 degrees *latitude* times 40 degrees *longitude*, whereas the Antarctica part comes in 6 more tiles of 30 times 60 degrees.

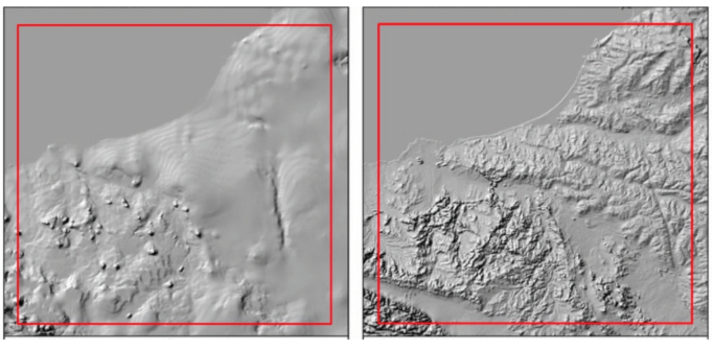


Figure 2.2: Comparison of *GTOPO30* (left) and *GMTED2010* (right) [14].

GMTED2010

The **G**lobal **M**ulti - resolution **T**errain **E**levation **D**ata **2010** database was meant to replace *GTOPO30*. The USGS and the National Geospatial - Intelligence Agency (NGA) created *GMTED2010* in collaboration as a patchwork of 11 raster - based elevation data sources, of which *SRTM* is the most important. The user can choose (for most parts of the world) in between a resolution of 30, 15 or 7.5 arc - seconds (1000, 500 or 250 meters), which cover the earth's surface from 84^oN to 56^oS and with the lower resolution even up to 90^oS. Since *GMTED2010* is put together from many different databases, the choice is between different aggregation methods: minimum elevation, maximum elevation, mean elevation, median elevation, standard deviation of elevation, systematic subsample, and breakline emphasis. For the analysis in the following two sections *mean elevation* was chosen. The product can be downloaded in 40 times 30 degrees tiles [14]. One can see that *GMTED2010* gives a much more detailed terrain information compared to *GTOPO30* even at a resolution of 30 arc - seconds, which is displayed in Figure 2.2.

SRTM

The **S**huttle **R**adar **T**opography **M**ission started on the 11th of February in 2000, with the launch of Space Shuttle *Endeavour*. The mission: recording the topographic surface of the earth with a technique called *single pass interferometry*. Similar to stereo imaging, two pictures from slightly different positions are taken in order to compute the surface elevation. In interferometry microwaves are used, which has the advantage, that no external light source is needed and weather conditions, such as clouds, cannot influence the measurements, thus imaging is possible at any time. The space shuttle carried two instruments on board: one transmitter and receiver antenna and another receiver, which was mounted in a distance of 60 m to the shuttle at the end of an extension mast. This way, the microwaves, which are reflected from the surface of the earth, can be recorded in two slightly different spatial positions at the same time, or in other words in the same, *single pass*. It should be pointed out, that the explained process stands in contrast to *repeat pass interferometry*, where two "pictures" are taken at different points in time from different orbits. The mission, which took 11 days, was a joint venture of NASA and the German DLR. It covers the surface of the earth only from 60^oN to 57^oS, due to the orbit of *Endeavour*, with a resolution of 1 arc - second in 1^o - by - 1^o tiles [11] [18].

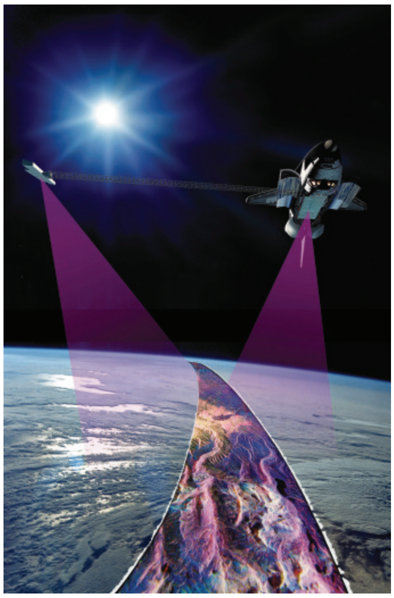


Figure 2.3: Space shuttle with mast [17].

2.2 Comparing Mountainous Areas

To evaluate which of the presented databases was appropriate to meet the needs of a flight analysis algorithm, a few test cases have been generated in order to check the correctness of the field elevation at certain geographical control points. The following table (2.1) shows 25 famous mountains together with their summit heights and summit coordinates from all around Europe. The summit coordinates were extracted from *Google Maps*, whereas the heights of the mountains can be found on *Wikipedia*.

Mountain	**Country**	**Height**	**Latitude**	**Longitude**
Acherkogel	Austria	3007 m	47.189261°	10.956065°
Ararat	Turkey	5137 m	39.702603°	44.299526°
Ben Nevis	Scotland	1345 m	56.796900°	-5.0036°
Carrantuohill	Ireland	1039 m	51.999100°	-9.7433°
Corno Grande	Italy	2912 m	42.469312°	13.564998°
Dachstein	Austria	2995 m	47.475154°	13.605624°
Demirkazik Dagi	Turkey	3756 m	37.836524°	35.144446°
Elbrus	Russia	5642 m	43.351722°	42.442077°
Etna	Italy	3329 m	37.751739°	14.995°
Galdhøpiggen	Norway	2469 m	61.636099°	8.3131260°
Janga	Georgia	5051 m	43.018422°	43.056420°
Kaskasapakte	Sweden	2043 m	67.942106°	18.579762°
Mont Blanc	France	4810 m	45.832543°	6.8651500°
Monte Cinto	France	2706 m	42.379642°	8.945751°
Mytikas	Greece	2918 m	40.088409°	22.358569°
P. d. Midi d'Ossau	France	2884 m	42.843623°	-0.4378°
Pico del Teide	Spain	3718 m	28.272409°	-16.643°
Psiloritis	Greece	2456 m	35.22600°	24.770800°
Punta La Marmora	Italy	1834 m	39.986916°	9.324055°
Seebergspitze	Austria	2085 m	47.466029°	11.679711°
Slieve Donard	N. Ireland	849 m	54.18000°	-5.9206°
Snowdon	Wales	1085m	53.0684°	-4.0763°
Uschba	Georgia	4737m	43.123992°	42.659819°
Vesuv	Italy	1281 m	40.82240°	14.42890°
Zugspitze	Germany	2962 m	47.421084°	10.985315°

Table 2.1: List of investigated mountains/control points.

Since the summit coordinates were extracted manually from *Google Earth* it could not be concluded with absolute certainty, that the true geographical summit of the mountain would be located on that exact point in space. It is for this reason, why a search algorithm had to be introduced. The following enumeration should outline the analysis procedure for a single control point CP. Note that the index in CP_i, for processing multiple control points, is omitted for the sake of simplicity.

1. Define control point to be analyzed by height and GPS position from *Google Earth*: $(x_{N,CP}, x_{E,CP}, h_{CP})^T =: \underline{x}_{(GPS),CP}$.
2. Load fitting map tiles of *ASTER GDEM*, *GMTED 2010*, *GTOPO30* and *SRTM* , which include the control point.
3. Define search area: $\underline{S} := [x_{N,CP} - 0.05^o, x_{N,CP} + 0.05^o] \times [x_{E,CP} - 0.05^o, x_{E,CP} + 0.05^o]$.
4. Find the position and elevation of the highest point within the defined search area on each map $\underline{m}_{(\cdot)}$, where $(\cdot)$ is a placeholder for the map name. μ extracts the terrain height on a map at a given position.

   $$\underline{x}_{(GPS),CP,(\cdot)} := \underset{\forall \underline{x}_{(GPS)} \in \underline{S}}{\operatorname{argmax}} \left\{ \mu\left(\underline{x}_{(GPS)}, \underline{m}_{(\cdot)}\right) \right\}.$$

   Please see equations (3.1) (p. 15) and (3.20) (p. 22) for a detailed introduction and definition of the used symbols. This step delivers the four "map summit heights" (corrected control point):

   $$h_{CP,(ASTER)}, h_{CP,(SRTM)}, h_{CP,(GTOPO)}, h_{CP,(GMTED)}.$$

5. Calculate height deviation for every database

   $$\Delta h_{CP,(\cdot)} := h_{CP,(\cdot)} - h_{CP}.$$

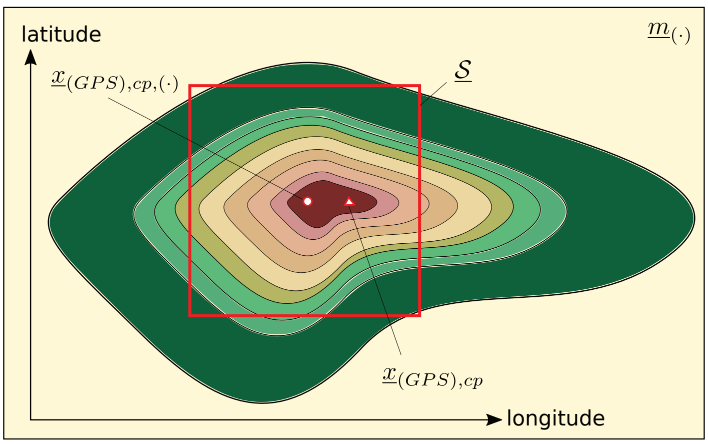


Figure 2.4: Control point and corrected control point within a topographical environment.

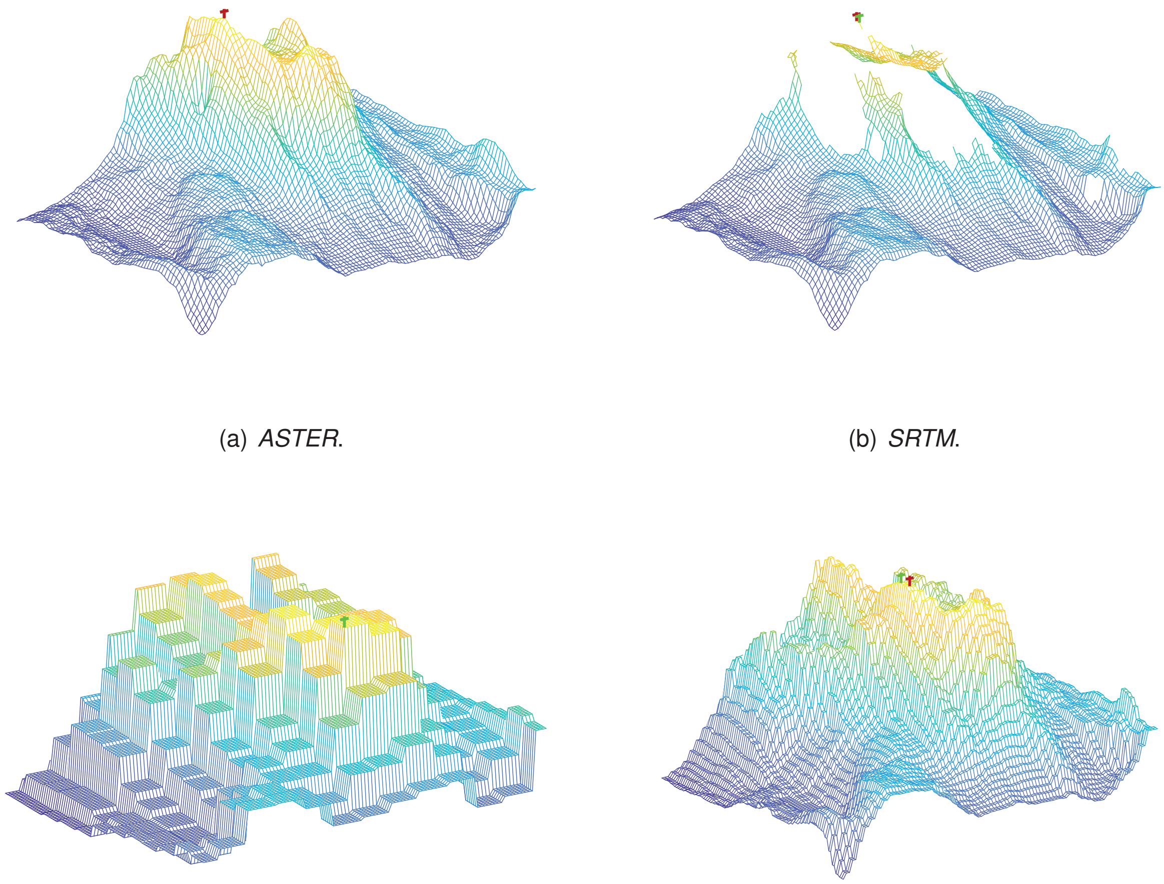
The described procedure is repeated for every control point CP_i with $i = 1, ..., Q$, whereby a mean deviation can be finally calculated:

$$\Delta\bar{h}_{CP,(\cdot)} = \frac{1}{Q}\sum_{i=1}^{Q}|\Delta h_{CP_i,(\cdot)}|$$

Indeed, it is possible that for some constellation in mountainous terrain the chosen size of the search area is too big, such that the algorithm would find another summit on another mountain. For the presented 25 mountains, the author made sure via visual inspection of the results, that the corrected control point is still located on the mountain of interest.

Mountain	$\Delta h_{CP,(ASTER)}$	$\Delta h_{CP,(SRTM)}$	$\Delta h_{CP,(GTOPO)}$	$\Delta h_{CP,(GMTED)}$
Acherkogel	-20 m	-27 m	-144 m	-56 m
Ararat	1 m	-28 m	-199 m	-47 m
Ben Nevis	7 m	-14 m	-19 m	-21 m
Carrantuohill	-36 m	-18 m	-37 m	-30 m
Corno Grande	-36 m	-16 m	-403 m	-41 m
Dachstein	-117 m	-214 m	-273 m	-182 m
Demirkazik Dagi	-63 m	-52 m	-129 m	-97 m
Elbrus	23 m	-25 m	-159 m	-45 m
Etna	-27 m	-9 m	-103 m	-41 m
Galdhøpiggen	-5 m		-198 m	-38 m
Janga	-91 m	-105 m	-236 m	-124 m
Kaskasapakte	13 m		-150 m	-8 m
Mont Blanc	-11 m	-146 m	-274 m	-97 m
Monte Cinto	-39 m	-55 m	-243 m	-95 m
Mytikas	-32 m	-47 m	-208 m	-58 m
P. d. Midi d'Ossau	-121 m	-53 m	-455 m	-140 m
Pico del Teide	-27 m	-29 m	-17 m	-73 m
Psiloritis	-18 m	-10 m	-27 m	-35 m
Punta La Marmora	-18 m	-11 m	-112 m	-27 m
Seebergspitze	-41 m	-9 m	-408 m	-78 m
Slieve Donard	7 m	-9 m	-96 m	-24 m
Snowdon	-23 m	-27 m	-17 m	-56 m
Uschba	-91 m	-161 m	-429 m	-213 m
Vesuv	-37 m	-10 m	-128 m	-49 m
Zugspitze	-11 m	-20 m	-303 m	-153 m

Table 2.2: List of height deviations.



(a) *ASTER*.

(b) *SRTM*.

(c) *GTOPO30*.

(d) *GMTED2010*.

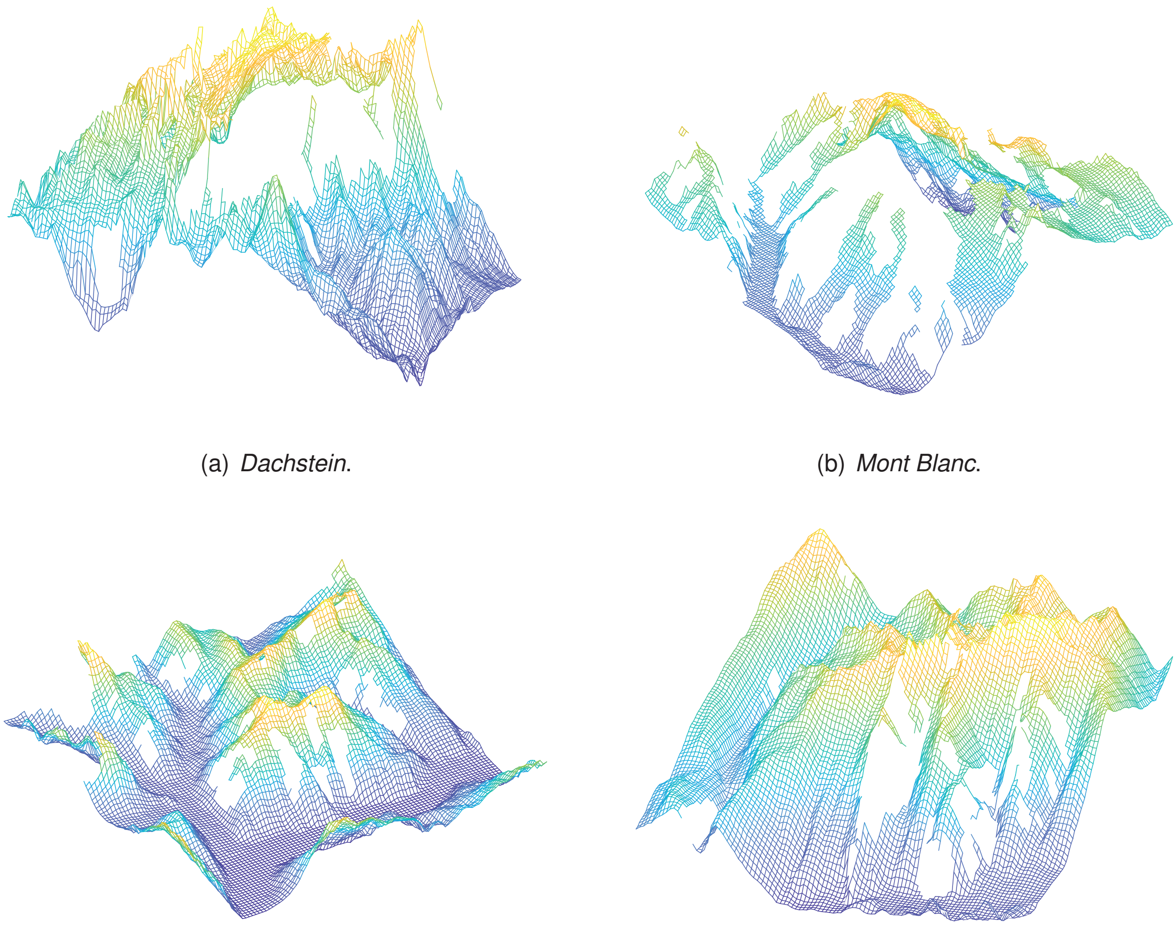
Figure 2.5: The mountain *Zugspitze* in different databases: *Google Earth* summit location marked by red cross, corrected coordinates marked by green cross.

Table 2.2 lists the computed deviations of height as defined above for every database and mountain under investigation. Note that there was no *SRTM* data available for *Galdhøpiggen* and *Kaskasapakte* since they are located above $60^o N$ *latitude*. The results are printed below in the mean deviations table 2.10:

$\Delta\bar{h}_{CP,(ASTER)}$	$\Delta\bar{h}_{CP,(SRTM)}$	$\Delta\bar{h}_{CP,(GTOPO)}$	$\Delta\bar{h}_{CP,(GMTED)}$
37 m	48 m	191 m	73 m

Table 2.3: List of mean absolute deviations.

It is clear to see that the low resolutions of *GTOPO30* and *GMTED2010* lead to worse elevation accuracy. *ASTER* is the most accurate closely followed by *SRTM*. But what about the mesh quality and the rest of the surrounding terrain. Solely by looking at



(a) *Dachstein*.
(b) *Mont Blanc*.
(c) *Seebergspitze*.
(d) *Acherkogel*.

Figure 2.6: Void data in the *SRTM* database.

one point on the grid, it is impossible to make a certain statement about the depiction of the terrain. Thus, take a look at Figure 2.5, where one can see how the mountain *Zugspitze* is mapped in the investigated databases. The *Google Earth* summit location is marked by a small red cross, whereas the by the algorithm corrected coordinates are marked with a green cross. Note, that the green cross of *ASTER* is exactly behind the red cross, which makes the former one invisible. *GTOPO's* red cross is behind one of the big blocks in the front and could thus not be seen, either. Although *ASTER* and *SRTM* are quite close in terms of the mean height deviation, the *SRTM* is unfit for a flight analysis algorithm because of the missing data in the grid. Since it is theoretically possible that there is only coincidentally void data on *SRTM's* mapping of the *Zugspitze*, the areas around the other 22 *SRTM* control points have been checked for holes as well (cf. Figure 2.6). All in all 5 of the 23 *SRTM* samples were corrupted, which is a rate of 22%. This supports the suspicion, that even though *SRTM* provides a *void* *filled* version (which was of course taken for this evaluation), there are still lots of areas without adequate coverage.

2.3 Comparing the Most Dangerous Areas

The purpose of this last section is not only to further evaluate the quality of the different databases with more control points, but also to emphasize the importance of *CFIT - Analysis* in modern aviation with some eye - opening facts about the most dangerous airports in the world. After a few words about every airport and its specialties, which were taken from [4], the reader can find an example visualization from one of the investigated databases for each of them. The *runway area* is marked in the color orange (please proceed to section 3.3.2 for a detailed description of the marking process). Field elevation of the airports as well as the data depicted in the tables next to the Figure comes from [1]. Note that the airport field elevation (h_{CP}) is measured at the corresponding *aerodrome reference point*, *ARP*.

Madeira The approach to the Portuguese island Madeira can be seen as one of the most dangerous in the world: The pilot flies straight towards a mountain and only turns in direction to the runway during the last minute. The runway itself is built on columns, which are anchored in the sea and if that was not enough the cockpit crew often has to struggle with strong downbursts. For this reason *go - arounds* after a *missed approach* are common during bad weather. Calculated height deviations $\Delta h_{CP,(\cdot)}$ from an airport field elevation of $h_{CP} = 59\,m$: *ASTER*: 3 m , *SRTM*: -3 m, *GTOPO30*: 104 m, *GMTED2010*: 5 m. It is pointed out that the *SRTM* mapping, depicted in Figure 2.7, contains again void data.

Country	**IATA - Code**
Portugal	FNC
Latitude	**Longitude**
32.697899°	-16.7745°

Table 2.4: Aeroporto Internacional d. Madeira.

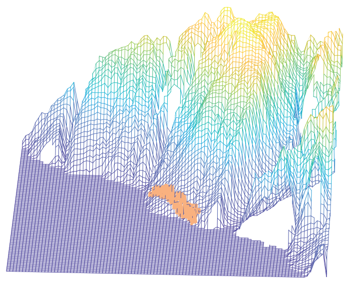


Figure 2.7: Madeira's Airport in *SRTM*.

Gibraltar Gibraltar's runway is bedded in the sea and comes hazardously close to the famous big Rock of the British Overseas Territory, which additionally causes air turbulence. The only street connecting Spain with Gibraltar passes over the taxiway. Aircraft have the right of way here. Calculated height deviations $\Delta h_{CP,(\cdot)}$ from an airport field elevation of $h_{CP} = 4\,m$: *ASTER*: 3 m , *SRTM*: -1 m, *GTOPO30*: 16 m, *GMTED2010*: -2 m.

Country	**IATA - Code**
Spain	GIB
Latitude	**Longitude**
36.151219°	-5.349664°

Table 2.5: Gibraltar Airport.

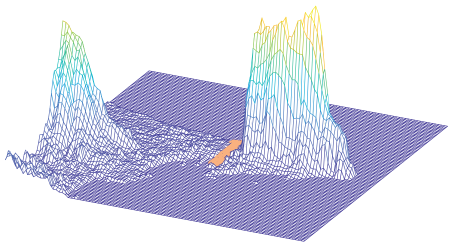


Figure 2.8: Gibraltar's Airport in *ASTER*.

Lukla The airport is located at the foot of the Mount Everest on a 600 m short runway which is bordered on one side by a steep mountain wall and on the other side by a 1000 m deep cliff. Since the air is too thin to generate the needed lift for takeoff, the pilots tumble the machines down the cliff and pull them right up before the next rock wall. In October 2008 18 people died in a tragic accident on this airport [4]. Calculated height deviations $\Delta h_{CP,(\cdot)}$ from an airport field elevation of $h_{CP} = 2846\ m$: *ASTER*: -21 m , *SRTM*: -25 m, *GTOPO30*: -360 m, *GMTED2010*: -26 m.

Country	**IATA-Code**
Nepal	LUA
Latitude	**Longitude**
27.686944°	86.729722°

Table 2.6: Tenzing - Hillary Airport.

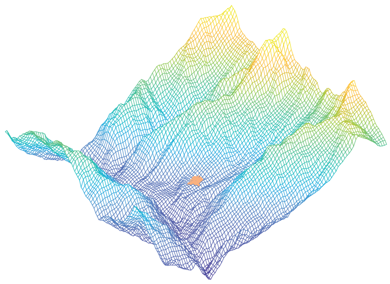


Figure 2.9: Lukla's Airport in *ASTER*.

Tegucigalpa With a short runway of almost 2000 m only aircraft up to the size of a Boeing 757 are allowed to land in Tegucigalpa, Honduras [4]. The surrounding mountains force the pilot on some risky curves shortly before landing with a very steep glide slope. Calculated height deviations $\Delta h_{CP,(\cdot)}$ from an airport field elevation of $h_{CP} = 1004\ m$: *ASTER*: -15 m, *SRTM*: -9 m, *GTOPO30*: -4 m, *GMTED2010*: -3 m.

Country	**IATA - Code**
Honduras	TGU
Latitude	**Longitude**
14.060883°	-87.217197°

Table 2.7: Aeropuerto Internacional Toncontín.

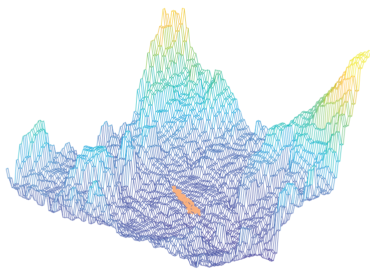


Figure 2.10: Honduras' Airport in *GMTED*.

Courchevel The little aerodrome in Courchevel is directly located beneath a ski piste. With a runway slope of 10.5° and only 600 m space to come to a full stop it is obvious, why big passenger machines do not fly to this airport. The takeoff reminds of ski jumping and the steep mountainous scenery made the airfield part of already two *"James Bond"* movies. Calculated height deviations $\Delta h_{CP,(\cdot)}$ from an airport field elevation of $h_{CP} = 2007\ m$: *ASTER*: -6 m , *SRTM*: -25 m, *GTOPO30*: -169 m, *GMTED2010*: -8 m.

Country	**IATA - Code**
France	CVF
Latitude	**Longitude**
45.3974°	6.6348°

Table 2.8: Altiport de Courchevel.

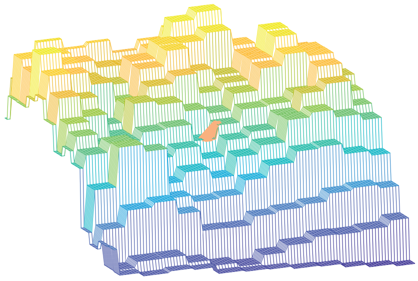


Figure 2.11: Courchevel in *GTOPO30*.

Innsbruck Pilots, who want to fly to Innsbruck, need a special simulator training (which is the same for most of the other mentioned airports). After takeoff, the aircraft has to climb quickly to pass the oncoming mountain summits. Calculated height deviations $\Delta h_{CP,(\cdot)}$ from an airport field elevation of $h_{CP} = 581\ m$: *ASTER*: -1 m , *SRTM*: -1 m, *GTOPO30*: -13 m, *GMTED2010*: -7 m. Also this *SRTM* grid is corrupted:

Country	**IATA-Code**
Austria	INN
Latitude	**Longitude**
47.260219°	11.343964°

Table 2.9: Flughafen Innsbruck.

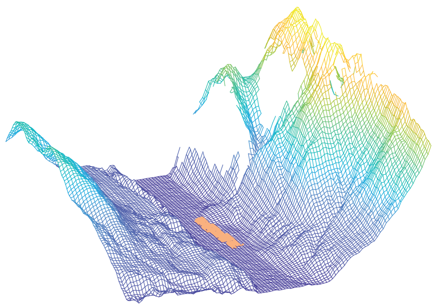


Figure 2.12: Innsbruck's Airport in *SRTM*.

Results The following table shows the mean absolute height deviations for every database of the six analyzed airports:

$\Delta\bar{h}_{CP,(ASTER)}$	$\Delta\bar{h}_{CP,(SRTM)}$	$\Delta\bar{h}_{CP,(GTOPO)}$	$\Delta\bar{h}_{CP,(GMTED)}$
8 m	13 m	111 m	9 m

Table 2.10: List of mean absolute deviations.

2.4 Conclusions of the Database Analysis

Applying the four Terrain Elevation Databases to airports could show that height deviations significantly decrease when it comes to flatter areas, such as the terrain, where airfields are built on. Furthermore, we can not assume with absolute certainty that the summit heights are exact, whereas field elevations of airports are normally well documented. Due to its low resolution, *GTOPO30* is definitely not suitable for the application of the *CFIT - algorithm*. Its successor, *GMTED2010*, has the advantage of a big tile size. This would cover in most cases already the whole *flight path* and could economize computational time, since there is no map selection process needed. Nevertheless, it needs to be mentioned, that the mean height deviation of *GMTED2010* for mountainous control points was almost double compared to *ASTER* and 1.5 times the mean deviation of *SRTM*. As a result, the database selection focuses on the latter two. In total 29 *SRTM* control points, 23 summits and 6 airports, were examined in the last sections, of which all in all 7 were corrupted. Whereby the total void data rate of *SRTM* accounts to 24%. It should now be obvious, why *ASTER* has been chosen as the database of this thesis: it is not only more exact in terms of elevation accuracy, but it also covers a big area of the earth in a high resolution, and convinced in the field test with mappings free from corruption.

3 CFIT - Analysis - Algorithm

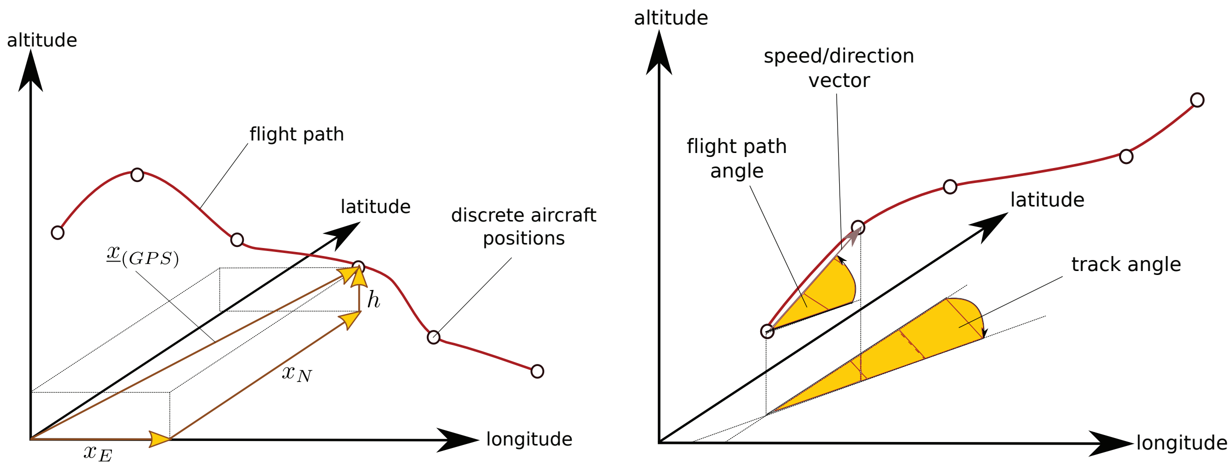
In the scope of this thesis an algorithm has been developed which enables the user to analyze the given flight data in regards to the accident scenario *Controlled Flight Into Terrain* (CFIT). We will briefly explain the basic functionality as an overview for the reader, before the several sections of the MATLAB based program are explained in detail later on.
With the data from the *FDM - recorder* the *flight path* of the aircraft can be reconstructed in a virtual topographic environment made available by any satellite database. It is obvious that the risk of crashing the aircraft into terrain must increase, if the *flight path* is close to any obstacle. Thus, the proximity to any obstacle for every time step from takeoff to touchdown has to be evaluated. We take one single point on the *flight path* and pose the following question: What would happen, if the pilot did not correct the current flight path or speed and flew further on with the same configuration. The answer to this question is the tangential straight line to the *flight path* at this considered point in time and returns a characteristic *time to impact*. One could measure the risk of crashing into terrain only by the *time to impact*, but on closer examination it is apparent that a *Point of no Return* exists (seconds before the impact), where the accident risk is already 100%. The goal of the analysis is to assess the risk of a certain flight regarding CFIT in only one number. Therefore, the algorithm evaluates the distance to the *Point of no Return* for every given aircraft position and sums the single risk values up to one value (normalized by the number of samples), which we call the *CFIT - Number*. In the beginning of the program the variables relevant for the *CFIT - analysis* are read out from the *FDM* - file of the selected flight.

Def.: Position The position of the aircraft in space $\underline{x}_{(GPS)}$ is described by the GPS - *latitude* $x_N \in [-90^o, 90^o]$, the GPS - *longitude* $x_E \in [-180^o, 180^o]$ and the *barometric altitude* $h \in (0, \infty)$.

$$\underline{x}_{(GPS)} := \begin{pmatrix} x_N \\ x_E \\ h \end{pmatrix} \tag{3.1}$$

By the index (GPS), it is indicated, that position specifications are given in the above defined format, though it is clear, that the *barometric altitude* is not measured by *GPS*.

Def.: Direction The direction of the *flight path* $\dot{\underline{x}}$, which can be seen as a tangential vector or in terms of discrete calculations a vector pointing towards the next position,



(a) Aircraft position values.

(b) Aircraft direction values.

Figure 3.1: Depiction of position and direction.

is given by the *track angle* ψ and the *flight path angle* γ. $V = \|\dot{\underline{x}}\|$ denotes the speed of the aircraft.

$$\dot{\underline{x}} = V \begin{pmatrix} \cos\gamma\cos\psi \\ \cos\gamma\sin\psi \\ \sin\gamma \end{pmatrix} \tag{3.2}$$

Since we obtain the *ground speed*, also known as the *horizontal speed* $V_H = V\cos\gamma$, from the *FDM* - file, the last expression changes to

$$\dot{\underline{x}} = V_H \begin{pmatrix} \cos\psi \\ \sin\psi \\ \tan\gamma \end{pmatrix}. \tag{3.3}$$

It is already apparent, that position calculations in the form of expressions like

$$\hat{\underline{x}}_{(GPS)} = \underline{x}_{(GPS)} + \Delta t \cdot \dot{\underline{x}}$$

are not possible due to unit inconsistencies. The implemented transformation formalism to solve this problem will be discussed later. In order to detect the time points takeoff and touchdown the *radio altitude* h_{RA} (given by the first sensor) and the *climb rate* $\dot{h}$ are read out as well. Keep in mind that the following definitions were made in the source code.

Def.: Takeoff The takeoff is defined as the time point, where the first significant *climb rate* is measured ($\dot{h} > 2\frac{m}{s}$).

Def.: Touchdown The touchdown is set to the point where the *radio altitude* drops below 5 meters after 50% of the samples of the flight ($h_{RA} \leq 5m$).

3.1 Vector Trimming

The variable read out returns R flight variables (such as *barometric altitude*, *longitude*, *latitude*, *speed* ...).

Def.: Set of Variables A set $\underline{Z}$ of R flight variables $\underline{z}_i$ is defined as specified below:

$$\underline{Z} := \{\underline{z}_i\}_{i=1}^{R} = \left\{ \begin{pmatrix} z_{1,i} \\ \vdots \\ z_{j,i} \\ \vdots \\ z_{l_i,i} \end{pmatrix} \right\}_{i=1}^{R} \text{ with } R \in \mathbb{N},\ \underline{z}_i \in \mathbb{R}^{l_i} \text{ and } l_i \in \mathbb{N} \tag{3.4}$$

The vector $\underline{z}_i$ contains all the recorded values for this specific variable from the beginning of the flight until the end. Thus, in practical application we are talking about several thousands of entries. In this context, speaking of the length of the variables $\underline{z}_i$ does not refer to the classical vector length defined by the *Euclidian norm* and motivates the introduction of a Variable Length Operator.

Def.: Variable Length Operator

$$\mathcal{L}(\underline{z}_i) := \dim(\mathbb{R}^{l_i}) = l_i \tag{3.5}$$

Here, $\mathcal{L}$ denotes the Variable Length Operator. Furthermore, every $\underline{z}_i$ carries its own characteristic sampling frequency f_S, in which the values of the vector have been recorded.

$$f_S : \underline{z}_i \mapsto f_S(\underline{z}_i) \in \mathbb{R} \tag{3.6}$$

It is not sure that all the variables $\underline{z}_i$, that are obtained from the flight recorder, have

1. the same length (*possibly* $\exists\ (i, j) : l_i \neq l_j$) and
2. are sampled with the same frequency (*possibly* $\exists\ (i, j) : f_S(\underline{z}_i) \neq f_S(\underline{z}_j)$) .

The first problem is solved by cutting the vectors to the same length l_{min} and deleting the protruding entries.

$$l_{min} := \min_{i=1...R} \{\mathcal{L}(\underline{z}_i)\} \tag{3.7}$$

Def.: Cutting Operation

$$\hat{\underline{Z}} := \left\{ \underline{z}_i \Big|_{j=1...l_{min}} \right\}_{i=1}^{R} \tag{3.8}$$

Here, $(\cdot)\Big|_{j=1...l_{min}}$ symbolizes, that only values until the length of l_{min} are further considered. Afterwards, the algorithm checks for every row, if all variables have a valid entry. If not, the whole row is deleted. Formally:

Def.: Deleting Operation Start at $j = 1$. Consider $z_{j,i}\ \forall i$. **IF** $z_{j,i} \in \mathbb{R}\ \forall i\ \rightarrow$ go on with $j+1$. **ELSE** delete row j. The final results are the trimmed variables and a trimmed set of variables denoted by the index T.

The following example should illustrate the problem. Imagine there is a set of $R = 3$ variables $\underline{z}_1, \underline{z}_2$ and $\underline{z}_3$ given

$$\underline{Z} = \left\{ \underline{z}_1 = \begin{pmatrix} z_{1,1} \\ z_{2,1} \\ z_{3,1} \end{pmatrix}, \underline{z}_2 = \begin{pmatrix} z_{1,2} \\ nan \\ z_{3,2} \\ nan \end{pmatrix}, \underline{z}_3 = \begin{pmatrix} z_{1,3} \\ z_{2,3} \\ z_{3,3} \\ z_{4,3} \end{pmatrix} \right\}$$

where nan means *not a number* and symbolizes an empty entry. We can assume that the time distance from one row to another is constant and the same for all variables (here e.g. 1 second). So the sampling frequencies f_S compute to

$$\begin{aligned} f_S(\underline{z}_1) =& 1\ Hz \\ f_S(\underline{z}_2) =& \frac{1}{2}\ Hz \\ f_S(\underline{z}_3) =& 1\ Hz \end{aligned}$$

and the lengths in this example are

$$\mathcal{L}(\underline{z}_1) = 3,\ \mathcal{L}(\underline{z}_2) = 4 \text{ and } \mathcal{L}(\underline{z}_3) = 4.$$

The trimming process can now be divided into *cutting* and *deleting*.

Cutting The first step of solving these issues is cutting the vectors to the same length, which is the minimum length of all extracted variables. This means that the protruding entries are deleted. With $l_{min} = 3$, the cutting operation delivers the following result.

$$\underline{\hat{Z}} = \left\{ \underline{\hat{z}}_1 = \begin{pmatrix} z_{1,1} \\ z_{2,1} \\ z_{3,1} \end{pmatrix}, \underline{\hat{z}}_2 = \begin{pmatrix} z_{1,2} \\ nan \\ z_{3,2} \end{pmatrix}, \underline{\hat{z}}_3 = \begin{pmatrix} z_{1,3} \\ z_{2,3} \\ z_{3,3} \end{pmatrix} \right\}$$

Deleting In this small example, the second row has an empty nan - entry. In a continuous analysis it can be fatal, if one of the variables is not properly defined. In order to obtain well defined data, the whole second row needs to be deleted.

$$\underline{Z}_T = \left\{ \underline{z}_{1,T} = \begin{pmatrix} z_{1,1} \\ z_{3,1} \end{pmatrix}, \underline{z}_{2,T} = \begin{pmatrix} z_{1,2} \\ z_{3,2} \end{pmatrix}, \underline{z}_{3,T} = \begin{pmatrix} z_{1,3} \\ z_{3,3} \end{pmatrix} \right\}$$

Remark Due to the deletion of a whole row, the time distance from the now neighboring rows has changed from 1 second to 2 seconds. This means that in practical implementations a concurrent time stamp variable is absolutely useful and needed from the beginning of trimming the vectors.

3.2 Evaluation of the Flight Path Maps

In this chapter the selection of the fitting maps for the selected flight will be discussed and explained in detail.

Def.: Flight Path The *flight path* $\underline{X}_{(GPS)}$ is the sequence of all given positions $\underline{x}_{(GPS),S}$ of the aircraft.

$$\underline{X}_{(GPS)} := \{\underline{x}_{(GPS),S}\}_{S=1}^{N} = \left\{ \begin{pmatrix} x_{N,S} \\ x_{E,S} \\ h_S \end{pmatrix} \right\}_{S=1}^{N} \text{ with } S \leq N \in \mathbb{N} \tag{3.9}$$

The *ASTER* database provides elevation data of wide parts of the world. Since the algorithm has been exclusively developed with the use of this database the following definition of map refers to *ASTER* - maps. For the sake of simplicity, the term "*ASTER* - map" will be simply abbreviated with "map" in the following.

Def.: (*ASTER*) - Map A single map $\underline{m}$ provides the elevation data for a region of 1^o *latitude* times 1^o *longitude*. It is named after its southwest corner, so it is uniquely defined by only two GPS - position values $m_N \in [-90^o, 90^o]$ and $m_E \in [-180^o, 180^o]$ (see Figure 3.2).

$$\underline{m} = \begin{pmatrix} m_N \\ m_E \end{pmatrix} \tag{3.10}$$

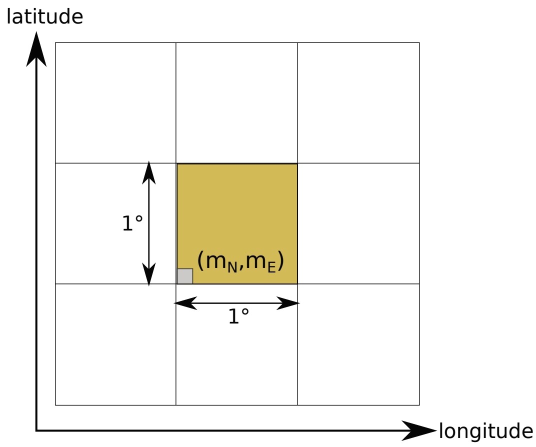


Figure 3.2: ASTER - maps.

Def.: Flight Path Maps All the maps that are needed to cover the *flight path* completely are called *flight path maps*. They are bundled in the sequence $\underline{M}$.

$$\underline{M} := \{\underline{m}_j\}_{j=1}^{Q} = \left\{ \begin{pmatrix} m_{N,j} \\ m_{E,j} \end{pmatrix} \right\}_{j=1}^{Q} \text{ with } Q \in \mathbb{N} \tag{3.11}$$

Indeed, it is depending on the sampling frequency of the *FDM* - system, but one can safely assume that the following statement is valid.

$$Q \ll N \tag{3.12}$$

We are looking for a formalism Φ that extracts the *flight path* maps $\underline{M}$ from the *flight path* $\underline{X}_{(GPS)}$.

$$\Phi : \underline{X}_{(GPS)} \mapsto \{\underline{m}_j\}_{j=1}^{Q} = \underline{M} \tag{3.13}$$

The third coordinate h_S of the aircraft positions $\underline{x}_{(GPS),S}$ can be neglected in regards to Φ, since the height has no influence on the choice of the map. Without introducing a new symbol the height will be omitted in the following considerations.
Every position $\underline{x}_{(GPS),S}$ can be uniquely assigned to a map $\underline{m}_i$, by rounding the *latitude* x_N and *longitude* x_E down to the nearest integer towards $-\infty$. In other words, this is the way how we find the southwest corner of the associated map. The fitting operator is introduced in the following:

Def.: Rounding Off Operator Let y be $\in \mathbb{R}$ then the rounding off operator is defined as

$$\lfloor y \rfloor := \max\{k \in \mathbb{Z} | k \leq y\}. \tag{3.14}$$

With this operator the association of a position $\underline{x}_{(GPS)}$ to a map $\underline{m}$ can be easily expressed:

$$\underline{m} = \lfloor \underline{x}_{(GPS)} \rfloor = \begin{pmatrix} \lfloor x_N \rfloor \\ \lfloor x_E \rfloor \end{pmatrix}. \tag{3.15}$$

Applying solely this rounding down operation to the whole *flight path* would lead to a map number Q that is equal to the number of samples in the path N. This is what leads us to the next step for completing the formalism Φ: the additional condition, that every map $\underline{m}_i$ in the *flight path* map sequence $\underline{M}$ has to be unique. Mathematically expressed:

$$\forall (i,j): \ \underline{m}_i \neq \underline{m}_j \text{ with } i \neq j \tag{3.16}$$

In the first step the rounding down operation is applied to all the points on the *flight path*.

$$\underline{\hat{M}} = \{\underline{m}_S\}_{S=1}^N = \left\{ \begin{pmatrix} \lfloor x_{N,S} \rfloor \\ \lfloor x_{E,S} \rfloor \end{pmatrix} \right\}_{S=1}^N \tag{3.17}$$

In the second step the doubles/triples (...) of the maps are deleted and we end up with the *flight path* maps.

$$\underline{M} = \{\underline{m}_i \in \underline{\hat{M}} \,\big|\, \forall (i,j) \colon \underline{m}_i \neq \underline{m}_j,\, i \neq j\} = \{\underline{m}_j\}_{j=1}^Q \tag{3.18}$$

The desired formalism Φ can be summed up as follows:

$$\{\underline{m}_j\}_{j=1}^Q = \Phi(\underline{X}_{(GPS)}) = \left\{\{\lfloor \underline{x}_{(GPS),S} \rfloor\}_{S=1}^N\right\}_{\underline{m}_i \neq \underline{m}_j}^{\forall (i,j), i \neq j} \tag{3.19}$$

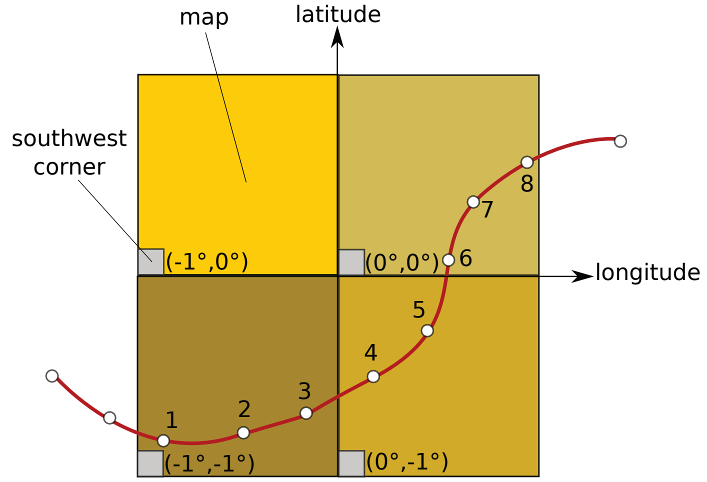


Figure 3.3: Exemplary depiction of a *flight path* in four example maps.

Example In Figure 3.3 a small example has been set up for the better understanding of the formally described algorithm of choosing the fitting maps. The discrete aircraft positions from 1 to 8 mark the *flight path*:

$$\underline{X}_{(GPS)} := \{\underline{x}_{(GPS),S}\}_{S=1}^{8} = \left\{ \begin{pmatrix} -0.85^o \\ -0.85^o \end{pmatrix}, \begin{pmatrix} -0.4^o \\ -0.8^o \end{pmatrix}, \begin{pmatrix} -0.2^o \\ -0.65^o \end{pmatrix}, \begin{pmatrix} 0.2^o \\ -0.5^o \end{pmatrix}, \right. \\ \left. \begin{pmatrix} 0.4^o \\ -0.2^o \end{pmatrix}, \begin{pmatrix} 0.5^o \\ 0.1^o \end{pmatrix}, \begin{pmatrix} 0.6^o \\ 0.4^o \end{pmatrix}, \begin{pmatrix} 0.95^o \\ 0.7^o \end{pmatrix} \right\}$$

As described above, in a first step the rounding down operation is applied to all given discrete positions.

$$\underline{\hat{M}} = \{\underline{m}_S\}_{S=1}^{8} = \{\lfloor \underline{x}_{(GPS),S} \rfloor\}_{S=1}^{8} = \left\{ \begin{pmatrix} -1^o \\ -1^o \end{pmatrix}, \begin{pmatrix} -1^o \\ -1^o \end{pmatrix}, \begin{pmatrix} -1^o \\ -1^o \end{pmatrix}, \begin{pmatrix} 0^o \\ -1^o \end{pmatrix}, \right. \\ \left. \begin{pmatrix} 0^o \\ -1^o \end{pmatrix}, \begin{pmatrix} 0^o \\ 0^o \end{pmatrix}, \begin{pmatrix} 0^o \\ 0^o \end{pmatrix}, \begin{pmatrix} 0^o \\ 0^o \end{pmatrix} \right\}$$

In a second step the doubles/triples (...) of the maps are deleted and we end up with the *flight path* maps.

$$\underline{M} = \{\underline{m}_i \in \underline{\hat{M}} \mid \forall (i,j) \colon \underline{m}_i \neq \underline{m}_j,\, i \neq j\} = \left\{ \begin{pmatrix} -1^o \\ -1^o \end{pmatrix}, \begin{pmatrix} 0^o \\ -1^o \end{pmatrix}, \begin{pmatrix} 0^o \\ 0^o \end{pmatrix} \right\}$$

Since we are now able to choose the correct maps to cover the whole *flight path*, it is lastly important to introduce a function, which reads out the elevation data of a certain map at a certain position located on the map.

Def.: Elevation Data Read Out Function The algorithm uses a function μ that takes a position $\underline{x}_{(GPS)}$ and its corresponding map $\underline{m} = \Phi(\underline{x}_{(GPS)})$ as input and supplies the terrain elevation h_M at $\underline{x}_{(GPS)}$ as output.

$$\mu\left(\underline{x}_{(GPS)}, \Phi(\underline{x}_{(GPS)})\right) = h_M \tag{3.20}$$

Of course, μ only processes *latitude* and *longitude* as position inputs.

3.3 CFIT - Analysis of the Flight Path

In this section, the main part of the algorithm will be discussed. It consists of a loop, that walks through every given aircraft position from *takeoff* to *touchdown* (as defined in one of the last subsections). For all those positions, a special number (the *CFIT - Incident - Number* $\hat{\zeta}$) is calculated as an indicator for the risk, that is caused by the considered flight state, regarding *Controlled Flight Into Terrain*. In the end, when all the indicator values are finally computed, an overall score is built, which we call the *CFIT - Number* ζ. The appealed number will be introduced and defined in the following subsections.

3.3.1 The CFIT - Trajectory

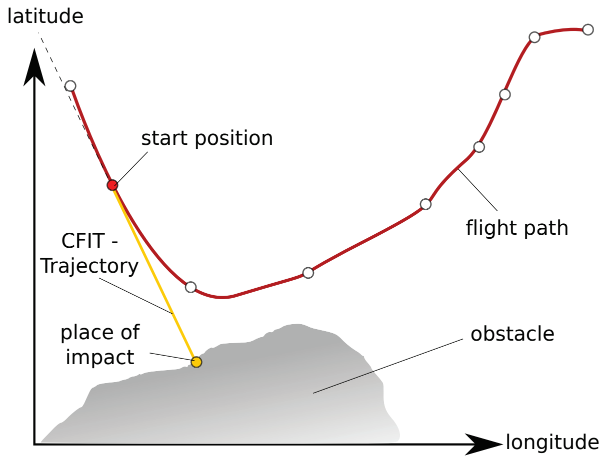
Before giving the mathematical definition, the *CFIT - Trajectory* shall be explained in simple words: What would happen, if a pilot, that flies the aircraft manually, did not correct heading or speed at a certain point of time during the flight. The answer to this question is the tangential straight line drawn to the 3D - curve, which depicts the *flight path*. The aircraft would leave the *flight path* and proceed on this straight line until it hits an obstacle such as terrain, water and so on.

Def.: CFIT - Trajectory Provided that the *speed vector* $\underline{\dot{x}}$ of the aircraft is given at a certain start position $\underline{x}_{(GPS),S}$ the *CFIT - Trajectory* $\underline{\tau}_{(GPS),S}$ can be written as

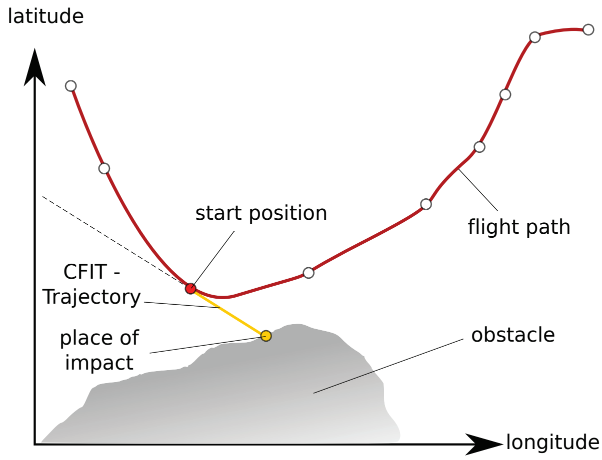
$$\underline{\tau}_{(GPS),S}(t) := \underline{x}_{(GPS),S} + \mathfrak{T}(t \cdot \underline{\dot{x}})\Big|_{\underline{x}_{(GPS),S}} \quad \text{with } S \in [1, N) \subset \mathbb{N} \tag{3.21}$$

Here, t denotes the *time* on the tangential line. S stands for the index at the start position and runs from 1 (*takeoff*) until $N-1$ (one step before *touchdown*). $\mathfrak{T}(\cdot)$ expresses a transformation operator, which is needed to avoid unit inconsistencies and will be explained subsequently after some further definitions.

Def.: CFIT - Position/ Place of (Virtual) Impact/ Time to (Virtual) Impact Any point on the *CFIT - Trajectory* is called *CFIT - Position*. The *CFIT - Positions* range from the



(a) $\underline{x}_{(GPS),S}$.



(b) $\underline{x}_{(GPS),S+1}$.

Figure 3.4: The CFIT - Trajectory beginning from two different starting positions.

recorded position of the aircraft in the *flight path* $\underline{x}_{(GPS),S}$ at $t = 0$ until to the simulated *Place of Impact* $\underline{x}_{(GPS),I}$ at the *Time of/to Impact* $t = t_I$.

$$t \in [0, t_I] \mapsto \underline{\tau}_{(GPS),S}(t) \in \left\{ \underline{x}_{(GPS),S}, \ldots, \underline{x}_{(GPS),I} \right\} \tag{3.22}$$

The *Time to Impact* is the shortest time in which the terrain elevation (given by the map) becomes greater than the actual height of the *CFIT - Trajectory* $h_\tau(t)$.

$$t_I := \underset{t}{\operatorname{argmin}} \{ \mu \left(\underline{\tau}_{(GPS),S}(t), \Phi \left(\underline{\tau}_{(GPS),S}(t) \right) \right) \geq h_\tau(t) \} \tag{3.23}$$

The *Place of Impact* is the position on the *CFIT - Trajectory* given at the *Time of Impact*:

$$\underline{x}_{(GPS),I} := \underline{\tau}_{(GPS),S}(t_I) \tag{3.24}$$

Note that $t = 0$ does not represent *takeoff* - time. The variable t can be interpreted as the curve parameter of the *CFIT - Trajectory*. The introduced terms and the procedure are visualized in Figure 3.4. For convenience, the scenario is reduced to two dimensions and shown in top view. The reader is advised, that the algorithm calculates the *CFIT - Trajectory* and their associated values for every position on the *flight path*.

Let us take a closer look at the units of $\underline{x}_{(GPS),S}$ and $t \cdot \underline{\dot{x}}$:

$$\left[\underline{x}_{(GPS),S} \right] = \left[\begin{pmatrix} x_{N,S} \\ x_{E,S} \\ h_S \end{pmatrix} \right] = \begin{pmatrix} deg \\ deg \\ m \end{pmatrix}$$

$$[t \cdot \underline{\dot{x}}] = \left[\begin{pmatrix} t \cdot V \cos\gamma \cos\psi \\ t \cdot V \cos\gamma \sin\psi \\ t \cdot V \sin\gamma \end{pmatrix} \right] = \begin{pmatrix} s \cdot m/s \\ s \cdot m/s \\ s \cdot m/s \end{pmatrix} = \begin{pmatrix} m \\ m \\ m \end{pmatrix}$$

It should now be obvious that the two expressions can't be summed up without a unit conversion operation. The main task of $\mathfrak{T}(\cdot)$ is converting *meters* to *degrees* in case of *latitude* and *longitude*. For the *altitude*, no transformation is needed.
But how many *meters* are one *degree* of *latitude* and *longitude*? The posed question is relatively easy to answer for the *latitude*. Unfortunately, the nomenclature is a bit confusing at this point. For deriving the conversion parameters it is necessary to "move" on circles where either *latitude* or *longitude* do not change. If we walked on a *circle of longitude*, the *longitude* itself does not change, but we could discover 360^o of *latitude* and vice versa. The *circles of longitude* have the same circumference C_{Long} all around the globe (see Figure 3.5 (a)). It can be calculated with the radius of the earth r_{earth}.

$$C_{Long} = 2r_{earth}\pi \tag{3.25}$$

The *altitude* of the aircraft only slightly influences this value, since $h \ll r_{earth}$. However, for the sake of completeness it is listed in this derivation.

$$C_{Long} = 2(r_{earth} + h)\pi \tag{3.26}$$

One whole *circle of longitude* makes 360^o of *latitude* and the conversion parameter $\sqcup_{Lat}$, that will be multiplied with a value in *meters*, requires the unit deg/m to realize an output in *degrees of latitude*.

$$\sqcup_{Lat} := \frac{360^o}{C_{Long}} = \frac{360^o}{2(r_{earth} + h)\pi} \tag{3.27}$$

In other words: this parameters defines how many *degrees of latitude* equal one meter. It is a bit more challenging to look at the *circles of latitude*, which vary in circumference due to their *latitude* x_N (cf. Figure 3.5 (b)). We define an angle θ, visualized in Figure 3.5 (c).

$$\theta := 90^o - x_N. \tag{3.28}$$

The radius of the *circle of latitude* computes to

$$r_{Lat} = \sin\theta(r_{earth} + h) = \sin(90^o - x_N)(r_{earth} + h). \tag{3.29}$$

So the the conversion operator $\sqcup_{Long}$ can be defined as follows:

$$\sqcup_{Long} := \frac{360^o}{C_{Lat}} = \frac{360^o}{2r_{Lat}\pi} = \frac{360^o}{2\sin(90^o - x_N)(r_{earth} + h)\pi}. \tag{3.30}$$

Thus, $\sqcup_{Lat}$ is a function of the *altitude* and $\sqcup_{Long}$ is a function of the *altitude* and the *latitude* at which the conversion takes place.

$$\sqcup_{Lat} = \sqcup_{Lat}(h)$$
$$\sqcup_{Long} = \sqcup_{Long}(x_N, h)$$

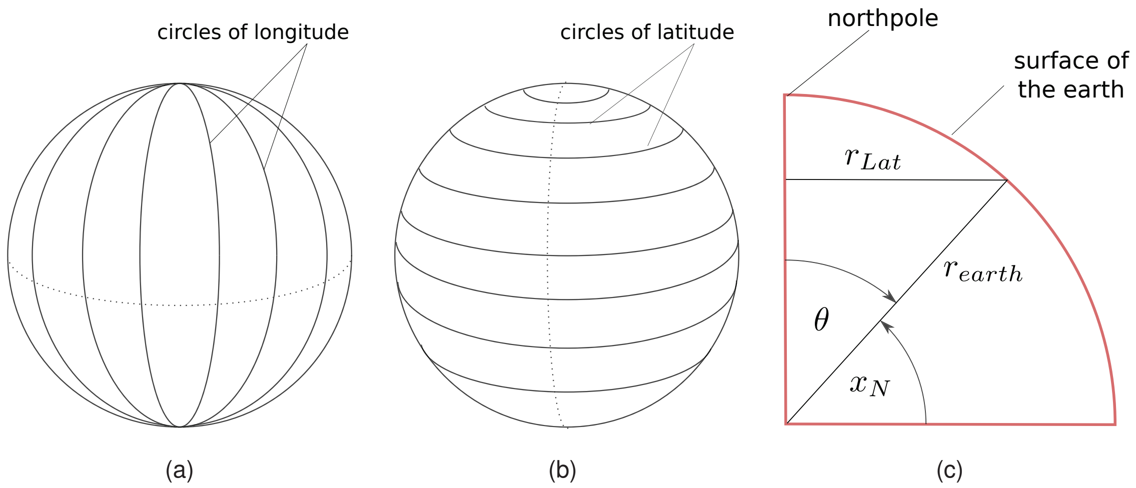


Figure 3.5: Circles of *longitude* (a), circles of *latitude* (b) and θ (c).

Def.: Meters to Degrees Transformation Operator The Meters to Degrees Transformation Operator $\mathfrak{T}(\cdot)$ takes the beginning of the *CFIT - Trajectory* $\underline{x}_{(GPS),S}$ as a reference, so the conversion parameters $\sqcup_{Lat}$ and $\sqcup_{Long}$ stay the same for one tangential line to the *flight path*. This assumption can be safely made, since the *CFIT - Trajectory* is sufficiently short.

$$\mathfrak{T}(t \cdot \underline{\dot{x}})\Big|_{\underline{x}_{(GPS),S}} := \begin{pmatrix} \sqcup_{Lat} & 0 & 0 \\ 0 & \sqcup_{Long} & 0 \\ 0 & 0 & 1 \end{pmatrix}\Bigg|_{\underline{x}_{(GPS),S}} \cdot \underline{\dot{x}}\Big|_{\underline{x}_{(GPS),S}} \cdot t \tag{3.31}$$

$$= \begin{pmatrix} \sqcup_{Lat}(h_S) & 0 & 0 \\ 0 & \sqcup_{Long}(x_{N,S}, h_S) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \underline{\dot{x}}\Big|_{\underline{x}_{(GPS),S}} \cdot t \tag{3.32}$$

3.3.2 The CFIT - Incident

The reader might have a first idea at this point, how the with respect to *Controlled Flight Into Terrain* at every single position in the *flight path* is rated and which tools are applied to achieve this risk assessment. But what is a true risk? A very small *time to impact* is intended during *final approach* and would distort the results of our analysis, when taken into account. It is for this reason, that methods to exclude "false risks" will be introduced in this subsection.

Def.: CFIT - Incident A *CFIT - Incident* is given, if the *CFIT - Trajectory*, starting from $\underline{x}_{(GPS),S}$, hits terrain outside the *runway area* $\underline{\mathcal{R}}$, and if the *time to impact* t_I is smaller than 60 seconds. Only *CFIT - Incidents* are taken into account for the overall risk assessment of the flight.

$$\textit{if}\ t_I \leq 60s\ \wedge\ \underline{\tau}_{(GPS),S}(t_I) = \underline{x}_{(GPS),I} \notin \underline{\mathcal{R}} \Rightarrow \underline{\tau}_{(GPS),S}(t) \in \underline{\oplus} \tag{3.33}$$

The symbol $\underline{\oplus}$ expresses a set, that contains all *CFIT - Trajectories* $\underline{\tau}_{(GPS),S}$ of the investigated flight, which trigger a *CFIT - Incident* according to the above given conditions. The cardinality $|\cdot|$ of $\underline{\oplus}$ is equal to the *Number of CFIT - Incidents* N_{CFIT}, that were detected for the selected flight.

$$\left|\underline{\oplus}\right| =: N_{CFIT} \tag{3.34}$$

Before the *runway area* $\underline{\mathcal{R}}$ is properly defined, its derivation shall be shown first for the sake of comprehensibility. In the practical implementation the airport database of the *Institute for Flight System Dynamics* is called to obtain the *runway reference point* $\underline{RRP}_{(GPS)}$, which marks the point where the runway threshold intersects the runway centerline, the *bearing* β_{RW}, the length of the runway l_{RW} as well as the *field elevation* h_{RW}. All those variables are depicted in Figure 3.6. In a first step the end of the runway $\underline{REP}_{(GPS)}$ needs to be computed. The considerations are based on the theory of analytic geometry, where, a straight line in space is defined by a *receptor point*, which lies on the line itself, and its *direction*. In other words: moving from the beginning of the runway (*receptor point*) to the end of the runway is possible by multiplying the *normalized direction vector* with the length of the runway.

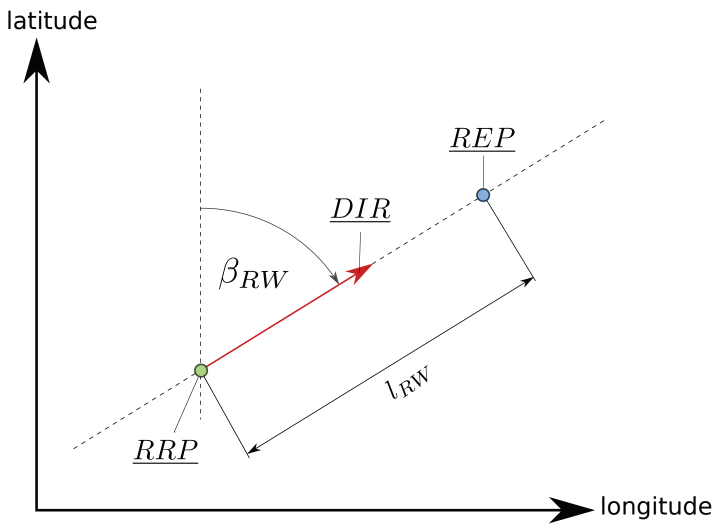
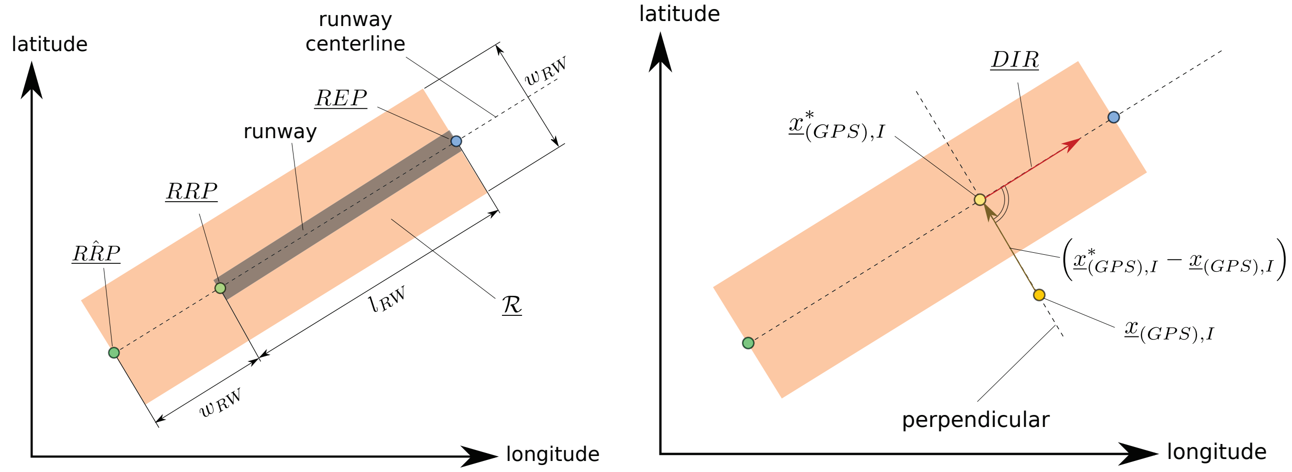


Figure 3.6: Runway parameters.

$$\underline{REP}_{(GPS)} = \underline{RRP}_{(GPS)} + \mathfrak{T}(l_{RW} \cdot \underline{DIR})\Big|_{\underline{RRP}_{(GPS)}} \quad \text{with } \|\underline{DIR}\| = 1 \tag{3.35}$$



(a) The runway area and its characteristic points. (b) The perpendicular drawn from an arbitrary *place of impact*.

Figure 3.7: Important parameters around and in the runway area.

The *Meters to Degrees Transformation Operator* is again needed to convert the added part $l_{RW} \cdot \underline{DIR}$ from meters to degrees. The *direction* can be expressed via the *bearing* of the runway:

$$\underline{DIR} = \begin{pmatrix} \cos(\beta_{RW}) \\ \sin(\beta_{RW}) \end{pmatrix} \text{ with } \|\underline{DIR}\| = \sqrt{\cos^2(\beta_{RW}) + \sin^2(\beta_{RW})} = 1 \tag{3.36}$$

The task of this part of the algorithm is to determine whether a given *place of impact* $\underline{x}_{(GPS),I}$ is inside the *runway area* $\underline{\mathcal{R}}$ or not. To increase the tolerance during *final approach* the width of the runway is enlarged to $w_{RW} = 400\ m$ and the *runway reference point* is shifted $400\ m = w_{RW}$ in $-\ \underline{DIR}$ direction (cf. Figure 3.7 (a)).

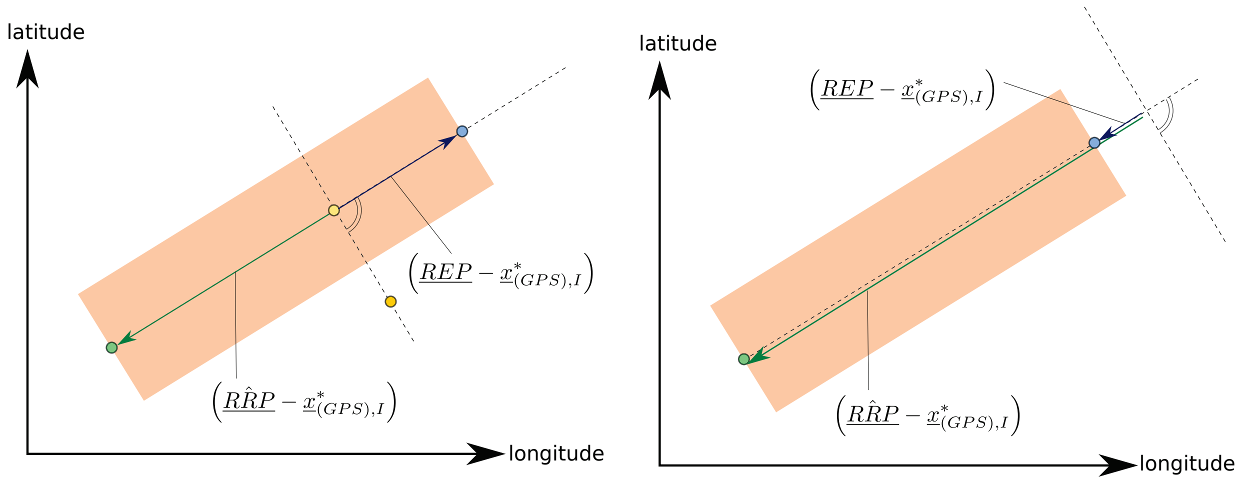
$$\underline{R\hat{R}P}_{(GPS)} = \underline{RRP}_{(GPS)} - \mathfrak{T}(w_{RW} \cdot \underline{DIR})\Big|_{\underline{RRP}_{(GPS)}} \tag{3.37}$$

This rectangular territory marks the *runway area*. Presumed that there is *place of impact* $\underline{x}_{(GPS),I}$ of which the affiliation to $\underline{\mathcal{R}}$ is unclear, then in the first step a perpendicular is drawn from $\underline{x}_{(GPS),I}$ to $\underline{x}^*_{(GPS),I}$, which lies on the (potentially elongated) *runway center line* (as depicted in Figure 3.7 (b)). If it lies on the centerline, it can be expressed in the same way as the end of the runway, but with a still unknown length shift λ^* from the *runway reference point* analogue to equation (3.35).

$$\underline{x}^*_{(GPS),I} = \underline{RRP}_{(GPS)} + \mathfrak{T}(\lambda^* \cdot \underline{DIR})\Big|_{\underline{RRP}_{(GPS)}} \tag{3.38}$$

The following orthogonal condition must become valid in order to compute the unknown point $\underline{x}^*_{(GPS),I}$:

$$\left\langle (\underline{x}^*_{(GPS),I} - \underline{x}_{(GPS),I}), \underline{DIR} \right\rangle \stackrel{!}{=} 0. \tag{3.39}$$



(a) Condition "Projection" is true.

(b) Condition "Projection" is false.

Figure 3.8: Condition "Projection".

The angle brackets $\langle\cdot\rangle$ indicate the *vectorial scalar product* in between a vector pointing from the *place of impact* to the orthogonally projected point on the runway center line and the direction vector of the runway. Combining equations (3.38) and (3.39) leads to:

$$\left\langle (\underline{RRP}_{(GPS)} + \mathfrak{T}(\lambda^* \cdot \underline{DIR})\Big|_{\underline{RRP}_{(GPS)}} - \underline{x}_{(GPS),I}), \underline{DIR} \right\rangle \overset{!}{=} 0. \tag{3.40}$$

Without introducing new symbols the third coordinate *barometric altitude* will not be considered in the following. $\underline{RRP}_{(GPS)} =: (a_N, a_E)^T$

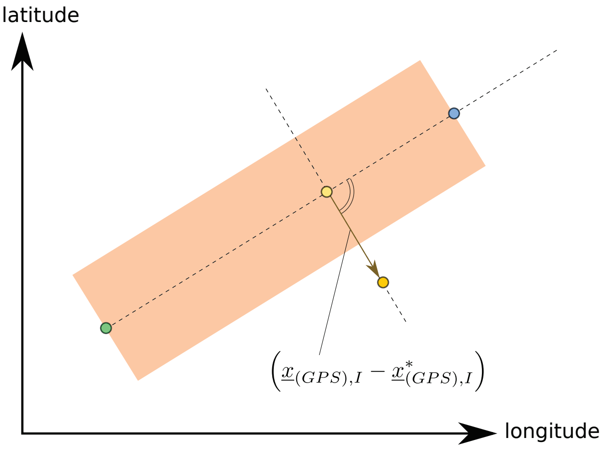
$$\left\langle \left(\left(\begin{array}{c} a_N \\ a_E \end{array} \right) + \lambda^* \left(\begin{array}{cc} \sqcup_{Lat} & 0 \\ 0 & \sqcup_{Long} \end{array} \right) \Bigg|_{\underline{RRP}_{(GPS)}} \left(\begin{array}{c} \cos(\beta_{RW}) \\ \sin(\beta_{RW}) \end{array} \right) - \underline{x}_{(GPS),I} \right), \underline{DIR} \right\rangle \overset{!}{=} 0. \tag{3.41}$$

$$\left\langle \left(\begin{array}{c} a_N + \lambda^* \sqcup_{Lat}(h_{RW}) \cos(\beta_{RW}) - x_{N,I} \\ a_E + \lambda^* \sqcup_{Long}(h_{RW}, a_N) \sin(\beta_{RW}) - x_{E,I} \end{array} \right), \left(\begin{array}{c} \cos(\beta_{RW}) \\ \sin(\beta_{RW}) \end{array} \right) \right\rangle \overset{!}{=} 0. \tag{3.42}$$

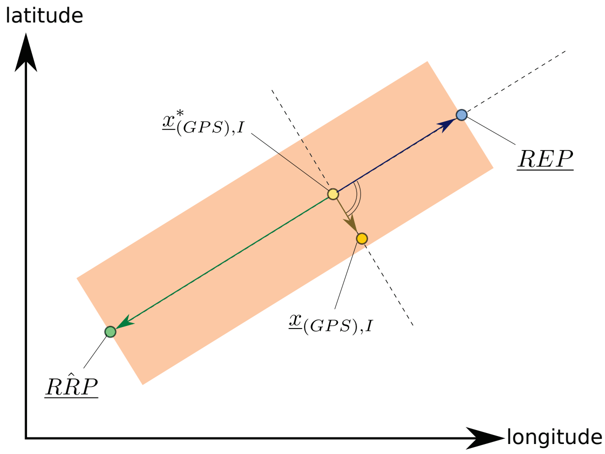
After transposing the equation λ^* can be specified as stated below.

$$\lambda^* = \frac{(x_{E,I} - a_E)\sin(\beta_{RW}) + (x_{N,I} - a_N)\cos(\beta_{RW})}{\sqcup_{Lat}(h_{RW})\cos^2(\beta_{RW}) + \sqcup_{Long}(h_{RW}, a_N)\sin^2(\beta_{RW})} \tag{3.43}$$

Since λ^* is now known, the orthogonally projected *place of impact* on the runway center line is known as well. Indeed, there are several ways to show, that the *place of impact* lies inside $\underline{\mathcal{R}}$ or not. Here, the method, that is implemented in the algorithm shall be explained. Two things need to be checked:



(a) Condition "Distance to Center Line" is false.



(b) Both conditions are true.

Figure 3.9: Condition "Distance to Center Line".

Condition "Projection" First, if the orthogonally projected *place of impact* $\underline{x}^*_{(GPS),I}$ lies in between the shifted *runway reference point* $\underline{R\hat{R}P}_{(GPS)}$ and the end of the runway $\underline{REP}_{(GPS)}$. This is true, if the distances to both of these markers are smaller than the tolerance runway length of $l_{RW} + w_{RW}$. In Figure 3.8 one can see that <u>both</u> of the following statements need to become valid in order to proof, that the projected *place of impact* lies inside the *runway area*.

$$\left\| \mathfrak{T}\left(\underline{R\hat{R}P}_{(GPS)} - \underline{x}^*_{(GPS),I}\right)\Big|_{\underline{RRP}_{(GPS)}} \right\| \leq l_{RW} + w_{RW} \tag{3.44}$$

$$\left\| \mathfrak{T}\left(\underline{REP}_{(GPS)} - \underline{x}^*_{(GPS),I}\right)\Big|_{\underline{RRP}_{(GPS)}} \right\| \leq l_{RW} + w_{RW} \tag{3.45}$$

Condition "Distance to Center Line" Second, if the vector from the orthogonally projected *place of impact* to the original *place of impact* itself is smaller than half of the tolerance width w_{RW}.

$$\left\| \mathfrak{T}\left(\underline{x}_{(GPS),I} - \underline{x}^*_{(GPS),I}\right)\Big|_{\underline{RRP}_{(GPS)}} \right\| \leq \frac{1}{2} w_{RW} \tag{3.46}$$

Figure 3.9 (a) visualizes a *place of impact*, which fulfills the first condition, but which is too far away from the center line. In 3.9 (b) both conditions become true, what makes the point part of the *runway area*.

Def.: Runway Area A place of impact $\underline{x}_{(GPS),I}$ is an element of the *runway area* $\underline{\mathcal{R}}$, if it fulfills equations (3.44) - (3.46).

3.3.3 The Point of no Return

The reader is now informed which *CFIT - Trajectories* contribute to the overall risk assessment, but there is still an outstanding question: Which of the computed values should be taken into account for the risk evaluation? In the scope of this thesis, the decision was made, that the risk is set to 100% at a point, where it is impossible for the pilot to avoid the imminent accident, if he flew along the *CFIT - Trajectory*. This point is defined as the *Point of no Return* (short: *PONR*). By those considerations, a few statements can already be made.

The Idea The time to the *Point of no Return* t_{PONR} is usually a few seconds smaller than the *time to impact*.

$$t_{PONR} < t_I \text{ thus } t_{PONR} \in (0, t_I) \tag{3.47}$$

It depends on its corresponding *CFIT - Trajectory* and of course the terrain elevation of the surrounding environment.

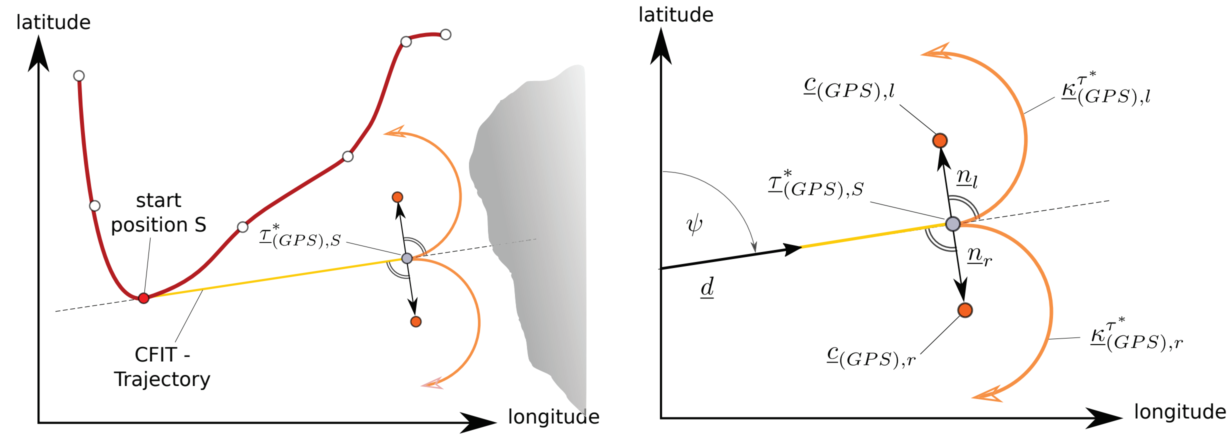
$$t_{PONR} = f\left(\underline{\mathcal{T}}_{(GPS),S}(t), \Phi(\underline{\mathcal{T}}_{(GPS),S}(t_I))\right) \tag{3.48}$$

Recall, that the formalism Φ provides the corresponding maps to a *flight path*, which is in this case the *CFIT - Trajectory*. But why does the terrain elevation of the surrounding environment influence the *PONR*? Imagine flying in flat terrain without mountains or even hills. No matter how you leave the *flight path*, it will always be possible to escape an accident by pulling the machine upwards, provided that there is enough space left underneath the aircraft. Whereas in mountainous terrain it might be impossible to avoid a crash even though there is enough distance to the ground left. Due to a limited maximal *flight path angle* flying out of a deep valley could not be possible anymore.

The Evasive Maneuvers The idea can be summed up in other words: The algorithm should check on every point of the *CFIT - Trajectory*, if any evasive maneuver is still possible to avoid the imminent crash. In case the accident is unavoidable the *Point of no Return* is reached. Within the framework of the implemented program three evasive maneuvers were considered.

- symmetric *pull - up*,
- horizontal 180^o - curve to the left,
- horizontal 180^o - curve to the right.

Obviously one could think of an infinity of combinations of those three named options, such as an asymmetric *pull - up*, but since the listed maneuvers are the elementary ones, the analysis has been restricted to these cases.



(a) Depiction of the horizontal evasion curves marked in orange.

(b) Zoom into (a).

Figure 3.10: Horizontal evasion curves.

The Horizontal Curves During this maneuver the aircraft is strictly kept on the same height, where the calculations started from. Of course, laws of physics forbid, that an aircraft changes from descent directly to a horizontal curve maneuver without losing any further height. This issue will be discussed later on, because the loss of *altitude* is modeled differently. For the moment we consider one point $\underline{\tau}^*_{(GPS),S}$ on the *CFIT - Trajectory*:

$$\underline{\tau}^*_{(GPS),S} := \underline{\tau}_{(GPS),S}(t^*) = \begin{pmatrix} x_{N,\tau*} \\ x_{E,\tau*} \\ h_{\tau*} \end{pmatrix} \text{ with } t^* \in [0, t_I] \tag{3.49}$$

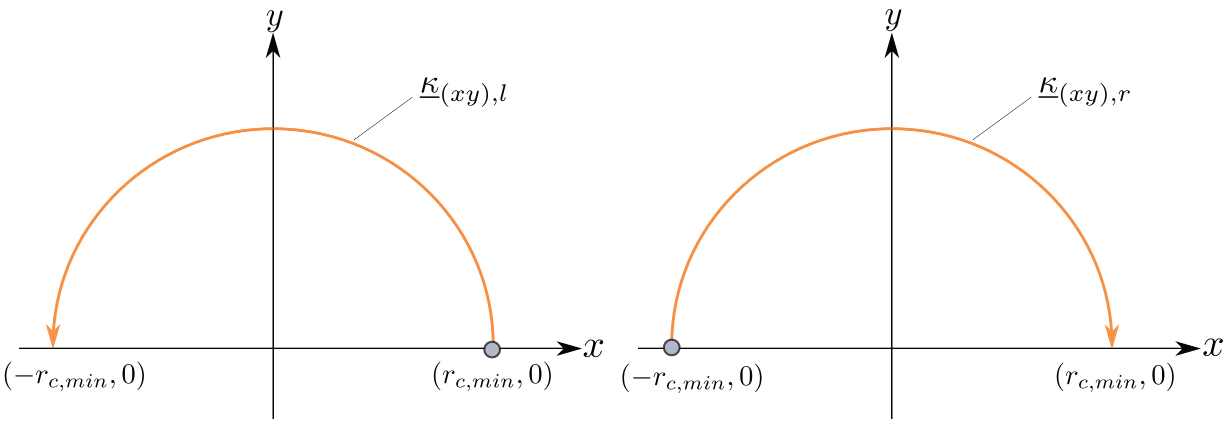
In the following the third coordinate $h_{\tau*}$ will be omitted, since - as mentioned - the aircraft is kept on the same height from the beginning of the maneuver on. The situation is described in Figure 3.10 (a). The first important variable, that needs to be computed is the *minimal possible curve radius* $r_{c,min}$.

$$r_{c,min} = \frac{V_H^2}{g\sqrt{n^2 - 1}} \text{ [9]} \tag{3.50}$$

The *horizontal speed* is expressed by V_H, whereas n characterizes the *load factor*, an aircraft specific structural constant, that was set to $n = 2.5$ (cf. page A-48 in [8]). As usual, g is the *gravity*. In the next step, we would like to find the center of the left evasion curve $\underline{c}_{(GPS),l}$ as well as the center of the right evasion curve $\underline{c}_{(GPS),r}$. The idea how to find them, is the same as described in the last section according to the theory of analytic geometry. They can be written as follows:

$$\underline{c}_{(GPS),r} = \underline{\tau}^*_{(GPS),S} + \mathfrak{T}(r_{c,min} \cdot \underline{n}_r)\Big|_{\underline{\tau}^*_{(GPS),S}}, \tag{3.51}$$

$$\underline{c}_{(GPS),l} = \underline{\tau}^*_{(GPS),S} + \mathfrak{T}(r_{c,min} \cdot \underline{n}_l)\Big|_{\underline{\tau}^*_{(GPS),S}}. \tag{3.52}$$



(a) $(xy)-$curve to the left.

(b) $(xy)-$curve to the right.

Figure 3.11: Curves in an $(xy)-$coordinate system.

$\underline{n}_r$ and $\underline{n}_l$ are the directions towards the curve centers. They stand perpendicular on the horizontal direction vector of the *CFIT - Trajectory* $\underline{d}$ (cf. Figure 3.10 (b)) and must therefore fulfill the following conditions:

$$\langle \underline{n}_r, \underline{d} \rangle \overset{!}{=} 0 \text{ with } \|\underline{n}_r\| \overset{!}{=} 1 \tag{3.53}$$

$$\langle \underline{n}_l, \underline{d} \rangle \overset{!}{=} 0 \text{ with } \|\underline{n}_l\| \overset{!}{=} 1 \tag{3.54}$$

One can express the horizontal direction vector of the *CFIT - Trajectory* $\underline{d}$ with the help of the *track angle* ψ:

$$\underline{d} = \begin{pmatrix} \cos(\psi) \\ \sin(\psi) \end{pmatrix} \tag{3.55}$$

It is easy to proof that the following vectors meet the above given specifications.

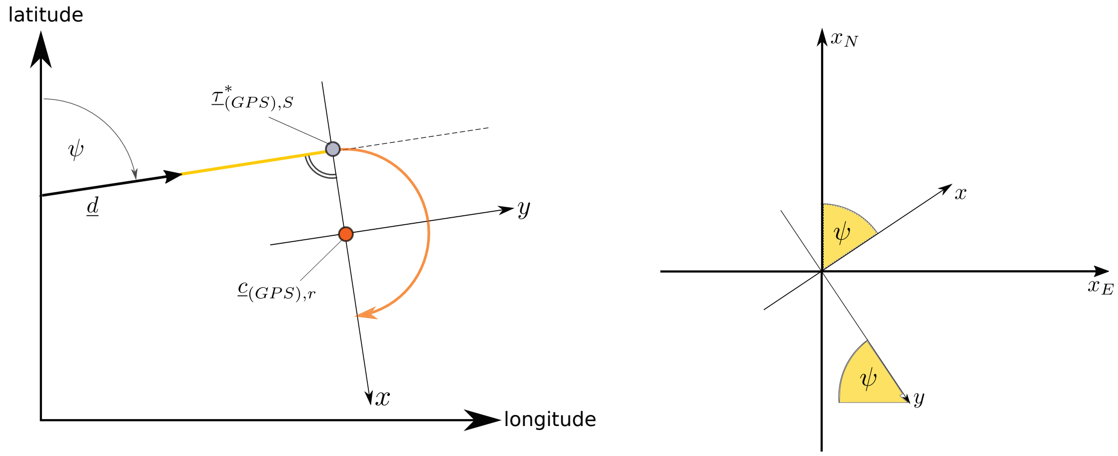
$$\underline{n}_r = \begin{pmatrix} -\sin(\psi) \\ \cos(\psi) \end{pmatrix}, \ \underline{n}_l = \begin{pmatrix} \sin(\psi) \\ -\cos(\psi) \end{pmatrix} \tag{3.56}$$

Thus with (3.51), (3.52), (3.50) and (3.56), the centers of the evasion curves can be assumed as known, but the description of the curve itself is still missing. Therefore, we start with a simple semicircle in an $(xy)-$coordinate system, which starts at $(r_{c,min},0)^T$ and comes to an end at $(-r_{c,min},0)^T$ (cf. Figure 3.11 (a)).

$$\underline{\mathcal{K}}_{(xy),l} : \omega \in [0,\pi] \to \mathbb{R}^2, \ \omega \mapsto \underline{\mathcal{K}}_{(xy),l}(\omega) := r_{c,min} \begin{pmatrix} \cos(\omega) \\ \sin(\omega) \end{pmatrix}_{(xy)} \tag{3.57}$$

For a reverse of the *traverse direction*, thus a start at $(-r_{c,min},0)^T$ (cf. Figure 3.11 (b)), only a small modification is necessary:

$$\underline{\mathcal{K}}_{(xy),r} : \omega \in [0,\pi] \to \mathbb{R}^2, \ \omega \mapsto \underline{\mathcal{K}}_{(xy),r}(\omega) := r_{c,min} \begin{pmatrix} \cos(\pi-\omega) \\ \sin(\pi-\omega) \end{pmatrix}_{(xy)} . \tag{3.58}$$



(a) Exemplary positioning of the $(xy)-$ coordinate system in the inertial frame.

(b) Twisting angle ψ of the $(xy)-$ coordinate system (not in proportion to the left sketch).

Figure 3.12: Locating the $(xy)-$coordinate system in the inertial frame.

Take the right evasion curve for example and take a look at Figure 3.12 (a): If we turn the $(xy)-$coordinate system in a way, that the $x-$axis aligns with the $\underline{n}_r$ direction and the center of the $(xy)-$coordinate system is equal to the right center of the curve $\underline{c}_{(GPS),r}$, then $\underline{\kappa}_{(xy),r}$ describes the right 180^o evasion curve. Therefore, the final goal is to obtain a formulation for

$$\underline{\kappa}^{\mathcal{T}^*}_{(GPS),r} \text{ and } \underline{\kappa}^{\mathcal{T}^*}_{(GPS),l}, \tag{3.59}$$

which express the evasion curves to the right and left beginning at a certain point $\underline{\mathcal{T}}^*_{(GPS),S}$. The values from the turned $(xy)-$coordinate system have to be transformed back to the $(GPS)-$frame. From Figure 3.12 (b) one can see that

$$x_N = y \cdot \cos(\psi) - x \cdot \sin(\psi) \text{ and} \tag{3.60}$$

$$x_E = y \cdot \sin(\psi) + x \cdot \cos(\psi). \tag{3.61}$$

These two lines are better summed up in a rotation matrix

$$\underline{\underline{R}}_\psi = \begin{pmatrix} -\sin(\psi) & \cos(\psi) \\ \cos(\psi) & \sin(\psi) \end{pmatrix}. \tag{3.62}$$

Vectors $\underline{x}_{(xy)}$, that are given in an $(xy)-$coordinate system, which is turned at ψ compared to the $(GPS)-$ frame, can be expressed in the inertial system:

$$\underline{x}_{(GPS)} = \underline{\underline{R}}_\psi \underline{x}_{(xy)} \tag{3.63}$$

This leads to the final result:

$$\underline{\kappa}^{\mathcal{T}^*}_{(GPS),r}(\omega) = \mathfrak{T}\Big(\underline{\underline{R}}_\psi \underline{\kappa}_{(xy),r}(\omega)\Big)\Big|_{\underline{\mathcal{T}}^*_{(GPS),S}} + \underline{c}_{(GPS),r} \tag{3.64}$$

$$\underline{\kappa}^{\mathcal{T}^*}_{(GPS),l}(\omega) = \mathfrak{T}\Big(\underline{\underline{R}}_\psi \underline{\kappa}_{(xy),l}(\omega)\Big)\Big|_{\underline{\mathcal{T}}^*_{(GPS),S}} + \underline{c}_{(GPS),l} \tag{3.65}$$

By $\underline{\underline{R}}_\psi$ the curve positions $\underline{\kappa}_{(xy),r}$ or $\underline{\kappa}_{(xy),l}$ are expressed aligned to the inertial coordinate system (GPS) (*rotation*) and transformed to degrees from meters by $\mathfrak{T}(\cdot)$. They need to be shifted by the position of the corresponding curve center afterwards (*translation*).

Def.: Blocked Evasion Curve If the terrain height (map height) $h_M = \mu(\cdot, \Phi(\cdot))$ detected on only one of the curve positions, is greater than the (constant) curve height $h_{\tau*}$, the evasion curve counts as blocked.

$$\exists\omega \in [0,\pi] : \mu(\underline{\kappa}^{\tau^*}_{(GPS),r}, \Phi(\underline{\kappa}^{\tau^*}_{(GPS),r})) \geq h_{\tau*} \Rightarrow \underline{\kappa}^{\tau^*}_{(GPS),r} = \textit{blocked} \tag{3.66}$$

$$\exists\omega \in [0,\pi] : \mu(\underline{\kappa}^{\tau^*}_{(GPS),l}, \Phi(\underline{\kappa}^{\tau^*}_{(GPS),l})) \geq h_{\tau*} \Rightarrow \underline{\kappa}^{\tau^*}_{(GPS),l} = \textit{blocked} \tag{3.67}$$

Symmetric Pull - Up In case the *bank angle* is zero and the aircraft is pulled up, it is spoken of a *symmetric pull - up*. In the framework of this thesis a few simplifying assumption were made.

- The aircraft reaches the lowest point of the *flare - out* 4 seconds after the beginning of the maneuver following the given glide path.
- The *maximal possible flight path angle* γ_{max} is applied directly after reaching the lowest point and set to the numerical solution of the following equation:

$$\sin(\gamma_{max}) + \epsilon_{max}\cos(\gamma_{max}) - \frac{T_{max}}{mg} = 0 \text{ [9]} \tag{3.68}$$

When the algorithm is applied to different aircraft types, reasonable estimations for T_{max} and ϵ_{max} can be found publically available (e.g. in the aircraft information material published by the manufacturer). For test calculations we considered an *Airbus A320*: ϵ_{max} was set to $\frac{1}{18}$ and the maximal thrust to $T_{max} = 220\,kN$, whereas the aircraft's mass m is read out of the *FDM* Data (e.g. for $m = 50\,t \Rightarrow \gamma_{max} = 23^o$). Further elaborations of an aerodynamic model were beyond the scope of this work. Please see the *Summary and Outlook* section for enhanced ideas on this topic. We jump back to the point under investigation on the *CFIT - Trajectory*, where the evasion possibilities should be checked from. It is recalled that this point is defined as:

$$\underline{\tau}^*_{(GPS),S} := \underline{\tau}_{(GPS),S}(t^*) = \begin{pmatrix} x_{N,\tau*} \\ x_{E,\tau*} \\ h_{\tau*} \end{pmatrix} \text{ with } t^* \in [0, t_I]$$

As defined above the lowest point of the *flare - out* is 4 seconds away from the point under investigation following the given *glide path*. A *flare - out* curve ϕ is introduced which starts at this lowest point and describes the subsequent climb with the *maximal possible flight path angle* γ_{max} (cf. Figure 3.13). Thus, the lowest point $\underline{\phi}^{\tau^*}_{(GPS),0}$ is given by

$$\underline{\phi}^{\tau^*}_{(GPS),0} := \underline{\tau}_{(GPS),S}(t^* + 4s) \tag{3.69}$$

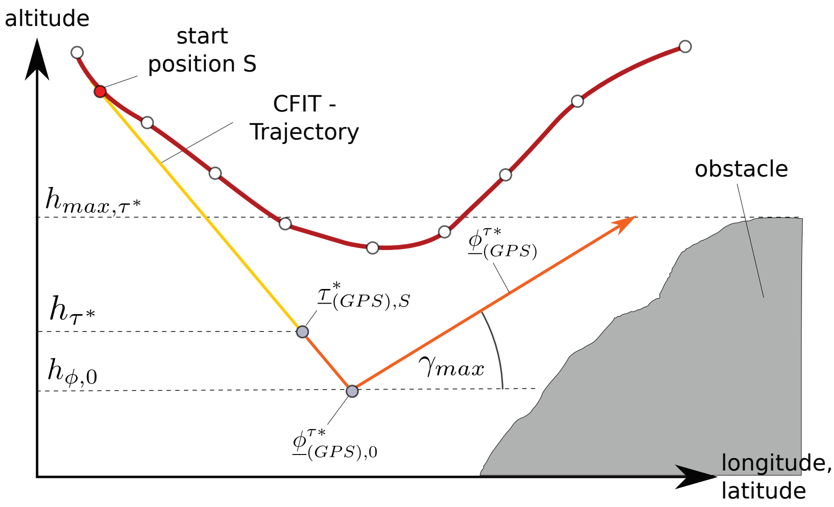


Figure 3.13: The parameters important to the *symmetric pull - up* depicted in *side view*.

In other words, following the *CFIT - Trajectory*, that reflects the given *glide path*, for 4 more seconds. Since the *bank angle* is zero the *track angle* ψ of the *CFIT - Trajectory* does not change. With this information the *flare - out* curve ϕ is now definable:

$$\underline{\phi}^{\tau^*}_{(GPS)}(\xi) := \underline{\phi}^{\tau^*}_{(GPS),0} + \mathfrak{T}(\xi \cdot \underline{V}_\phi)\Big|_{\underline{\tau}^*_{(GPS),S}} \quad \text{with } \xi \in \mathbb{R}_{\geq 0} \tag{3.70}$$

The parameter ξ represents climb time starting from the bottom, whereas the *flare - out speed* and direction is defined by $\underline{V}_\phi$:

$$\underline{V}_\phi := V \begin{pmatrix} \cos\gamma_{max}\cos\psi \\ \cos\gamma_{max}\sin\psi \\ \sin\gamma_{max} \end{pmatrix}. \tag{3.71}$$

Again, ψ is equal to the *track angle* of $\underline{\tau}_{(GPS),S}$ and V is taken from $\underline{\tau}_{(GPS),S}$ as well. We assumed, that the climb is complete and the aircraft is safe, if an *altitude* is reached which is bigger than the maximum terrain height on the map of $\underline{\tau}^*_{(GPS),S}$ and all of its adjacent maps $\underline{M}_{adj,\tau^*}$. The map, on which $\underline{\tau}^*_{(GPS),S}$ lies, is computed by the formalism Φ:

$$\underline{m}_{\tau^*} := \begin{pmatrix} m_{N,\tau^*} \\ m_{E,\tau^*} \end{pmatrix} = \Phi(\underline{\tau}^*_{(GPS),S}). \tag{3.72}$$

There are eight adjacent maps in total plus one original map, which can be seen in Figure 3.14. All of the relevant maps are summed up in the area $\underline{M}_{adj,\tau^*}$.

$$\underline{M}_{adj,\tau^*} = \left\{ \begin{pmatrix} m_{N,\tau^*} + i \\ m_{E,\tau^*} + j \end{pmatrix} \right\}_{i,j\in[-1,1]} \tag{3.73}$$

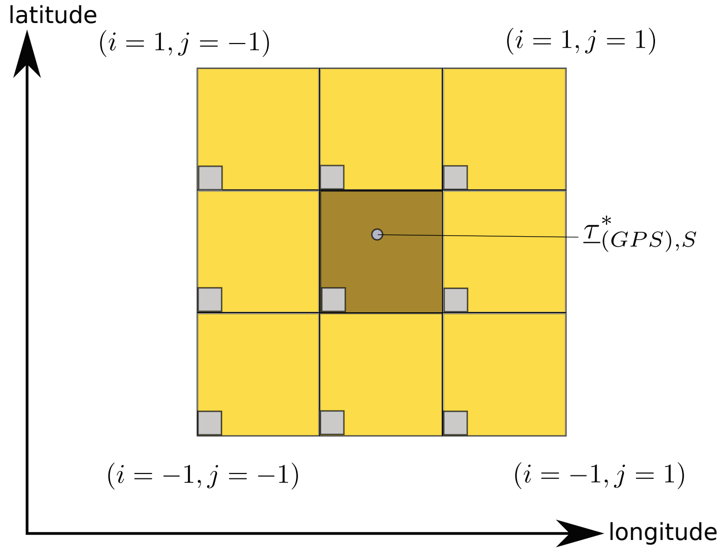


Figure 3.14: The adjacent maps surrounding the point under investigation.

The maximum terrain *altitude* h_{max,τ^*}, that surrounds the point under investigation $\underline{\tau}^*_{(GPS),S}$ is extracted with the help of μ (p. 22) as follows:

$$h_{max,\tau^*} = \max_{\forall \underline{x}_{(GPS)} \in \underline{M}_{adj,\tau^*}} \left\{ \mu\left(\underline{x}_{(GPS)}, \Phi(\underline{x}_{(GPS)})\right) \right\} \tag{3.74}$$

The *flare - out* curve has three coordinates, all dependent of ξ:

$$\underline{\phi}^{\tau^*}_{(GPS)}(\xi) = \begin{pmatrix} x_{N,\phi}(\xi) \\ x_{E,\phi}(\xi) \\ h_{\phi}(\xi) \end{pmatrix} \tag{3.75}$$

Def.: Blocked Symmetric Pull - Up If the *altitude* of the *flare - out curve* h_ϕ is still below the maximum terrain *altitude* h_{max,τ^*} and there is a point on the curve, for which the terrain height (map height) $h_M = \mu(\cdot, \Phi(\cdot))$ is bigger or equal than h_ϕ, the *symmetric pull - up* counts as blocked.

$$h_\phi(\xi) < h_{max,\tau^*} : \exists \xi : \mu\left(\underline{\phi}^{\tau^*}_{(GPS)}, \Phi(\underline{\phi}^{\tau^*}_{(GPS)})\right) \geq h_\phi(\xi) \Rightarrow \underline{\phi}^{\tau^*}_{(GPS)} = \textit{blocked} \tag{3.76}$$

As mentioned before it is problematic that the horizontal curves are simulated without initial loss of *altitude*. If we the define the *Point of no Return*, as the point, where all three evasion possibilities are blocked at the same time, approaching completely flat terrain would never provide a *Point of no Return* before the *place of impact*. Evading to the left and right would always be possible until the aircraft virtually crashes, which can be seen in Figure 3.15. That is the reason for the following definition.

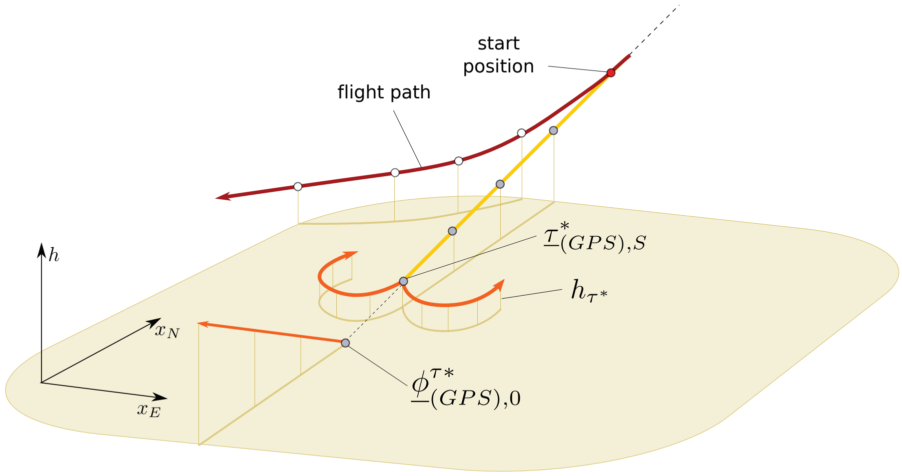


Figure 3.15: 3D - visualization of the *Point of no Return* approaching flat terrain.

Def.: Blocked Flare - Out Bottom If the terrain height (map height) $h_M = \mu(\cdot, \Phi(\cdot))$ detected on the position of the *lowest point* of the *flare - out*, is greater than the height $h_{\phi,0}$ of this point, the *flare - out bottom* counts as blocked.

$$\mu\left(\underline{\phi}^{\tau^*}_{(GPS),0}, \Phi(\underline{\phi}^{\tau^*}_{(GPS),0})\right) \geq h_{\phi,0} \Rightarrow \underline{\phi}^{\tau^*}_{(GPS),0} = \textit{ blocked} \tag{3.77}$$

Def.: Point of no Return The *Point of no Return* is reached at $\underline{\tau}^*_{(GPS),S}$, if the *flare - out bottom* is *blocked* according to equation (3.77). In case the *flare - out bottom* is clear, the *Point of no Return* is reached if the *symmetric pull - up* and both *horizontal evasion curves* are blocked according to equations (3.76) and (3.66)-(3.67). Therefore, t^* is called t_{PONR} and $\underline{\tau}^*_{(GPS),S}$ is the *Place/Point of no Return*.

Based on this definition and according to all considerations of the last subsections the function Π extracts the time to the *Point of no Return* t_{PONR} out of the *CFIT - Trajectory* $\underline{\tau}_{(GPS),S}(t)$, which started from sample S on the original *flight path*:

$$t_{PONR,S} =: \Pi(\underline{\tau}_{(GPS),S}(t)). \tag{3.78}$$

3.3.4 The CFIT - Number

The result of the last subsection, is the last required part to finally assess the risk of the flight with regard to *Controlled Flight Into Terrain*. The closer the original *flight path* is located to a *Point of no Return*, the higher the risk. Obviously, the risk equals 100% if $t_{PONR} = 0s$, but what about the other end? How far away from the computed *Point of no Return* is the risk set to 0%? The decision was made to link a t_{PONR} of $60s$ with a risk of 0%, since the *time to impact* must be smaller than $60s$, otherwise the trajectory does not count as an *CFIT - Incident* (cf. equation (3.33)). The following function covers those characteristics:

$$\hat{\zeta} := \frac{1}{\Delta^2}\left(\Delta - t_{PONR}\right)^2 \text{ with } \Delta = 60s = const. \tag{3.79}$$

Indeed, a linear function would meet the above given conditions as well. It is for this reason that we take a look at the absolute value $|\cdot|$ of the derivative of $\hat{\zeta}$:

$$\left|\frac{\partial \hat{\zeta}}{\partial t_{PONR}}\right| = \left| -\frac{2}{\Delta^2}\left(\Delta - t_{PONR}\right)\right| = 2\left|\frac{1}{\Delta} - \frac{t_{PONR}}{\Delta^2}\right| \tag{3.80}$$

Not only the risk $\hat{\zeta}$ increases if $t_{PONR} \rightarrow 0s$, but also the rate of change of $\hat{\zeta}$ increases. Due to the increasing psychological pressure on the cockpit crew, when approaching terrain, and the thereby caused worse flight and reaction performance, these assumptions might be satisfied.

Def.: CFIT - Incident - Number $\hat{\zeta}_S$ describes the *CFIT - Incident - Number* $\in [0,1)$, which belongs to a *CFIT - Trajectory*, that started from sample S on the *flight path*:

$$\hat{\zeta}_S = \hat{\zeta}_S(\underline{\tau}_{(GPS),S}) := \begin{cases} \frac{1}{\Delta^2}\left(\Delta - \Pi(\underline{\tau}_{(GPS),S})\right)^2 & , \forall \underline{\tau}_{(GPS),S} \in \underline{\oplus} \\ 0 & , \forall \underline{\tau}_{(GPS),S} \notin \underline{\oplus} \end{cases} \tag{3.81}$$

In case $\underline{\tau}_{(GPS),S}$ does not meet the conditions of a *CFIT - Incident* ($\notin \underline{\oplus}$), then the risk is set to 0.

Def.: CFIT - Number ζ describes the *CFIT - Number* $\in [0,1)$, which belongs to the whole flight and rates the risk of the selected flight in regards to *Controlled Flight Into Terrain.*

$$\zeta := \frac{1}{N-1}\sum_{S=1}^{N-1} \hat{\zeta}_S \tag{3.82}$$

Recall that N is defined as the number of samples in the *flight path* from *takeoff* to *touchdown*.

3.4 Final Approach Distribution Analysis

The introduced algorithm gives further possibilities in terms of a statistical *final approach* analysis. We state, that there is an optimal glide path for *final approach*, which consists of a straight line pointing in *bearing* direction towards the runway, which hits the touchdown zone at an angle of γ_{opt} (cf. Figure 3.16). γ_{opt} is the *angle of descent* given by the *Instrument Landing System* (*ILS*). Including the *reference speed* V_{ref} for *final approach*, the *optimal time to touchdown* t_{opt} is given by

$$t_{opt}(h) := \frac{h - h_{RW}}{V_{ref} \cdot \sin\gamma_{opt}} \tag{3.83}$$

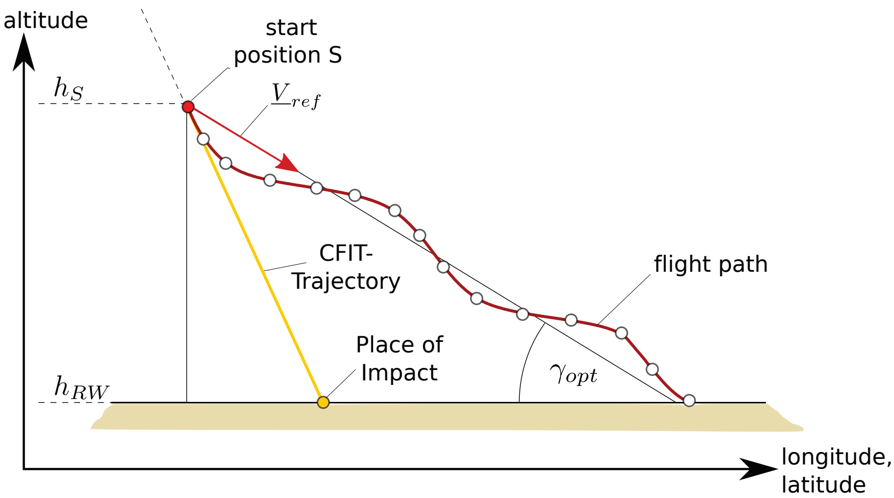


Figure 3.16: An optimal approach compared to a real descent.

The elevation of the runway h_{RW} has to be subtracted from the *barometric altitude* h of the aircraft (assuming that the barometric altimeter reference setting is correctly set to QNH). As discussed in the last chapters, we already have the *time to impact* $t_{I,S}$ available, for every position S on the *flight path*, especially for all samples of *final approach*.

$$t_{I,S} := \underset{t}{\text{argmin}}\{\mu\left(\underline{\tau}_{(GPS),S}(t), \Phi\left(\underline{\tau}_{(GPS),S}(t)\right)\right) \geq h_\tau(t)\}$$

In case the approach is *optimal* in sample S with respect to *speed* and *angle of descent* the following expression should amount to 0:

$$\delta_S := t_{I,S} - t_{opt,S} \text{ with } t_{opt,S} = \frac{h_S - h_{RW}}{V_{ref} \cdot \sin \gamma_{opt}} \tag{3.84}$$

where h_S depicts the aircraft's *altitude* at position $\underline{x}_{(GPS),S}$. Thus, δ_S is among other things a function of the position of the aircraft, of which only those are taken into account, that belong to the set of *final approach* samples $\underline{\mathcal{F}}$.

$$\delta_S = \delta_S(\underline{x}_{(GPS),S}) = t_{I,S} - t_{opt,S}, \forall \underline{x}_{(GPS),S} \in \underline{\mathcal{F}} \tag{3.85}$$

$\underline{\mathcal{F}}$ is a subset of the whole *flight path*:

$$\underline{\mathcal{F}} := \left\{\underline{x}_{(GPS),S} \in \underline{X}_{(GPS)} \middle| h_S - h_{RW} \leq 300m \wedge \gamma_S < 0\right\} \tag{3.86}$$

It usually starts, when the vertical distance to the runway drops below $300\ m$ during descent. We consider the *mean value* of all δ_S as well as the *standard deviation*.

$$\bar{\delta} := \frac{1}{|\underline{\mathcal{F}}|} \sum_{S=N-|\underline{\mathcal{F}}|}^{N-1} \delta_S(\underline{x}_{(GPS),S}), \forall \underline{x}_{(GPS),S} \in \underline{\mathcal{F}} \tag{3.87}$$

The cardinality of $\underline{\mathcal{F}}$ states how many samples belong to the *final approach*. Thus the *sample index* S starts from $N - |\underline{\mathcal{F}}|$ and ends shortly before touchdown at $N - 1$. The variance computes to

$$\sigma_\delta^2 = \frac{1}{|\underline{\mathcal{F}}| - 1} \sum_{S=N-|\underline{\mathcal{F}}|}^{N-1} \left(\delta_S - \bar{\delta}\right)^2, \forall \underline{x}_{(GPS),S} \in \underline{\mathcal{F}}, \tag{3.88}$$

with its corresponding *standard deviation*

$$\sigma_\delta = \sqrt{\sigma_\delta^2} = \sqrt{\frac{1}{|\underline{\mathcal{F}}| - 1} \sum_{S=N-|\underline{\mathcal{F}}|}^{N-1} \left(\delta_S - \bar{\delta}\right)^2}, \forall \underline{x}_{(GPS),S} \in \underline{\mathcal{F}}. \tag{3.89}$$

The pair of values

$$\left(\bar{\delta}, \sigma_\delta\right)$$

describes the quality of the approach statistically. In case the pilot did not follow the glide slope correctly (for example with a constant angle offset), the *mean value* is expected to be shifted away from zero. Whereas during an unstable approach where *flight path angle* and speed frequently change, the standard deviation must rise. Please see the *Results and Discussion* section for the application of this analysis procedure to real world flight data.

3.5 Program Options

In the beginning of the source code, the user has a few options to choose from by setting of boolean variables, of which the most important ones shall be explained in this section.

3.5.1 Trajectory Smoothing

The sensors of the aircraft deliver the *flight path angle* γ and the *track angle* ψ as a "snapshot" of the momentary position. Due to measurement inaccuracies, it is possible that therefore the tangential lines to the *flight path* do not reflect the actual direction angles anymore. It might be useful in some cases to switch on the *Trajectory Smoothing*, which calculates the mentioned angles itself by considering the positions

$$\underline{x}_{(GPS),S} \text{ and } \underline{x}_{(GPS),S+\Delta S} \text{ with } \Delta S = 5 = const. \tag{3.90}$$

Figure 3.17 shows the height difference as well as the difference of *latitude* and *longitude*:

$$\Delta h := h_{S+\Delta S} - h_S,\ \Delta x_N = x_{N,S+\Delta S} - x_{N,S} \text{ and } \Delta x_E = x_{E,S+\Delta S} - x_{E,S}. \tag{3.91}$$

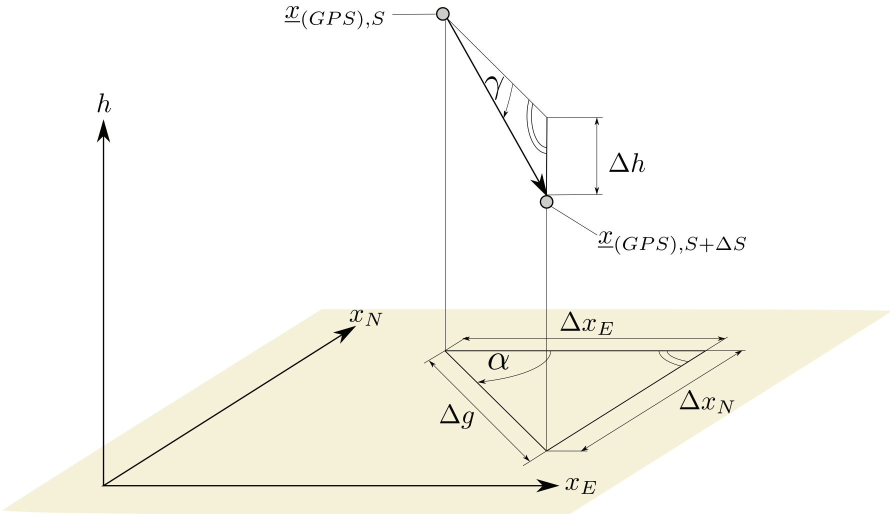


Figure 3.17: Two recorded aircraft positions together with their affiliated distances and angles.

The variable Δg describes the length of the vector from $\underline{x}_{(GPS),S}$ to $\underline{x}_{(GPS),S+\Delta S}$, projected to the surface of the earth. The marked angles α and γ can be expressed by trigonometric relations:

$$\tan \gamma = \frac{\Delta h}{\Delta g} \tag{3.92}$$

In contrast to Δh, which has the unit meter, one can see that Δx_N is given in degree. The conversion parameter $\sqcup_{Lat}(h)$ helps to transform degree to meter, when it is inverted. The reference height for the projection is set to the *altitude* of the runway.

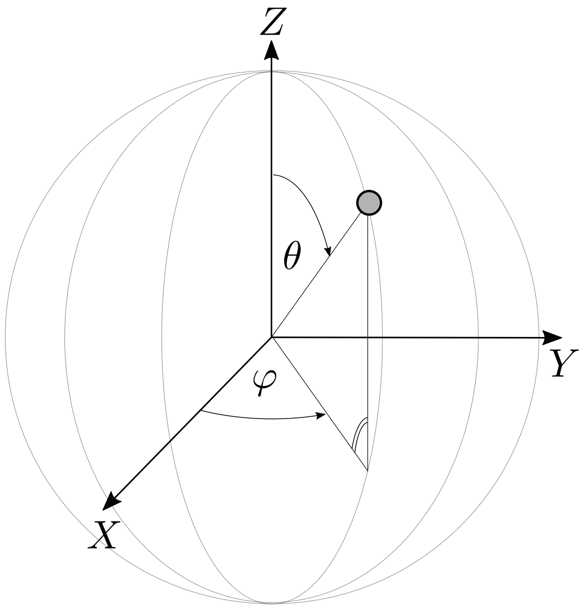
$$\sin \alpha = \frac{\Delta x_N}{\Delta g \cdot \sqcup_{Lat}(h_{RW})} \tag{3.93}$$

In order to proceed, the derivation of Δg will be necessary.
Latitude and *longitude* express the position of a point on the surface of the earth. In a first approximation, the earth can be seen as a sphere, and therefore positions are expressible in *spherical coordinates*. As depicted in Figure 3.18 (a), the angle φ is introduced together with the already described θ inside of an $(XYZ)-$coordinate system, that has its origin in the middle of the earth's core. Reminder:

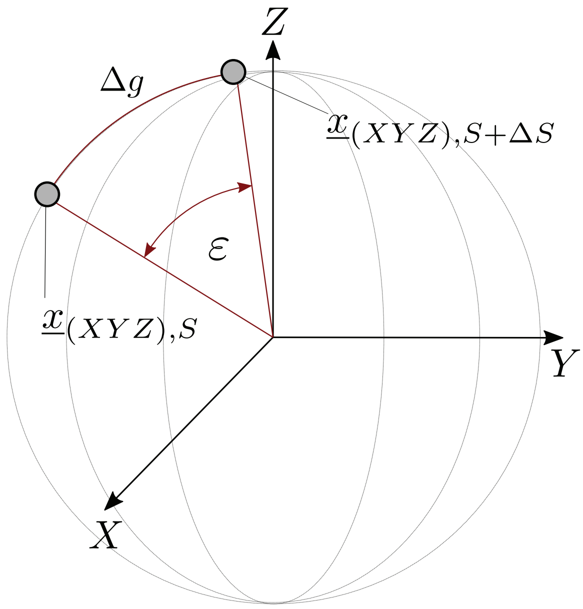
$$\theta_S = 90^o - x_{N,S}.$$

The angle φ_S, which belongs to position S, is computed with the *longitude*:

$$\varphi_S := \begin{cases} x_{E,S} & , x_{E,S} \geq 0^o \\ 360^o + x_{E,S} & , x_{E,S} < 0^o \end{cases} \tag{3.94}$$



(a) Placement of the angles θ and φ.



(b) Angle ε in between two aircraft positions.

Figure 3.18: The $(XYZ)-$coordinate system.

We express the given positions $\underline{x}_{(GPS),S}$ and $\underline{x}_{(GPS),S+\Delta S}$ within the cartesian $(XYZ)-$coordinate system:

$$\underline{x}_{(XYZ),S} = r_{earth} \begin{pmatrix} \sin(\theta_S)\cos(\varphi_S) \\ \sin(\theta_S)\sin(\varphi_S) \\ \cos(\theta_S) \end{pmatrix} \tag{3.95}$$

$$\underline{x}_{(XYZ),S+\Delta S} = r_{earth} \begin{pmatrix} \sin(\theta_{S+\Delta S})\cos(\varphi_{S+\Delta S}) \\ \sin(\theta_{S+\Delta S})\sin(\varphi_{S+\Delta S}) \\ \cos(\theta_{S+\Delta S}) \end{pmatrix} \tag{3.96}$$

This enables the possibility of computing the angle ε that lies in between them (confer Figure 3.18 (b)).

$$\varepsilon = \arccos\left(\frac{\left\langle \underline{x}_{(XYZ),S}, \underline{x}_{(XYZ),S+\Delta S}\right\rangle}{\|\underline{x}_{(XYZ),S}\| \cdot \|\underline{x}_{(XYZ),S+\Delta S}\|}\right) \tag{3.97}$$

The *orthodrome* in between the points under investigation is a part of the circumference of the whole earth and given by

$$\Delta g = 2\pi(r_{earth} + h_{rw})\frac{\varepsilon}{2\pi} = \varepsilon \cdot (r_{earth} + h_{RW}) \tag{3.98}$$

The base level for the projection has been set to the elevation of the runway, therefore $(r_{earth} + h_{RW})$. This formula will only work, if ε used in rad. With equations (3.92), (3.93), and (3.98) the angles γ (*flight path angle*) and α are known. In the final step α

is transformed to the track angle ψ, since the $\arcsin(\cdot)-$function delivers values only in between $[-90^o, 90^o]$.

$$\psi = \begin{cases} 90^o - \alpha & , \Delta x_E > 0^o \\ 270^o + \alpha & , \Delta x_E < 0^o \end{cases} \tag{3.99}$$

3.5.2 Height Cut Off

Indeed, the main task of this algorithm will be to analyze flights which come close to terrain, or in other words which are executed in mountainous terrain. It is well known, that the aircraft is not even close to any obstacle, in most cases during cruise. For this reason the user of the program has the possibility to enable a program option which deletes the points in the *flight path* above a certain *altitude*. This special *altitude* is the maximum terrain height on the maps , which cover the *flight path*, plus an extra 30%. Recall: Remember that initially the *flight path* $\underline{X}_{(GPS)}$ is given. The formalism Φ extracts the maps, which cover the whole path:

$$\underline{M} = \Phi(\underline{X}_{(GPS)})$$

In the next step, the maximum terrain height of all maps $h_{max,\underline{M}}$ is calculated:

$$h_{max,\underline{M}} := \max_{\forall \underline{x}_{(GPS)} \in \underline{M}} \left\{ \mu \left(\underline{x}_{(GPS)}, \Phi(\underline{x}_{(GPS)} \right) \right\} \tag{3.100}$$

The part of the *flight path* that is subsequently analyzed by the algorithm is simply said a subset of $\underline{X}_{(GPS)}$:

$$\underline{X}_{(GPS),cut} := \left\{ \underline{x}_{(GPS),S} \in \underline{X}_{(GPS)} \middle| h_S \leq h_{max,\underline{M}} \cdot 1.3 \right\} \tag{3.101}$$

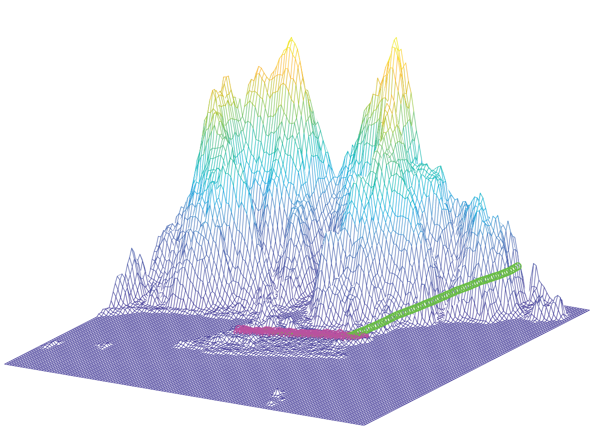
3.5.3 Consider Only Final Approach

If the user wishes to consider only the samples that belong to *final approach*, the already defined set $\underline{\mathcal{F}}$ is solely taken into account of the analysis:

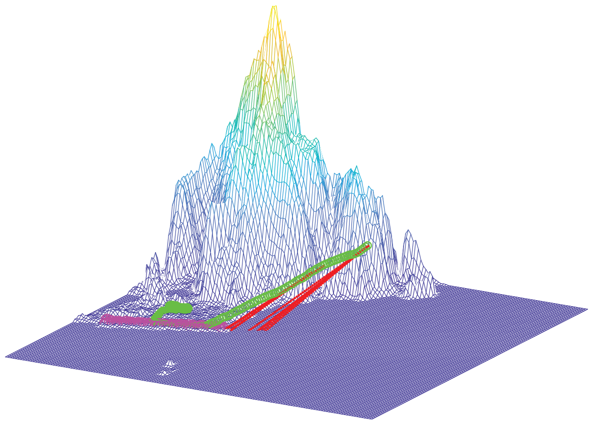
$$\underline{\mathcal{F}} = \left\{ \underline{x}_{(GPS),S} \in \underline{X}_{(GPS)} \middle| h_S - h_{RW} \leq 300m \wedge \gamma_S < 0 \right\}$$

4 Results and Discussion

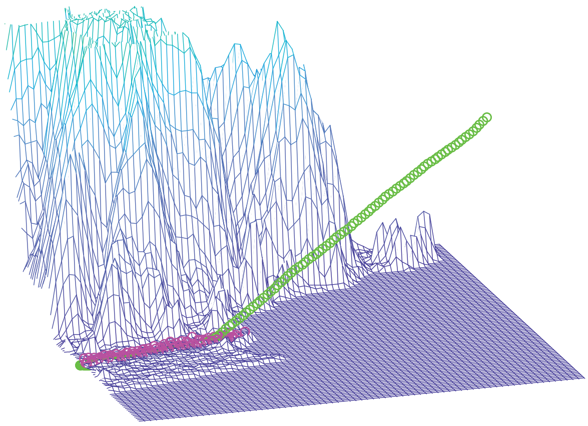
The described algorithm was applied to real flight data. All results have been created by *considering only the final approach* and without the *Trajectory Smoothing Option* (which means that *flight path angle* and *track angle* are directly read out of the *FDM* Data).



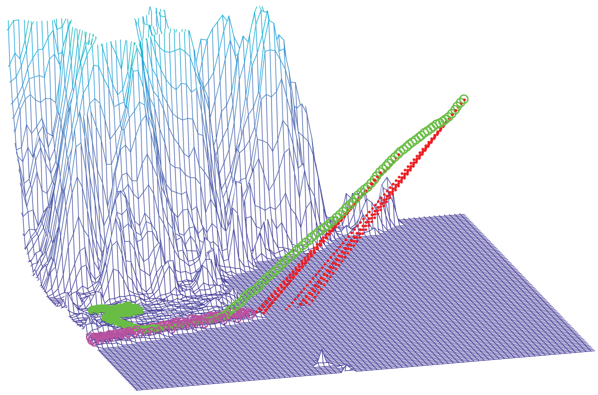
(a) Flight A.



(b) Flight B.



(c) Flight A (zoom).



(d) Flight B (zoom).

Figure 4.1: Exemplary flights to [redacted].

[redacted] In Figure 4.1 two exemplary flights to [redacted] are depicted. On the left side, one can see the final approach of Flight A, whereas Flight B is shown on the right. The pictures on top display an

Flight	N_{CFIT} $(-)$	ζ $(-)$	$\bar{\delta}$ (s)	σ_{δ} (s)
A	0	0.0	0.0	2.28
B	10	0.0121	-5.25	8.46
∅([redacted])	2.18	0.0016	-1.66	4.2

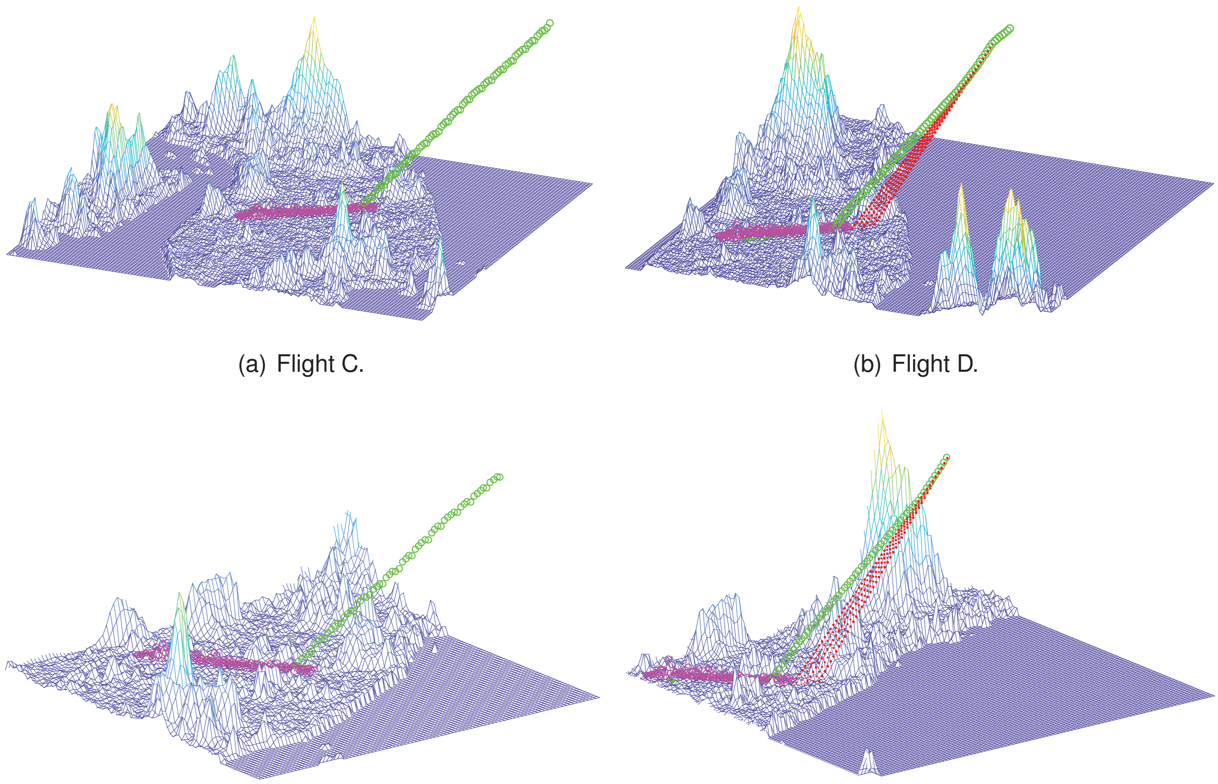
Table 4.1: Exemplary flights to [redacted].

overview, and are zoomed in below. They are *MATLAB* figures, that are produced and stored at the end of the calculation and depict the *flight path* $\underline{X}_{(GPS)}$ (green), the *runway area* $\underline{\mathcal{R}}$ (purple) and all the detected *CFIT - Incidents* in form of the according *CFIT - Trajectory* $\underline{\mathcal{T}}_{(GPS),S}$ on the topographic environment of *ASTER GDEM*. Together with the flight number, four more values are stored after processing the data: the number of *CFIT - Incidents* N_{CFIT}, the *CFIT - Number* of the approach or flight ζ, and the pair of values according to the *Final Approach Distribution Analysis* $\left(\bar{\delta}, \sigma_\delta\right)$. In table 4.1 the described numbers for both flights and the airport's average (a mean value of 23 flights) are written down. Considering only the referred parameters, the approach of Flight A is in a positive sense above average: Since there was no *CFIT - Incident* detected, of course ζ computed to 0. Furthermore it seems, that the pilot followed the ILS glidepath almost perfectly, because the mean value $\bar{\delta}$, which describes the deviations from a perfect approach, is 0. Its very small standard deviation implies, that there were no major corrections necessary during final approach. The landing, shown on the right side of Figure 4.1, is far below the airport's average: 10 *CFIT - Incidents* were computed and account to a *CFIT - Number* of 0.0121, which comes from the tangential lines (red), that do not hit the *runway area* and impact ground in less than 60 seconds. In 4.1 (d), the reason becomes obvious: There is a bend visible, in the approach path, which most likely appears, because of a necessary course correction during an ILS landing maneuver. This bend gravely influences the pair of values $\left(\bar{\delta}, \sigma_\delta\right)$: the mean value $\bar{\delta}$ is shifted negatively and can be interpreted as an approach steeper than the optimum, whereas σ_δ indicates strong fluctuations around the optimal approach path.

Flight	$N_{CFIT}\ (-)$	$\zeta\ (-)$	$\bar{\delta}\ (s)$	$\sigma_\delta\ (s)$
C	0.0	0.0	-0.55	2.22
D	13	0.0139	-3.68	6.40
∅([redacted])	3.83	0.0072	-3.79	3.8

Table 4.2: Exemplary flights to [redacted].

[redacted] Figure 4.2 displays two approaches to [redacted] Here, Flight C is the positive example, whereas Flight D is in a negative sense far below average. The numbers are printed in table 4.2 together with the airport's average values computed from six landings. With zero *CFIT - Incidents* and a $\bar{\delta}$ of -0.55, as well as a very low σ_δ, the landing maneuver of C exhibits no abnormalities at all and seems to be perfectly executed. The approach of Flight D is way too steep in the beginning, such that 13 *CFIT - Trajectories* impact terrain in front of the runway. Note, that due to the high sampling rate not all 13 tangential straight lines are visible in 4.2 (d), because some of them are located underneath the others. The steepness is expressed in the negative shift of $\bar{\delta}$ and the glidepath corrections amount to a σ_δ of 6.40.



(a) Flight C. (b) Flight D.

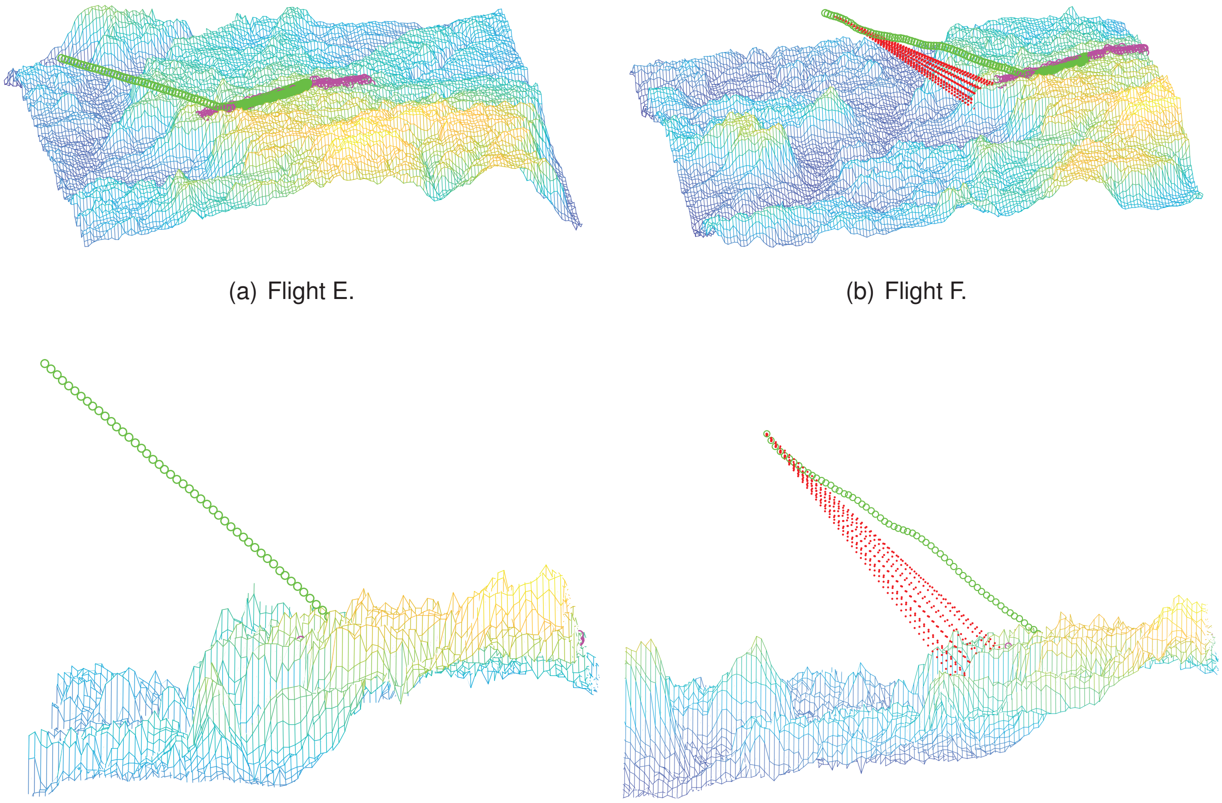
(c) Flight C (zoom). (d) Flight D (zoom).

Figure 4.2: Exemplary flights to [redacted]

[redacted] Again, Figure 4.3 contains a positive on the left and a negative example on the right side. The results are displayed in table 4.3 together with the airport mean values of 67 approaches. The two flights have been chosen, in order to prove, that a low mean value $\bar{\delta}$ doesn't necessarily coincide with a small number of *CFIT - Incidents*. With a $\bar{\delta}$ of -1.7 Flight F produced 11 *CFIT - Incidents*, which becomes clear when observing Figure 4.3 (d). The glidepath is stepwise oscillating from a too steep approach to a flattened one, which leads to a quite inconspicuous mean value. Only by the help of the standard deviation σ_δ of the *CFIT - Distribution - Analysis* the obtained N_{CFIT} becomes explainable.

Flight	N_{CFIT} $(-)$	ζ $(-)$	$\bar{\delta}$ (s)	σ_δ (s)
E	0.0	0.0	0.163	2.44
F	11	0.0116	-1.7	9.57
∅([redacted]	3.28	0.0068	2.14	4.66

Table 4.3: Exemplary flights to [redacted].



(a) Flight E.

(b) Flight F.

(c) Flight E (zoom).

(d) Flight F (zoom).

Figure 4.3: Exemplary flights to [redacted].

Further Comparisons If the problems, that will be pointed out in the next chapter, are solved, further comparisons will be possible in the future. The following enumeration shall give a few ideas about what will be comparable based on their average *CFIT - Numbers*:

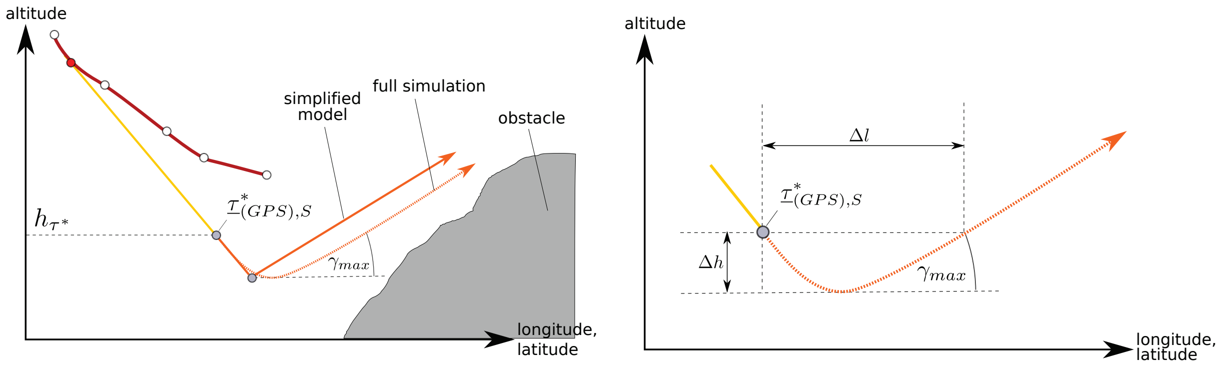
- flights,
- aircraft types,
- pilots,
- runways,
- airports,
- approach maneuvers,
- areas [redacted],
- weather conditions at the same airport,
- airlines and much more.

5 Outlook and Summary

5.1 Outlook

5.1.1 An Enhanced Pull - Up Model

As mentioned in section 3.3.3 a further elaborated *pull - up* simulation model has been developed in the scope of this thesis (in close cooperation with *Phillip Koppitz*). Only the previously described simplified version is implemented in the main routine, since there are still uncertainties in the reliability of the advanced model, which will be presented in the following paragraphs. The effects of wind are neglected. Figure 5.1



(a) The more sophisticated model compared to the simplified approach.

(b) Zoom into (a).

Figure 5.1: The more sophisticated *pull - up* model.

compares the simplified (implemented) approach to a fully simulated *flare - out*. The aim of the simulation would be to compute the three in 5.1 (b) displayed parameters Δh, Δl and γ_{max}. The aircraft travels a distance of Δl above ground and loses Δh of its initial *altitude*, until it finally ascends with a constant *flight path angle* of γ_{max} in the end of the maneuver.

Input Values At the point under investigation, $\mathcal{T}^*_{(GPS),S}$, where the evasion possibilities shall be checked from, we obtain the following initial information:

- Initial *height* of the *flare - out*: $h_{\mathcal{T}^*}$.
- Initial *speed* of the *flare - out*: $V_{\mathcal{T}^*}$.
- Initial *flight path angle* of the *flare - out* $\gamma_{\mathcal{T}^*}$.
- *Flap setting* (taken from start point S on the original flight path).
- *Gear setting* (taken from start point S on the original flight path).

Constant	Unit	Value
S	m^2	105
g	$m \cdot s^{-2}$	9.80665
T_0	K	288
R	$J \cdot K^{-1} mol^{-1}$	8.314
ρ_0	$kg \cdot m^{-3}$	1.293
T_{ref}	kN	180
h_{ref}	m	0
n	$-$	1.235
n_V	$-$	-0.25
n_ρ	$-$	0.75
$n_{z,max}$	$-$	2.5

Table 5.1: Table of constants

Aerodynamic Parameters The *flap* and *gear* settings are processed by the *FSD Aerodynamic Model* depicted in Figure 5.2. where k represents a resistance factor, C_D stands for *drag coefficient* and C_L for *lift coefficient*. The model delivers an explicit model function f_M, which computes C_D with the help of C_L. For data protection reasons, we are not allowed to explicitly print f_M here.

Nature and Aircraft Constants Table 5.1 defines the used nature constants and refers to a *Boeing 737*. By S the wing area is defined, T_0 describes the ground temperature, R stands for the *gas constant* and ρ_0 depicts the air density at mean sea level. At a height of h_{ref} the aircraft has a thrust of T_{ref} available and reaches a speed of V_{ref}. The maximum load factor $n_{z,max}$ indicates the structural limits of the aircraft, when pulled up. n_ρ and n are exponents related to gas dynamics and n_V is set to the value of a turbofan engine.

Figure 5.2: Input and output parameters of the FSD model.

Initial Conditions of the Simulation With the help of the above introduced parameters and values, the initial conditions for the simulation can be defined. Note, that they are indicated by the index (I).

$$h_{(I)} = h_{\tau^*} \tag{5.1}$$

$$V_{(I)} = V_{\tau^*} \tag{5.2}$$

$$\gamma_{(I)} = \gamma_{\tau^*} \tag{5.3}$$

$$l_{(I)} = 0\ m \tag{5.4}$$

$$\rho_{(I)} = \rho_0 \cdot \left(1 - \frac{n-1}{n} \cdot \frac{g}{RT_0} \cdot h_{(I)}\right)^{\frac{1}{n-1}} \tag{5.5}$$

$$C_{L,(I)} = \frac{2mg\cos\gamma_{(I)}}{\rho_{(I)}V_{(I)}^2 S} \tag{5.6}$$

$$C_{D,(I)} = f_M(C_{L,(I)}) \tag{5.7}$$

$$D_{(I)} = T_{(I)} = \frac{1}{2}\rho_{(I)} \cdot S \cdot C_{D,(I)}V_{(I)}^2 \tag{5.8}$$

$$n_{z,(I)} = \begin{cases} \frac{\rho_{(I)} \cdot S \cdot C_{L,max} V_{(I)}^2}{2mg} & , C_{L,(I)} > C_{L,max} \\ n_{z,max} & , else \end{cases} \tag{5.9}$$

By l the distance above ground traveled from the beginning of the maneuver is indicated and set to $0\ m$ in the beginning. Except for that, we let the thrust be equal to the drag when we start the simulation. n_Z needs to be fitted according to the structural limits of the aircraft.

Thrust Characteristic It can be expected, that the pilot would apply the maximum available thrust, immediately after pulling the machine upwards, for a quick climb. In a first approximation we assumed the thrust characteristic to be a straight line, which rises the engine power from the initial thrust, to the maximum thrust available at the starting height of the maneuver.

$$T_{max} = T_{ref} \cdot \left(\frac{\rho_{(I)}}{\rho_{ref}}\right)^{n_\rho} \text{ with } \rho_{ref} := \rho(h_{ref}) \tag{5.10}$$

$$T(t) = T_{(I)} + t \cdot \frac{T_{max} - T_{(I)}}{\Delta t_T} \text{ with } \Delta t_T = 3\ s = const \tag{5.11}$$

Numerical Evaluation of γ_{max} The following equation needs to be solved numerically in order to obtain V_γ

$$f(V_\gamma) = n_V \cdot \frac{L_p V^{*n_V}}{mg} \cdot \left(\frac{V_\gamma}{V^*}\right)^{n_V - 1} + \epsilon^* \left(\left(\frac{V^*}{V_\gamma}\right)^3 - \left(\frac{V_\gamma}{V^*}\right)\right) = 0 \tag{5.12}$$

$$\text{with } L_p = \frac{T_{max}}{V^{*n_V}}, \tag{5.13}$$

$$C_L^* = \sqrt{\frac{C_{D,0}}{k} + C_{L,0}^2}, \tag{5.14}$$

$$V^* = \sqrt{\frac{2mg}{C_L^* \rho_{(I)} S}}, \tag{5.15}$$

$$C_D^* = C_{D,0} + k \cdot (C_L^* - C_{L,0})^2, \tag{5.16}$$

$$\epsilon^* = \sqrt{\frac{C_D^*}{C_L^*}} \tag{5.17}$$

Afterwards γ_{max} is given by:

$$\gamma_{max} = \arcsin\left(\frac{\epsilon^*}{2n_V}\left((2 - n_V) \cdot \left(\frac{V_\gamma}{V^*}\right)^2 - (2 + n_V) \cdot \left(\frac{V^*}{V_\gamma}\right)^2\right)\right) \tag{5.18}$$

Secondary Conditions During the whole simulation (and of course in the beginning) the following auxiliary conditions are iteratively checked in order to correct the related parameters if necessary.

$$n_z(t) = \begin{cases} \frac{\rho(t) \cdot S \cdot C_{L,max} V^2(t)}{2mg} & , C_L(t) > C_{L,max} \Rightarrow C_L(t) \stackrel{\triangle}{=} C_{L,max} \\ n_{z,max} & , else \Rightarrow C_L(t) \stackrel{\triangle}{=} \frac{2n_{z,max} mg}{\rho(t) V^2(t) S} \end{cases} \tag{5.19}$$

$$\gamma(t) = \begin{cases} \gamma_{max} & , \gamma(t) > \gamma_{max} \Rightarrow n_{z,max} \stackrel{\triangle}{=} \cos(\gamma_{max}) \\ \gamma(t) & , else \end{cases} \tag{5.20}$$

$$T(t) = \begin{cases} T_{max} & , T(t) > T_{max} \\ T(t) & , else \end{cases} \tag{5.21}$$

The triangle $\triangle$ above the equal sign means *"set to"*. It is shown, that by n_z the whole climb maneuver is controlled: As long as the lift factor C_L is not out of its physical limits given by the *FSD Model*, the load factor is set to its maximum, which produces a quick change of the climb angle of the simulated aircraft. Of course, *flight path angle* and thrust need to stay inside their physical limits as well.

Main Equations of the Simulation The following set of differential and algebraic equations form the main part of the simulation. Together with the *auxiliary conditions* they are iteratively evaluated.

$$\dot{V}(t) = \frac{T(t) - D(t)}{m} - g \sin\gamma(t) \tag{5.22}$$

$$\dot{\gamma}(t) = \frac{\rho(t)V(t)C_L(t)S}{2m} - \frac{g}{V(t)}\cos\gamma(t) \tag{5.23}$$

$$\dot{h}(t) = V(t)\sin\gamma(t) \tag{5.24}$$

$$\dot{l}(t) = V(t)\cos\gamma(t) \tag{5.25}$$

$$\rho(t) = \rho_0 \cdot \left(1 - \frac{n-1}{n} \cdot \frac{g}{RT_0} \cdot h(t)\right)^{\frac{1}{n-1}} \tag{5.26}$$

$$C_L(t) = \frac{2n_{z,max}mg}{\rho(t)V^2(t)S} \tag{5.27}$$

$$C_D(t) = f_M(C_L(t)) \tag{5.28}$$

$$D(t) = \frac{1}{2}\rho(t) \cdot S \cdot C_D(t)V^2(t) \tag{5.29}$$

$$T(t) = T_{(I)} + t \cdot \frac{T_{max} - T_{(I)}}{\Delta t_T} \tag{5.30}$$

Results Figure 5.3 shows first results of calculations with the described model. Starting at an initial height of $500\ m$ with an incident *flight path angle* of -3^o, three *pull - up maneuvers* with different initial speeds have been evaluated (*flaps* 3, *gear down*, $m = 50\ t$).

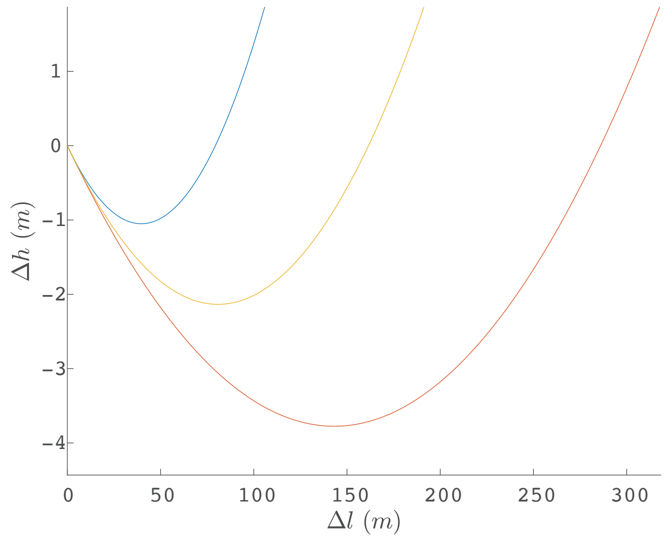


Figure 5.3: Blue: $V_{\tau*} = 100\ m/s \Rightarrow \Delta h = 1.05\ m$, $\Delta l = 80\ m$, $\gamma_{max} = 15.4^o$, Yellow: $V_{\tau*} = 150\ m/s \Rightarrow \Delta h = 2.13\ m$, $\Delta l = 163\ m$, $\gamma_{max} = 15.4^o$, Red: $V_{\tau*} = 200\ m/s \Rightarrow \Delta h = 3.78\ m$, $\Delta l = 287\ m$, $\gamma_{max} = 15.4^o$.

5.1.2 Time to TAWS Alert

As discussed in the previous chapters, a central part of the algorithm calculates the *Time to the Point of no Return* depending on the *CFIT - Trajectory*.

$$t_{PONR,S} = \Pi(\underline{\tau}_{(GPS),S}(t)).$$

The subsequent risk assessment is based on $t_{PONR,S}$:

$$\hat{\zeta}_S = \hat{\zeta}_S(\underline{\tau}_{(GPS),S}) = \begin{cases} \frac{1}{\Delta^2}\left(\Delta - \Pi(\underline{\tau}_{(GPS),S})\right)^2 & , \forall \underline{\tau}_{(GPS),S} \in \underline{\oplus} \\ 0 & , \forall \underline{\tau}_{(GPS),S} \notin \underline{\oplus} \end{cases}$$

The *Point of no Return* marks a place, where, according to the given definitions and simplifications, a crash of the aircraft is unavoidable. Another approach to mark a place of high accident risk would be to calculate the *time to TAWS alert*, where *TAWS* stands for **T**errain **A**wareness and **W**arning **S**ystem. This device gives visual and aural warnings to the cockpit crew, if certain safety limits are exceeded. According to the *Honeywell MKV manual* [10], on which the elaborations of this section are based on, there are several modes implemented in a *TAWS* system. The following enumeration lists the most important ones regarding *CFIT*:

- Mode 1: *Excessive Descent Rate*
- Mode 2: *Excessive Closure to Terrain*
- Mode 3: *Altitude Loss after Takeoff*
- Mode 4: *Unsafe Terrain Clearance*
- Mode 5: *Excessive Deviation Below Glideslope*

Each mode shall be briefly explained in the next paragraphs.

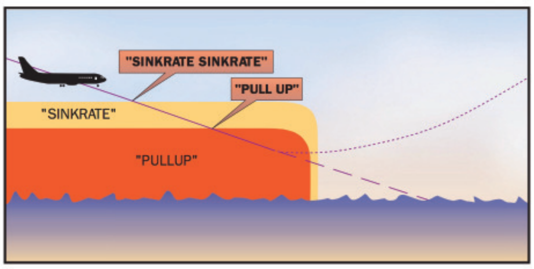


Figure 5.4: Excessive Descent Rate [10].

Excessive Descent Rate This mode is active for all flight phases and has inner and outer alert boundaries. In the yellow area, which is less critical, the pilots will hear the message *"Sinkrate, Sinkrate!"* in addition to flashing lights on the control panel. If the inner alert boundary is penetrated, which is marked by the red area, the aural message changes to *"Pull Up, Pull Up!"*.

Excessive Closure to Terrain This mode is split into two "submodes", of which one is active for climbout, cruise and initial approach, whereas the other one is automatically selected when the flaps are in landing configuration. The warnings are based on the measurement of the *radio altitude* and are split into two areas: *"Terrain, Terrain!"* and the more hazardous *"Pull Up, Pull Up!"*. They are activated when the terrain level rises to quickly underneath the *flight path*, such that a mountain can be expected ahead.

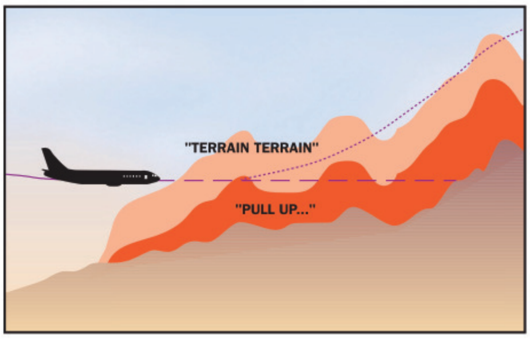


Figure 5.5: Excessive Closure to Terrain [10].

Altitude Loss after Takeoff Until the aircraft ascended to a safe flight level, the mode *Altitude Loss after Takeoff* is monitoring the takeoff. Based on the height above ground, measured by the *radio altitude* and the *vertical speed*, alarm messages are given as a function of momentary height and sink rate. In contrast to the other modes, *Altitude Loss after Takeoff* has only one alarm area: *"Don't Sink, Don't Sink!"*.

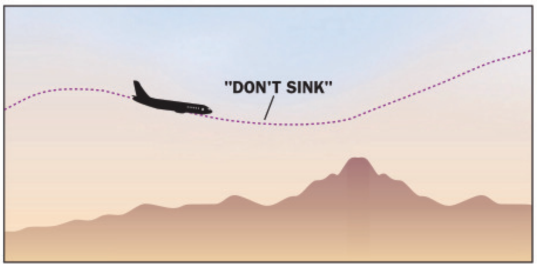


Figure 5.6: Altitude Loss after Takeoff [10].

Unsafe Terrain Clearance Mode 4 splits into three further "submodes". All in all, this mode is available during the whole flight, but the different alert classes are selected due to *flap* and *gear* configuration. It helps preventing unintended *gear up landing* or reminds of setting the *flaps* to landing configuration, when approaching the runway. In clean configuration the distance above ground is monitored, whereas in non - clean configuration the alerts are triggered by the computed airspeed, since it is expected to set *flaps full* and *gear down* when speed decreases significantly. The critical aural message of mode 4 is *"Too Low Terrain!"*.

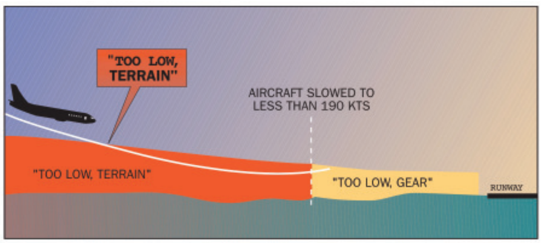


Figure 5.7: Unsafe Terrain Clearance [10].

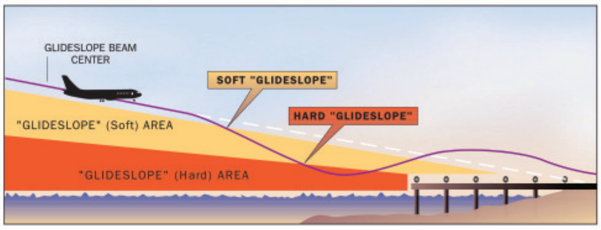


Figure 5.8: Excessive Deviation Below Glideslope [10].

Excessive Deviation Below Glideslope The last mode is implemented in two alerting stages, as well, and depends on the *radio altitude* and the *glideslope deviation* measured in dots. The pilot hears *"Glideslope, Glideslope!"* at half the volume of the red area, when a warning is produced above 300 feet. Below this height, warnings come up in the usual loudness.

According to the five presented alert modes, the function Π is modified, such that it returns the *time to TAWS alert*. That means, that our new function $\tilde{\Pi}$ calculates the time until one of the critical (red) *TAWS alerts* would become active, if the aircraft followed the given *CFIT - Trajectory* $\underline{\mathcal{T}}_{(GPS),S}$:

$$t_{PU,S} := \tilde{\Pi}(\underline{\mathcal{T}}_{(GPS),S}(t)). \tag{5.31}$$

This new variable $t_{PU,S}$ is simply called *time to pull - up* and would directly influence the subsequent risk assessment:

$$\hat{\zeta}_S = \hat{\zeta}_S(\underline{\mathcal{T}}_{(GPS),S}) = \begin{cases} \frac{1}{\Delta^2}\left(\Delta - t_{PU,S})\right)^2 & ,\forall \underline{\mathcal{T}}_{(GPS),S} \in \underline{\oplus} \\ 0 & ,\forall \underline{\mathcal{T}}_{(GPS),S} \notin \underline{\oplus} \end{cases} \tag{5.32}$$

The exact numbers and borders for every alert mode are very well documented in the *Honeywell manual* [10] and could thus be implemented as an extension to the original algorithm.

5.1.3 Deviations due to Magnetic Declination

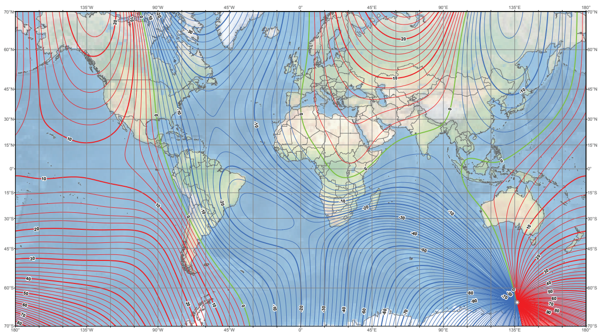


Figure 5.9: Isolines of magnetic declination [3].

Using a compass for navigational purposes is not as easy as it seems at first sight. The needle points to the magnetic north pole of the earth, which does not coincide with the geographic north pole. Furthermore, the magnetic field of the earth varies all across the surface due to irregularities in the earth's crust and changes in time as well [3]. The airport database of the *Institute for Flight System Dynamics* delivers the *bearing angle* $\underline{DIR}$ of the runway. During the evaluation of the results it turned out, that $\underline{DIR}$ is given in *magnetic north* and not in *geographic north*, as expected. There is a software (language C) supplied by the *National*

Geospatial - Intelligence Agency (NGA) called *"The World Magnetic Model"* [3], which calculates the *declination angle* based on position and date. Since there are *MATLAB* routines given, which easily integrate external C - code, *"The World Magnetic Model"* can extend the *CFIT - Analysis - Algorithm* in the future.

Further Integration into the Flight Safety IT Environment

The time points for *takeoff* and *touchdown* shall be taken from the *Flight Safety IT System* in the future and not be calculated in the *CFIT - Analysis - Algorithm* itself anymore.

5.2 Summary

In the course of this semester thesis a *MATLAB* algorithm was implemented, that takes *FDM* Data as input and computes based on the *ASTER GDEM* four newly introduced parameters: the number of *CFIT - Incidents* N_{CFIT}, the *CFIT - Number* of the approach or flight ζ, and the pair of values according to the *Final Approach Distribution Analysis* $\left(\bar{\delta}, \sigma_{\delta}\right)$. We could show, that all together are able to describe the quality of the flight or approach regarding the accident scenario *Controlled Flight Into Terrain*, short *CFIT*. The underlying terrain database was chosen after a comparison of four freely available elevation models, of which *ASTER GDEM* turned out to be the best, because of coverage, void patching and accuracy. There are still open tasks and problems to be solved such as the magnetic declination, as well as a further elaboration of physical models inside the analysis, but by the evolution of the algorithms the tool will be able to assess airports, flights, runways, pilots and many more regarding *CFIT - risk*.

References

[1] *Airport Database*, Visited on 11.10.2017, URL: http://worldaerodata.com.

[2] ASTER-GDEM-Validation-Team, "ASTER Global Digital Elevation Model Version 2 – Summary of Validation Results", in: (2011).

[3] Chulliat, Arnaud et al., "The US/UK World Magnetic Model for 2015-2020: Technical Report", in: (2015), DOI: 10.7289/V5TB14V7.

[4] *Die gefährlichsten Landebahnen der Welt - Hilfe, wir landen!*, Visited on 11.10.2017, URL: http://www.sueddeutsche.de/reise/die-schlimmsten-runways-der-welt-fertig-vom-landen-1.385780.

[5] *Earth Explorer*, Visited on 08.10.2017, URL: https://earthexplorer.usgs.gov.

[6] EASA, *Flight Data Monitoring on ATR Aircraft*, vol. 1, 2016.

[7] EOFDM-Working-Group-A, *Review of Controlled Flight Into Terrain (CFIT) - Precursors from an FDM Perspective*, vol. 1, 2017.

[8] FAA, *Statistical Loads Data for the Airbus A-320 Aircraft in Commercial Operations*, 2002.

[9] Florian Holzapfel, *Flugsystemdynamik 1: Skriptum zur Vorlesung*.

[10] Honeywell, *MK V and MK VII - Enhanced Ground Proximity Warning System (EGPWS) and Runway Awareness Advisory System (RAAS) Pilot Guide*, English, version 060-4241-000.

[11] *How is a digital elevation model made?*, Visited on 08.10.2017, URL: http://www.dlr.de/eoc/en/Portaldata/60/Resources/dokumente/7_sat_miss/srtm_DEM_generierung_en.pdf.

[12] IATA, *Controlled Flight Into Terrain Accident Analysis Report*, vol. 1, 2014, ISBN: 978-92-9252-777-8.

[13] ICAO, *Aviation Occurrence Categories - Definitions and Usage Notes*, vol. 4.6, 2013.

[14] Jeffrey J. Danielson, Dean B. Gesch, "Global Multi-resolution Terrain Elevation Data 2010 (GMTED2010)", in: (2011).

[15] Koornstra, Matthijs et al., *Transport Safety Performance in the EU - A Statistical Overview*, vol. 1, 2003, ISBN: 90-76024-154.

[16] Ritter, Niles, "GeoTIFF Format Specification - GeoTIFF Revision 1.0", in: (1995).

[17] *Space shuttle with extension mast*, Visited on 08.10.2017, URL: https://www2.jpl.nasa.gov/srtm/images/artwork/srtm_40_hi.jpg.

[18] *SRTM Mission*, Visited on 08.10.2017, URL: http://www.dlr.de/eoc/en/Portaldata/60/Resources/dokumente/7_sat_miss/srtm_mission_en.pdf.

[19] *Terra Sattelite*, Visited on 08.10.2017, URL: https://www.nasa.gov/mission_pages/terra/spacecraft/index.html#aster.

Appendix

General Instructions and Notes

- Start *MATLAB* in administrator mode.
- Choose your program options in the beginning of the source code of `cfit_risk_analyzer.m` and save the file.
- Note, that here is another graphical display option which has not been explained yet: `graphical_flight_path_display`. When activated, each flight is displayed in top and side view above ground after finishing the computations.
- Call the `FlightSelection` app.
- Select the flights to be analyzed and proceed to `Generalized Function Call`.
- The algorithm checks if all maps, that cover the flight path are available. **Important:** In case they are not, a warning is displayed, but the calculation continues: Reason: *ASTER GDEM* contains only maps, which cover parts of the earth's land mass. In case a map would have a constant elevation of zero (which is valid for all areas of the oceans), it is not listed in the terrain model. This is why the terrain height of missing maps is internally set to zero by `cfit_risk_analyzer.m`.
- A dot is displayed for every analyzed sample of the flight path.
- The results ($\bar{\delta}$,σ_δ,N_{CFIT},`flightnumber` ,ζ) are stored in a *mat* - file in the following folder structure:
  `result_folder\airline_database\arrival_airport\Flight_flightnumber.mat`
- Only for final approach analysis: In the end a figure is created, which depicts the terrain data, the *flight path*, the *runway area* and all *CFIT - Trajectories* in a 3D - plot, stored in the same folder as `Flight_flightnumber_fig.fig`.

Description of the Implemented MATLAB Functions

Function	Input		Output	
`calculate_rw_end`	`RRP_deg`	$\underline{RRP}_{(GPS)}$		
	`bearing_deg`	$\underline{DIR}_{(GPS)}$	`end_rw_deg`	$\underline{REP}_{(GPS)}$
	`length_m`	l_{RW}		
`create_cfit_distribution`	`tti_alt_cfit`	$(t_{I,S}\ h_S)_j$		
	`V_ref_fa_mDs`	$\|V_{ref}\|$	`mean_value`	$\bar{\delta}$
	`num_cfits`	N_{CFIT}		
	`gldslpeAngle_deg`	γ_{opt}	`sigma`	σ_δ
	`ap_elevation_m`	h_{RW}		
`display_all_cfit_incidents`	`all_cfit_trjectories`	$\underline{\oplus}$		
	`lat_deg`			
	`long_deg`	$\underline{X}_{(GPS)}$	`figure`	
	`alt_m`			
	`start_rw_deg`	$\underline{RRP}_{(GPS)}$		
	`end_rw_deg`	$\underline{REP}_{(GPS)}$		
`direction_angles`	`act_pos`	$\underline{x}_{(GPS),S}$	`alpha_rad`	α
	`future_pos`	$\underline{x}_{(GPS),S+\Delta S}$	`gamma_rad`	γ
`evaluate_map_name`			`foldername`	
	`lat_deg`	$\underline{x}_{(GPS),S}$	`filename`	
	`long_deg`		`map_name`	
			`ref_name`	
`isonrunway`	`start_rw_deg`	$\underline{RRP}_{(GPS)}$		
	`end_rw_deg`	$\underline{REP}_{(GPS)}$	`is_on_runway`	`bool`
	`p_deg`	$\underline{x}_{(GPS),I}$		
`ispointofnoreturn`	`cfit_position`	$\underline{\tau}^*_{(GPS),S}$	`point_of_no_return`	`bool`
	`V_mDs`	V	`curve1`	$\underline{\kappa}^{\tau^*}_{(GPS),r}$
	`alpha_rad`	α	`curve2`	$\underline{\kappa}^{\tau^*}_{(GPS),l}$
	`gamma_rad`	γ	`climb_path`	$\underline{\phi}^{\tau^*}_{(GPS)}$
`meter2deg`	`lat1_deg`	$x_{N,S}$	`meter2deglat`	$\sqcup_{Lat}$
	`alt1_m`	h_S		
	`alt2_m`	$h_{S+\Delta S}$	`meter2deglong`	$\sqcup_{Long}$

Legal Notices

The following conditions had to be agreed in order to obtain the whole *ASTER GDEM* database:

- *"I agree to redistribute the ASTER GDEM *only* to individuals within my organization or project of intended use or in response to disasters in support of the GEO Disaster Theme."*
- *"When presenting or publishing ASTER GDEM data, I agree to include "ASTER GDEM is a product of METI and NASA." "*
- *"Because there are known inaccuracies and artifacts in the data set, please use the product with awareness of its limitations. The data are provided "as is" and neither NASA nor METI/ERSDAC will be responsible for any damages resulting from use of the data."*

LP DAAC User Services Contact Information:
USGS Earth Resources Observation and Science (EROS) Center
47914 252nd Street
Sioux Falls, SD 57198-0001

Voice:	605-594-6116
Toll Free:	866-573-3222
Fax:	605-594-6963
E-mail:	LPDAAC@usgs.gov
Web:	http://LPDAAC.usgs.gov