DEVELOPMENT OF AN AUTOLAND SYSTEM FOR GENERAL AVIATION AIRCRAFT

Diana Siegel and R. John Hansman

This report is based on the S. M. Thesis of Diana Siegel submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Master of Science at the Massachusetts Institute of Technology.

The work presented in this report was also conducted in collaboration with Avidyne, Inc

Report No. ICAT-2011-09
Sept 2011

MIT International Center for Air Transportation (ICAT)
Department of Aeronautics & Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
Development of an Autoland System for General Aviation Aircraft

by

Diana Siegel

Submitted to the Department of Aeronautics and Astronautics on September 28, 2011, in partial fulfillment of the requirements for the degree of Master of Science in Aeronautics and Astronautics

Abstract

Accidents due to engine failure, pilot disorientation or pilot incapacitation occur far more frequently in general aviation than in commercial aviation, yet general aviation aircraft are equipped with less safety-enhancing features than commercial aircraft. This thesis presents the design of an emergency autoland system that includes automatic landing site selection, guidance to the selected landing site and guidance along the final approach, in addition to the automatic landing capability provided by conventional autoland systems. The proposed system builds on the capability of a general aviation autopilot, flight management system and GPS/WAAS augmented, integrated navigation system. The system provides this automatic landing capability without the use of automatic throttle control and without the use of a radar altimeter, which are essential to conventional autoland systems, but are typically lacking on general aviation aircraft. The design addresses the challenge of no automatic throttle control by utilizing only two simple power settings: cruise power and zero power. The lack of radar altimeter is addressed by appropriate flare planning and placement of the target touchdown point. The approach from the point of autoland initiation, to the approach fix at the the landing site, is performed at cruise power, provided that power is available. The final approach from the approach fix to touchdown, is performed at zero power. Control of the touchdown point location during the final approach is achieved through adjustment of the length of the trajectory, whenever the aircraft’s glide performance deviates from the expected performance. The aircraft’s glide performance is measured online as the aircraft tracks the planned trajectory. The performance of the final design is evaluated in simulation in terms of touchdown point dispersion, sink rate and attitude on touchdown.

Thesis Supervisor: R. John Hansman
Title: Professor of Aeronautics and Astronautics
Acknowledgments

This work has been performed in collaboration with Avidyne, Inc. and was supported by the Avidyne Fellowship.
Contents

1 Introduction 21

1.1 Background .............................................. 21
1.2 Motivation for Design of Autoland System ....................... 22
1.3 Design Goal ........................................... 25
1.4 Design Constraints ....................................... 27
1.5 Literature Review ....................................... 30

2 System Description 33

2.1 System Concept .......................................... 33
2.1.1 Overview ........................................ 33
2.1.2 Engine Status Detection ................................ 35
2.1.3 Landing Site Selection ................................ 36
2.1.4 Autoland Approach Procedure .......................... 39
2.1.5 Baseline Holding Pattern and Traffic Pattern ................. 41
2.1.6 Trajectory Planning using Energy Management .................. 44
2.1.7 Performance Prediction using Online Performance Estimation .... 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.8</td>
<td>Aircraft Energy Monitoring and Energy Error Detection</td>
<td>50</td>
</tr>
<tr>
<td>2.1.9</td>
<td>Trajectory Updating</td>
<td>53</td>
</tr>
<tr>
<td>2.1.10</td>
<td>Placement of the Target Touchdown Point</td>
<td>63</td>
</tr>
<tr>
<td>2.1.11</td>
<td>Flare</td>
<td>70</td>
</tr>
<tr>
<td>2.2</td>
<td>Autoland - FMS Interface</td>
<td>71</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Trajectory Specification</td>
<td>71</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Use of Local and Global Coordinate Systems</td>
<td>74</td>
</tr>
<tr>
<td>2.3</td>
<td>User Operations</td>
<td>77</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Autoland Initiation</td>
<td>77</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Increase Throttle Command</td>
<td>77</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Status Display for Selected Landing Site and Planned Trajectory</td>
<td>78</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Prepare to Land Tutorial</td>
<td>79</td>
</tr>
<tr>
<td>2.3.5</td>
<td>Request to Reduce Throttle at Initial Point</td>
<td>79</td>
</tr>
<tr>
<td>2.3.6</td>
<td>Request to Apply Brakes</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td><strong>System Architecture</strong></td>
<td>81</td>
</tr>
<tr>
<td>3.1</td>
<td>Static Decomposition</td>
<td>82</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Autoland Layers</td>
<td>82</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Autoland Components</td>
<td>83</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Autoland Modules</td>
<td>85</td>
</tr>
<tr>
<td>3.2</td>
<td>Autoland Modes and Phases</td>
<td>89</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Modes</td>
<td>89</td>
</tr>
</tbody>
</table>
3.2.2 Phases ......................................................... 91
3.3 Autoland Interfaces ............................................ 94
  3.3.1 External Interfaces ......................................... 94
  3.3.2 Internal Interfaces .......................................... 97
  3.3.3 Design Constants ............................................ 102
  3.3.4 Complete System Diagram ................................ 103

4 Simulation Environment ........................................ 105
  4.1 Aircraft Model for Evaluation of Touchdown Point Dispersion .......... 105
    4.1.1 Point Mass Model ......................................... 106
    4.1.2 Trajectory Tracking Controller .......................... 109
  4.2 Aircraft Model for Evaluation of Flare Performance .................. 111
    4.2.1 High-fidelity Aircraft Model ............................ 111
    4.2.2 Flare Controller ........................................... 112
  4.3 Wind Model .................................................. 113
  4.4 Wind Estimator Model ......................................... 114
  4.5 Performance Estimator Model .................................. 115
  4.6 Energy Manager Model ........................................ 116
  4.7 Trajectory Updater Model ...................................... 116

5 Results .......................................................... 117
  5.1 Evaluation of Effect of Wind on Touchdown Point Dispersion ........ 118
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.1</td>
<td>Test Setup For Evaluation of Effect of Wind on Touchdown Point Dispersion</td>
<td>118</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Illustration of Test Case 1 for Evaluation of Effect of Wind on Touchdown Point Dispersion</td>
<td>120</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Summary of Effect of Wind on Touchdown Point Dispersion</td>
<td>125</td>
</tr>
<tr>
<td>5.1.4</td>
<td>Discussion of Effect of Wind on Touchdown Point Dispersion</td>
<td>126</td>
</tr>
<tr>
<td>5.2</td>
<td>Evaluation of Flare Performance</td>
<td>127</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Test Setup for Evaluation of Flare Performance</td>
<td>127</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Illustration of Test Case 1 for Evaluation of Flare Performance</td>
<td>128</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Summary of Flare Performance Test Results</td>
<td>131</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Discussion of Flare Performance Test Results</td>
<td>132</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion and Future Work</td>
<td>135</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusion</td>
<td>135</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Work</td>
<td>136</td>
</tr>
<tr>
<td>A</td>
<td>Module Descriptions</td>
<td>141</td>
</tr>
<tr>
<td>A.1</td>
<td>Modules of Layer Destination Planning</td>
<td>141</td>
</tr>
<tr>
<td>A.1.1</td>
<td>Engine Status Detection</td>
<td>141</td>
</tr>
<tr>
<td>A.1.2</td>
<td>Range calculator</td>
<td>142</td>
</tr>
<tr>
<td>A.1.3</td>
<td>Landing Site Prioritizer</td>
<td>144</td>
</tr>
<tr>
<td>A.1.4</td>
<td>Initial Point Calculator</td>
<td>149</td>
</tr>
<tr>
<td>A.1.5</td>
<td>Touchdown Energy Calculator</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Function</td>
<td>Page</td>
</tr>
<tr>
<td>---</td>
<td>------------------</td>
<td>------</td>
</tr>
<tr>
<td>A.4.4</td>
<td>calcDistToGo</td>
<td>200</td>
</tr>
<tr>
<td>A.4.5</td>
<td>calcInterceptPoint</td>
<td>202</td>
</tr>
<tr>
<td>A.4.6</td>
<td>calcTurnDist</td>
<td>204</td>
</tr>
</tbody>
</table>
## List of Figures

1-1 Autoland integration with its environment ................................................. 26

1-2 Airspeed and flight path angles for various throttle settings ....................... 28

2-1 Overview of autoland concept ........................................................................ 35

2-2 Pseudocode for automatic engine status detection ....................................... 36

2-3 Landing site selection process ........................................................................ 38

2-4 Aircraft trajectory during autoland for case engine operational and no obstacles 40

2-5 Sequence of functions performed during approach procedure in case engine
  operational ........................................................................................................... 41

2-6 Length and shape of the baseline trajectory .................................................... 43

2-7 Naming convention for holding pattern legs and traffic pattern legs .............. 46

2-8 Energy dissipation with ground distance travelled ......................................... 47

2-9 Determination of observed glide ratio while in flight ..................................... 49

2-10 Calculation of the expected glide ratio, $L/D_{exp}$, from $L/D_0$ .................... 50

2-11 Concept of excess energy correction through trajectory elongation .............. 51

2-12 Update of energy vs. range-to-go reference curve upon receipt of new aircraft
  glide performance and wind estimate ................................................................... 52
A-9 Baseline and updated trajectory for max. and min. energy error expected
(rotated by $-90^\circ$ from local coordinate system) .......................... 169
A-10 Local energy vs. range curve function representation .................. 173
A-11 Energy error tolerance bounds ............................................. 174
A-12 d vs. E/W plot for various $L$ with $\alpha_p = \text{const} = 45^\circ$ .............. 178
A-13 d vs. E/W plot for various $\alpha_p$ with $L=\text{const} = 5,000 \text{ ft}$ ............. 179
A-14 Geometry used to determine difference in path length between the current
and updated trajectory ................................................................. 181
A-15 d vs. E/W plot for various $L$ with $d_{total}=500 \text{ ft}$ and $\alpha_p = \text{const} = 30^\circ$ .... 183
A-16 d vs. E/W plot for various $\alpha_p$ with $d_{total} = 500 \text{ ft}$ and $L = 3,000 \text{ ft}$ ...... 183
A-17 d vs. E/W plot for various $d_{total}$ with $\alpha_p = 45^\circ$ and $L = 3,000 \text{ ft}$ ...... 184
A-18 d vs. E/W plot for extreme $\Delta E/W$ to illustrate corner cases .............. 185
A-19 Geometry used to determine location of IP and OP ........................ 190
A-20 Calculation of turn entry line .................................................. 197
A-21 Geometry used to determine whether turn entry line has been crossed at
location, P ................................................................. 200
A-22 Calculation of Range-To-Go until start of the next segment .............. 202
A-23 Calculation of distance from line ............................................. 203
A-24 Geometry used to determine the lead distance to the waypoint at which a
turn should be initiated ................................................. 205
List of Tables

1.1 GA accidents in 2007 and 2008 ........................................ 23
1.2 Requirements on autoland system .................................... 27
3.1 Autoland external interfaces ........................................... 96
3.2 Autoland internal interfaces .......................................... 101
3.3 Autoland design constants ............................................ 102
5.1 Summary of test results ............................................... 125
5.2 Summary of flare performance test results ....................... 132
Chapter 1

Introduction

1.1 Background

Automatic landing systems have existed for about half a century, with British Airways having made the first automatic landing during a commercial flight in June 1965 using an automatic flare controller in the pitch axis [1]. A landing with full control in all three axes was achieved shortly after in 1966. Landing aids that guide the aircraft down to flare height have been developed since 1920 [16]. On September 19, 1930, H. Diamond and F.W. Dunmore presented the design and hardware for a landing aid based on radio navigation similar to today’s Instrument Landing System (ILS) [9]. Today’s commercial autoland systems utilize the ILS’ vertical and lateral guidance to descend towards the runway at a glide path of $-3^\circ$, while keeping the aircraft’s track aligned with the runway centerline. At around 30 ft above the ground, the aircraft enters the flare mode, during which the aircraft pitches up to reduce the vertical speed from its current descent rate to 1-3 ft/s by touchdown [18].

Automatic landing equipment is certified to either category CAT I, CAT II or CAT III, which determines the minimum altitude above the touchdown zone that the automatic landing system may be used. This minimum altitude is defined as the decision height. At the decision height, the pilot must have the runway in sight and be able to see as far down the
runway as prescribed by the minimum *runway visual range*. If, at the decision height, the runway is not in sight, or runway visual range is smaller than the minimum range specified, the pilot is required to execute a missed approach and go-around to re-attempt the landing or divert to another airport. In order to be able to perform an approach in a specific category, both the on-board and the ground-based equipment need to be certified to at least this specific category.

General aviation aircraft level equipment is typically certified to CAT I only, which specifies a decision height not lower than 200 ft and a runway visual range not less than 1,800 ft. Therefore, general aviation autopilot systems must be decoupled at 200 ft and the landing performed manually. Commercial aircraft are typically certified to CAT III, which is sub-divided into CAT IIIa, CAT IIIb and CAT IIIc. CAT IIIc specifies zero decision height and zero runway visual range and is to the author’s knowledge at this time not certified for any ground-based equipment at any airport worldwide. CAT IIIb authorizes landings with as low as zero decision height and runway visual range as low as 150 ft. The CAT IIIb ILS approach to runway 25L at LAX, for example, is certified for zero decision height and runway visual range of 600 ft. Certification to CAT III requires redundant auto-flight systems including redundant ways of actuating all flight controls and redundant autopilots. Furthermore redundant radar altimeters are required to accurately measure the distance over ground. This type of equipment is commonly found on commercial aircraft but not typically on general aviation aircraft since the hardware cost is prohibitive for most private operators.

1.2 Motivation for Design of Autoland System

Accidents in commercial aviation have continuously decreased since the early days of aviation with typical fatal accident rates now being less than 0.02 per 100,000 flight hours with a total of 16 million flight hours [22]. The fatal accident rate in general aviation, however, is more than 50 times higher than that, with a rate of 1.3 per 100,000 flight hours for a total of 30 million flight hours [6]. In part, this is due to the fact that general aviation pilots fly
shorter routes and hence have more takeoffs and landings, the most accident-prone part of the flight, compared to commercial operators. The difference in pilot training and aircraft equipage, however, also contributes to the high accident rates in general aviation compared to commercial aviation. Commercial aircraft are equipped with many safety enhancing features, such as traffic advisory systems, ground proximity warning systems, etc. that general aviation aircraft lack. Given the accident statistics, however, there is an equal need for additional safety enhancing features on general aviation aircraft.

Accidents in general aviation occur due to mechanical failures, adverse weather and pilot error to name a few. This thesis addresses two specific causes for accidents in general aviation, which taken together produce a large proportion of the fatal accidents: pilot incapacitation and loss of power. Though pilot incapacitation contributes few numbers of accidents, the chance that this type of accident ends fatal, is much higher than for any other type of accident. Pilot incapacitation may occur due to hypoxia from excess altitude, or any other medical condition. Loss of power is either due to mechanical failure or fuel mismanagement, which occurs surprisingly often, despite being entirely preventable. Table 1.1 summarizes accident statistics for general aviation aircraft in 2007 and 2008. Accidents due to loss of power and pilot incapacitation amounted to 25.0% and 22.6% of all general aviation accidents in 2007 and 2008 respectively [3, 2].

In case of pilot incapacitation, the pilot is either completely or at least partially inca-

<table>
<thead>
<tr>
<th></th>
<th>2007 Total (fatal)</th>
<th>2008 Total (fatal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanical failure (with power loss)</td>
<td>132 (14)</td>
<td>104 (9)</td>
</tr>
<tr>
<td>Unknown loss of power</td>
<td>118 (12)</td>
<td>102 (14)</td>
</tr>
<tr>
<td>Fuel mismanagement</td>
<td>90 (9)</td>
<td>73 (9)</td>
</tr>
<tr>
<td>Pilot incapacitation</td>
<td>6 (4)</td>
<td>3 (3)</td>
</tr>
<tr>
<td>Total loss of power + pilot incapacitation accidents</td>
<td>346 (39)</td>
<td>283 (35)</td>
</tr>
<tr>
<td>Total number of GA accidents</td>
<td>1385 (252)</td>
<td>1254 (236)</td>
</tr>
</tbody>
</table>

Table 1.1: GA accidents in 2007 and 2008
pable to maintain control of the aircraft. In case of loss of power, the achievable flight range is heavily reduced, which makes it difficult for the pilot to judge which landing sites are reachable. Hence, providing automatic control of the aircraft’s trajectory and selecting and guiding the aircraft to a reachable landing site, are two main objectives that an autoland system aimed at the scenarios of pilot incapacitation and loss of power will need to address.

There would be no obstacles to implementing this kind of system, if autoland had access to a representation of the aircraft’s environment and location of landing sites, and had automatic control over all flight control surfaces and the throttle. The autopilot and flight management systems as commonly found on commercial aircraft, provide precisely this functionality. A decade ago, these systems were rarely found on general aviation aircraft, but they are starting to be integrated more and more on this class of aircraft.

For example, the Cirrus SR-22 incorporates a complete flight automation solution with autopilot and flight management system. This system allows full control of the aircraft’s motion in the lateral plane, but only limited control in the vertical plane, since this class of aircraft typically lacks automatic throttle control. Furthermore, its integrated navigation systems (INS) features WAAS (wide area augmentation system) enabled GPS receivers, which provide a much better position accuracy than previous GPS receivers. Without Selective Availability, which was turned off in 2000, previous GPS receivers achieved a typical accuracy of 15 m laterally and 20 m vertically. This level of accuracy was clearly insufficient for the landing task. The WAAS specification requires that a position accuracy of better than 7.6 m be achieved both laterally and vertically at least 95% of the time. However, measurements in the continental US have shown that GPS/WAAS receivers achieve a typical accuracy of < 2 m laterally and vertically, which provides the potential to perform a precision approach based on the WAAS position information only [4]. This improvement in the accuracy of position sensing is important, since general aviation aircraft typically do not possess a radar altimeter, which provides commercial aircraft with centimeter level vertical position accuracy during the final stage of the landing.
1.3 Design Goal

The aim of this thesis is to design an autoland system to be used in case of emergency. Specifically, the design addresses the following scenarios:

- Partial or complete pilot incapacitation
- Loss of power

Since loss of power can occur at any point during the flight and the pilot can become incapacitated at any point in time, the autoland system to be developed needs to have a larger scope than the typical autoland systems on commercial aircraft, which only take over once the aircraft is aligned with the runway centerline on final approach. For the case of partial or complete pilot incapacitation, the pilot needs to be able to rely on autoland to automatically select a suitable landing site, guide the aircraft to the landing site and perform the approach and landing. For complete pilot incapacitation, autoland may not require any input from the pilot. However, there is the possibility of a passenger being present, which autoland aims to draw on, in order to provide control inputs to enhance autoland performance in the event of complete pilot incapacitation. This passenger is assumed to be untrained in piloting the aircraft, so that any interaction with the passenger should be kept as simple as possible. The same applies to any interaction with a partially incapacitated pilot.

For the case of loss of power, the pilot can assumed to be alert. However, in this situation, it is difficult for the pilot to judge which landing sites are reachable, how to best reach the landing site given en-route terrain and how to match the length of the final approach to the glide performance of the aircraft. Hence, automatic landing site search and guidance to the landing site is required in this scenario, in addition to guidance along the final approach.

This capability will need to be achieved with only the typical hardware equipage found on a general aviation aircraft. Adding hardware is considered to not be an option, since the cost would inhibit adoption of the system. The model aircraft used to represent a "typical" general aviation aircraft is a Cirrus SR-22 equipped with an Avidyne DFC-90 autopilot,
GPS/WAAS augmented integrated navigation system (INS) and a flight management system (FMS). It is acknowledged that many general aviation aircraft do not possess this level of automation. However, this design will be targeted at aircraft that are already equipped with an autopilot, FMS and INS.

Figure 1-1 shows the environment, in which autoland is expected to operate. For landing site selection and trajectory planning, autoland can draw aircraft state information, such as aircraft location, velocity and angular rates from the autopilot, and navigation data from the FMS. After having constructed a path to the selected landing site, autoland can pass the desired trajectory to the FMS for tracking in the lateral plane. Based on the commanded trajectory and the aircraft’s current position, the FMS generates roll steering commands, that the autopilot subsequently passes on to the aileron servo. The autopilot itself controls
### Functional requirements

<table>
<thead>
<tr>
<th>FUNC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select a suitable landing site</td>
</tr>
<tr>
<td>2</td>
<td>Compute an obstacle-free trajectory to the selected landing site</td>
</tr>
<tr>
<td>3</td>
<td>Guide the aircraft to the selected landing site</td>
</tr>
<tr>
<td>4</td>
<td>Compute a feasible approach pattern to the touchdown zone</td>
</tr>
<tr>
<td>5</td>
<td>Guide aircraft along the approach pattern</td>
</tr>
<tr>
<td>6</td>
<td>Perform automatic touchdown</td>
</tr>
<tr>
<td>7</td>
<td>Indicate autoland status to pilot and passenger</td>
</tr>
</tbody>
</table>

### Non-functional requirements

<table>
<thead>
<tr>
<th>NFUNC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Implement functionality as software upgrade only (no additional hardware to be included)</td>
</tr>
<tr>
<td>2</td>
<td>Commands given to the operator shall be simple enough to require no prior pilot training</td>
</tr>
<tr>
<td>3</td>
<td>Design system interface to the FMS and the autopilot so that little to no modification of the FMS/autopilot functionality is required</td>
</tr>
</tbody>
</table>

Table 1.2: Requirements on autoland system

The elevator based on the selected vertical mode. Aileron and elevator may be actuated manually or automatically, as illustrated by the connector from the operator to the aircraft and from the autopilot to the aircraft. The throttle, rudder, brakes and flaps, however, may only be actuated manually. This imposes natural limitations on the performance of the autoland system for general aviation aircraft when compared to the performance of the autoland system of a commercial aircraft, which also possesses automatic control over rudder, throttle and brakes. In addition to that, autoland will need to perform the touchdown without a radar altimeter. A summary of the requirements on the autoland system to be developed is given in table 1.3.

### 1.4 Design Constraints

The lack of auto-throttle has the strongest effect on autoland performance, since it limits the achievable flight path angles. Figure 1-2 shows the limitations imposed by the lack of throttle control, by illustrating the range of airspeeds and flight path angles achievable for a
given throttle setting. The throttle setting ranges from 0 to 1 representing a change in engine power output from 0% to 100% ignoring losses and non-linear behavior. If the throttle is stuck in one position, the aircraft’s motion in the vertical plane can only be effected by the elevator. The elevator can be used to increase or decrease the flight path angle and increase and decrease the airspeed, but cannot control both the flight path angle and the airspeed at the same time. As a result, the elevator can increase the flight path angle only until the airspeed reaches the stall speed, and decrease the flight path angle only until the airspeed exceeds the aircraft’s maximum design speed. Hence, the lack of auto-throttle limits the range of flight path angles achievable for a given throttle setting.

If the pilot became incapacitated during cruise flight, i.e. 0° flight path angle, the throttle

![Figure 1-2: Airspeed and flight path angles for various throttle settings](image)

would typically be fixed between 50% and 70%. The airspeed would hence measure between 140 - 160 kt. The standard ILS glideslope guides the aircraft along a flight path angle of −3°. At a throttle setting of 60%, this flight path angle would result in an airspeed of 170 kt, which is excessive given the recommended approach speed of 90 kt for this aircraft.
Approaching at this speed would result in the nose wheel hitting the ground first, if an attempt to land were made, or the aircraft floating off the runway and regain altitude, if the nose was attempted to be pulled up before touchdown. In order to avoid stall or excessive airspeed during final approach, the throttle setting should be kept within 6% to 8%. It should be recognized, that manually setting the throttle to this very narrow region is a non-trivial task, especially when the exact numbers will need to be adjusted for wind and aircraft performance from one landing to another. Assuming the pilot is incapacitated, a passenger would most likely not be able to establish the correct throttle setting in time. Therefore, the autoland system to be developed, is forced to consider alternatives to the standard approach on a −3° glideslope. During the loss of power scenario, the throttle setting can be taken as effectively zero. Inspection of Figure 1-1 shows that the flight path angle at the recommended approach speed of 90 kt is −6°.

The lack of radar altimeter is considered to have the second strongest effect on autoland performance. Not knowing the exact time and location, at which the aircraft will touch down, leads to touchdown point dispersion and the requirement on autoland to plan the flare so that the aircraft an earlier, or later than expected touchdown can be tolerated.

The lack of rudder affects autoland’s ability to ”de-crab” the aircraft during a crosswind landing before touchdown, i.e. aligning the nose of the aircraft with the runway centerline. Touching down sideways with the nose pointing off the runway centerline, would exert an undesirable load on the landing gear and potentially cause the aircraft to run off the side of the runway. However, this is deemed to be tolerable, given that the system is aimed at an emergency situation.

Finally, if neither the pilot nor the passenger are able to operate the brakes, the lack of automatic braking would cause an elongation in rollout distance of at minimum a factor of five, which would cause the aircraft to run off the end of the runway. This is considered to be a dangerous scenario, since obstacles are likely to be present beyond the end of the runway. Hence the pilot or passenger would need to be requested to apply brake pressure, if at all possible. If no manual input can be provided, autoland would need to exercise any
means available to decrease the stopping distance. One possibility is to deliberately run the aircraft off the side of the runway to utilize the higher rolling friction provided by grassy or dirt areas.

It should be noted that, if the aircraft autoland is designed for, uses a retractable gear, the pilot would need to be requested to lower the gear before the approach is started. Since the Cirrus SR-22 is fitted with non-retractable landing gear, this step is not necessary.

1.5 Literature Review

Emergency landing aids have previously been investigated for application to commercial aircraft. Meuleau, Plaunt and Smith proposed an emergency landing planner for damaged aircraft, which determines potential landing sites and feasible trajectories to the landing site and organizes each solution by increasing amount of risk. The decision to which airport to fly along which route is left to the pilot [15]. Furthermore, Atkins, Portillo and Strube presented an emergency flight planner specifically applied to total loss of thrust for general aviation aircraft. This planner calculates the aircraft’s reachable footprint and uses prioritization based on information from the FMS database to automatically find the most suitable landing site. Subsequently, the planner generates a trajectory to the touchdown zone by using estimates on the minimum and maximum achievable vertical flight path angle. The planner is adaptive in the sense that it updates the projected trajectory based on changes in aircraft performance [7]. The space shuttle’s mission abort flight manager (SAFM) uses a similar approach. Based on the current state vector of the shuttle, the SAFM determines a feasible landing site and calculates a trajectory to this landing site using aircraft performance boundaries. These boundaries are determined from knowledge of the maximum and minimum drag profiles achievable by the control system [14]. The space shuttle’s terminal area energy manager (TAEM) synthesizes a landing trajectory from pre-defined maneuvers and controls energy vs. range-to-go until touchdown by modulating dynamic pressure and speed brake. If these modulation techniques are insufficient, the TAEM elongates or shortens the
trajectory to match the observed shuttle energy [10].

Controllers that guide the aircraft along a suitable vertical profile once the aircraft has been aligned with the runway centerline have been designed for commercial aircraft, small and large-scale Unmanned Aerial Vehicles (UAV) and also for general aviation sized aircraft. In all cases, automatic control over thrust, however, has been assumed. Wagner and Valasek, for example, investigated the performance of a QFT controller in comparison to the standard proportional-integral controller for automatic landing of a medium-sized UAVs and general aviation aircraft [21]. Shue and Agarwal designed an H2/H-Infinity controller for automatic landing of commercial aircraft [19].

In 1994, the FAA and UPS successfully tested automatic landings using only differential GPS (DGPS) for positioning information with a Boeing 757 [13]. DGPS uses local ground stations installed at the airport to transmit corrections for the raw GPS signal received from the satellite. Hence, given accurate position information and automatic throttle control, automatic landing controllers have been proven successful for any size of aircraft from small UAVs to large transport aircraft. The specific contribution of this thesis, is to investigate the feasibility of an automatic landing system for general aviation aircraft without automatic throttle control and positioning information only to the accuracy of WAAS augmented GPS.
Chapter 2

System Description

2.1 System Concept

2.1.1 Overview

The autoland system proposed encompasses the tasks from autoland initiation, automatic scenario detection, landing site selection and trajectory planning, to final approach and landing. The autoland system resolves the challenge of no automatic throttle control by using only two simple, distinct power settings: cruise power and zero power. Cruise power is used to fly from the point of autoland initiation to an initial point located at a fixed distance and direction from the runway threshold of the selected landing site. Zero power is then used to fly the final approach from the initial point to the touchdown point. This choice is due to the fact that at zero power, it is always possible to perform a landing without further input from the pilot, assuming that initial conditions and wind speed remain within design range. If the autoland system attempted to perform an approach on the standard $-3^\circ$ glideslope, the system would need to rely on the pilot to reduce the throttle to zero at the time of flare in a timely manner, or else the aircraft would be trapped in a situation, where it is unable to climb in order to abort the landing, and unable to land due to excess power. Excess power
on landing would either result in the nose-wheel touching down first or the aircraft floating off the runway. Furthermore, requesting the operator to reduce the throttle all the way to zero is a simple command to follow. It is reasonable to assume that a partially incapacitated pilot or untrained passenger will be able to follow a command requesting to pull the throttle back when given visual and audible cues. Finally, this approach has the advantage that the "Incapacitated Pilot" and "Loss of Power" scenarios can be treated equally from the start of the gliding approach at the initial point. The two cases need to be treated differently only for landing site selection and guidance to the initial point. Figure 2-1 summarizes the overall design concept.

After initiation, autoland automatically determines whether the engine is operational or has failed. Based on the engine status, autoland selects a landing site within the aircraft’s range and guides the aircraft to the initial point. At the initial point, autoland requests the pilot to reduce the throttle to zero, if the power is not zero already, so that the final approach can be performed at power off. With this choice, the timing of the throttle reduction is not critical, which reduces reliance on the human operator. As long as the power has not been reduced to zero, autoland commands the aircraft to loiter in a holding pattern at the initial point. With this approach, the system is also capable of fully autonomous operation, given that the initial throttle setting is sufficient for straight and level flight. In the event that the pilot is fully incapacitated and cannot actuate the throttle, the aircraft loiters until fuel is depleted. Once the power has reduced to zero due to fuel starvation, the approach is started.

With this concept, the autoland system is capable of autonomously guiding the aircraft from its current location to the selected landing site given that the throttle setting is greater than the setting required to maintain straight and level flight, or smaller than the maximum power at which a successful power on approach can be performed. If the power setting is in between these thresholds, the aircraft will not be able to hold altitude, while not being able to land in a nose-up attitude. In this situation, autoland will audibly and visually request the operator to increase throttle, while aiming for an unpopulated area at wings level. As soon as autoland has detected an increase in power, re-planning will be performed in order
to select a more favorable emergency landing site.

2.1.2 Engine Status Detection

Engine status detection is performed automatically using the %power indicator available on the Cirrus SR-22. The engine status will either be classified as "Engine Operational", "Engine Out" or "Indeterminate". The engine status is deduced to be "Engine Operational" if the engine output power is high enough so that straight and level flight can be maintained. This setting will vary from aircraft to aircraft and is denoted by $p_u$. For the SR-22, this value is about 20%. The engine is deduced to have failed, if power output is smaller than the maximum power setting at which a successful power-on approach can be performed. This value is referred to by $p_l$. For the SR-22, this is about 5%. This value will again be different
for different aircraft. In case the power setting is in between \( p_u \) and \( p_l \), the engine status becomes "Indeterminate". This is a dangerous case, since it is not immediately handled by the current system design without throttle adjustment from the operator. At this throttle setting, the aircraft is neither capable of maintaining altitude nor capable of performing a landing without risking nose-wheel barrowing or floating down the runway. Furthermore, the shallow descent angle also makes the aircraft prone to hitting obstacles while on approach. If this case is detected, autoland visually and audibly requests the pilot to increase throttle to a cruise power setting, which is about 60%.

### 2.1.3 Landing Site Selection

The landing site selection function is responsible for choosing a landing site within range that can be reached without colliding with terrain or known man-made obstacles along the way. The term "within range" requires that the achievable range is 30% higher than the range required to reach the landing site from the current location in order to provide margin. Landing site selection operates differently depending on engine status. The achievable range is determined from glide performance in case the engine is out and from fuel range in case
the engine is operational. Furthermore, since there is a large difference in range between cases engine out and engine operational, the landing site search is performed under more or less demanding criteria levels. The criteria levels will be divided into levels 1-4.

In case engine operational, it is likely that there is a very large number of candidate runways within range. Therefore, the landing site selection process can afford to be more selective. However, a runway 1,000 miles away that is twice as long as a runway 50 miles away, is not better than the runway close by, if the length of the runway close by is sufficient. Hence, not all runways within range should be retrieved and compared to find the very best. Rather, autoland searches for a runway that fulfills a set of ”good enough” criteria within increasing search radii and stops the search process as soon as a runway is found that meets the criteria. This ensures that runways closest to the aircraft’s current position are prioritized. For the engine operational case, the ”good enough” criteria simply consists of the requirement that the runway is at least 5,000 ft long. This is the length required for rollout, absorption of the touchdown point dispersion due to GPS/WAAS vertical position error, error in estimation of wind and glide performance, and a 1,000 ft safety margin. Many other criteria could have been applied, such as wind conditions at the landing site, runway surface, availability of facilities, etc. However, runway length was determined to have the strongest influence on success or failure of the landing and is therefore used as the single selection criteria for the engine operational case. Any runway longer than 5,000 ft is considered to fulfill criteria level 1.

If the engine is out, the range is dramatically reduced, which limits the number of candidate runways up to the point where no reachable runway can be found at all. Therefore, it makes little sense to impose criteria level 1. Instead, autoland searches the full range for a candidate runway and selects the longest one it could find. This runway is then said to fulfill criteria level 2, which only requires that the selected landing site is a runway. In case that no runway could be found, the landing site selector relaxes the criteria level further and searches for a known emergency landing site, which are pre-loaded into the FMS from a database such as the Worldwide Soaring Turnpoint Exchange maintained by the glider
community [11]. A known emergency landing site fulfills criteria level 3. Finally, criteria level 4 consists of any unpopulated area.

In case the engine status is classified as "Indeterminate", the criteria level is initialized to level 4. This is due to the aircraft not being able to perform a landing according to the autoland approach procedure at this intermediate power setting. Hence, the goal is to guide the aircraft away from regions were it could cause damage to the population and to minimize the chances of the aircraft colliding with man-made obstacles.

Figure 2-3 illustrates the landing site selection procedure. First, autoland determines

Figure 2-3: Landing site selection process
the achievable range and initializes the criteria level. Landing site search is initialized with criteria level 1 in case engine out, criteria level 2 in case engine operational and criteria level 4 for engine status indeterminate. The landing site prioritization function then searches for candidate landing sites at the specified criteria level and orders them according to preference. The operation of this function is detailed in the appendix, chapter A.1.3. For criteria level 1 and 2, i.e. runways, candidates are ordered by runway length, with the approach direction chosen as the headwind direction, if available. For criteria level 3, i.e. emergency landing sites, distance to go to the landing site is used for prioritization. For criteria level 4, i.e. unpopulated area, the size of the area determines the position in the ordered list. Autoland then steps through the list of candidate landing sites and attempts to generate an obstacle-free trajectory to the landing site. If successful, the landing site is selected and all lower ranked candidates discarded. If the end of the list is reached and no feasible trajectory could have been generated, autoland relaxes the criteria level and repeats the search at the new criteria level and the process repeats. Details of path planning function, that attempts to generate an obstacle-free trajectory to the initial point are discussed in the appendix, chapter A.2.4

2.1.4 Autoland Approach Procedure

After the landing site has been selected and a feasible path to the initial point determined, autoland uses the trajectory following functionality of the the FMS to guide the aircraft from its current position to the initial point. If there are no obstacles along the straight line trajectory between the aircraft’s current location and the initial point, the aircraft flies a straight line path to the initial point and aligns with the downwind leg by the time it reaches the initial point as shown in Figure 2-4. In this illustration, it is assumed that the engine is operational and that straight and level flight is possible. In case the engine is out, the aircraft would lose altitude along the path to the initial point and the straight line trajectory to the initial point would slope downwards.
At the initial point, autoland requests power to be cut, if it is not zero already. While

Figure 2-4: Aircraft trajectory during autoland for case engine operational and no obstacles waiting for the power cut to occur, autoland keeps the aircraft in a holding pattern at constant altitude. The final approach trajectory is a standard right or left traffic pattern consisting of downwind leg, base leg and final leg. After power has been cut, the aircraft’s altitude, or expressed differently, the aircraft’s energy, could still be too high to start the traffic pattern. In this case, autoland calculates the number and length of holding pattern loops required in order to reduce the aircraft energy from its current level to the energy level required to start the traffic pattern. Upon exit of the holding pattern, autoland adjusts the length of the baseline traffic pattern to match the current aircraft energy and the expected glide performance. Autoland begins to track the generated pattern, using the autopilot to maintain the recommended approach speed of 90 kt.

As pointed out in chapter 1.4, the aircraft will only be able to follow the generated trajectory at constant airspeed, if its performance matches the expected performance exactly. Naturally, the actual glide performance of the aircraft will differ from the expected performance. This difference will make the aircraft deviate off the trajectory, which would result in the aircraft landing long or short of the target touchdown point. In order to correct any error in expected and actual glide performance of the aircraft, autoland monitors the air-
craft’s energy status and adjusts the length of the trajectory in the lateral plane if deviations off the planned trajectory in the vertical plane are observed. This scheme effectively makes the path length the controlled variable to substitute for the lack of control over the aircraft power. Figure 2-5 illustrates this process.

The autoland trajectory can be thought of to consist of three trajectory phases: The

![Figure 2-5: Sequence of functions performed during approach procedure in case engine operational](image)

trajectory to the initial point, the holding pattern and the traffic pattern. While in the holding pattern and traffic pattern, autoland uses trajectory updating to remove energy errors. While the aircraft is tracking the trajectory to the initial point, autoland cannot shorten the path since it already presents the shortest path found. Therefore, the trajectory is planned including range margin, which is monitored as the aircraft proceeds towards the initial point. If the range margin becomes depleted before the initial point is reached, autoland performs re-planning of the currently selected destination and trajectory.

### 2.1.5 Baseline Holding Pattern and Traffic Pattern

The baseline holding pattern consists of two 180°, constant radius turns with zero straight line segments in between the two semi-circles as shown in Figure 2-6. Elongation of the holding pattern to compensate of energy errors is achieved by increasing the length of the straight line segments. The zero length straight line segments were chosen as a baseline, since this provides the smallest possible discretization of altitude levels, at which the holding pattern can be exited.

The nominal bank angle, at which the DFC-90 autopilot performs turns to track the
trajectory is $\phi = 22.5^\circ$. At an airspeed of 90 kt, this results in a turn radius of 1,600 ft from equation 2.1, which results in a circle track length of 10,000 ft.

$$r = \frac{V^2}{g \cdot \tan(\phi)}$$ (2.1)

where $g = 9.81 \frac{m}{s^2}$. For a baseline glide ratio of 10:1, the altitude levels at which the turn can be exited are spaced by 1,000 ft, which creates a challenge for the trajectory generation process upon exit of the holding pattern. If the bank angle was increased to 30°, the turn radius would reduce to 1,100 ft and the circle track length to 7,000 ft, which is a significant improvement. More importantly, however, a small turn radius is needed while following the traffic pattern, in order to allow trajectory updating for a sizable proportion of the overall length of the traffic pattern. When using the current FMS trajectory tracking functions without modification, trajectory updating is inhibited while the FMS captures the course of a new leg. For the two 90° turns within the traffic pattern, this means that trajectory updating is inhibited for one turn radius on the downwind leg and for two turn radii on the base leg. Given that the usual length of the traffic pattern legs is between 3,000 - 5,000 ft, a turn radius of 1,600 ft for the smallest sized traffic pattern would completely inhibit trajectory updating on the base leg. Even if the leg lengths were increased to 5,000 ft, the proportion of the traffic pattern available for trajectory updating is still unsatisfactory at less than 50%. Therefore, it is recommended to increase the nominal bank angle during turns to 30°, which will be assumed to be the nominal bank angle for the remainder of this thesis.

The baseline traffic pattern trajectory is a standard, rectangular, right or left traffic pattern as shown in Figure 2-6. The traffic pattern direction is chosen so that the runway is not crossed during the aircraft’s approach to the initial point. Implementation details are provided in the appendix, chapter A.1.4. The length of the pattern legs is a tradeoff between the capability to correct energy errors as the aircraft follows the downwind leg and base leg and the amount of energy error accumulated on the final leg. The shorter the final leg, the less energy error can be accumulated on the final leg, but at the same time, the less energy
error can be corrected while still on the base leg. The maximum amount, by which the remaining trajectory can be shortened at the start of the base leg is 30%, which is achieved, if base leg and final leg are of equal length. A similar argument applies to the downwind leg and base leg, which motivated the use of all equal length legs. The length of the legs was chosen to be 5,000 ft in order to provide a trajectory updating capability of greater than 50% of the time on the downwind leg and base leg if the turn radius is 1,100 ft. The choice of 5,000 ft of base leg length presents a compromise between trajectory updating capability on the base leg and the goal to minimize the length of the final leg.

Trajectory updating capability during the turns would allow for continuous energy correction capability and allow for a shorter traffic pattern and hence final leg, but would require the FMS to accept course changes, while it is capturing a previously updated course. Alternatively, a direct interface to the autopilot could be implemented. For the current design, the simple autoland/FMS interface is deemed sufficient. For subsequent iterations, performance enhancements can be achieved by modifying the autoland/FMS interface if considered

Figure 2-6: Length and shape of the baseline trajectory
necessary.

2.1.6 Trajectory Planning using Energy Management

In order to determine the number and length of loops the aircraft follows in the holding pattern and to determine the length of the traffic pattern after the aircraft has exited the holding pattern, autoland requires knowledge of:

1. The amount of energy available to dissipate normalized by the aircraft weight $\Delta E/W$.

2. The rate of energy dissipation with ground distance travelled.

The aircraft energy at any point in time is defined as the sum of kinetic and potential energy:

$$E = mgh + \frac{1}{2}mV^2$$  \hspace{1cm} (2.2)

By normalizing the energy, $E$, by the aircraft weight, $W$, no knowledge of the exact aircraft mass is required to compute the numerical value of $E/W$.

$$E/W = h + \frac{V^2}{2g}$$  \hspace{1cm} (2.3)

Furthermore, the normalization by weight results in $E/W$ being expressed in units of altitude, which allows "energy" to be replaced by "altitude" whenever the airspeed does not change. The amount of energy available to dissipate, $\Delta E/W$, is simply the difference between the current energy, $E/W$, and the desired energy at touchdown, $E_{td}/W$.

$$\Delta E/W = E/W - E_{td}/W = (h_0 + \frac{V_0^2}{2g}) - (h_{td} + \frac{V_{td}^2}{2g})$$  \hspace{1cm} (2.4)

where $h_0$ and $V_0$ denote the current altitude and airspeed respectively, while $h_{td}$ and $V_{td}$ denote the touchdown zone elevation and desired airspeed at touchdown respectively.

While the aircraft follows the pattern, the airspeed is set to the best glide speed, $V_{bg}$,
which is 90 kt for the Cirrus SR-22. In order to reduce the descent rate before touchdown, autoland performs a flare maneuver that causes the airspeed to reduce from $V_{bg} = 90$ kt to $V_{td} = 75$ kt. Hence, if $V_0 = V_{bg}$, the energy available to dissipate simply becomes the difference between the altitude of the current location and the altitude of the touchdown zone, plus the difference in kinetic energy resulting from reducing the airspeed from 90 kt to 75 kt, which is equivalent to an additional 100 ft of altitude.

The rate of energy dissipation with ground distance travelled, $R$, can be derived by differentiating equation 2.3 with respect to $R$ and setting the airspeed, $V$, equal to the constant best glide speed, $V_{bg}$. This is justified even though the airspeed is changing from $V_{bg}$ to $V_{td}$ during the flare phase, since the flare is an out-of-trim maneuver, where the reduced altitude loss is compensated for by a reduction in airspeed. The total loss of energy over ground distance travelled during the flare maneuver, however, is the same as during the gliding approach phase. The energy dissipation rate with ground distance travelled can be written as:

$$\frac{d}{dR}(E/W) = \frac{dh}{dR} = \frac{1}{L/D}$$  \hspace{1cm} (2.5)

Due to the trajectory being planned for constant airspeed, the rate of change of energy normalized by weight is only dependent on the rate of altitude lost over ground distance travelled. This value is known as the aircraft’s $L/D$ or glide ratio. It is directly related to the flight path angle $\gamma$ by the relation: $\tan \gamma = \frac{1}{L/D}$. Here, it is assumed that the glide ratio and flight path angle are referenced to the inertial frame. If no wind is present, the glide ratio of the aircraft as measured with respect to the airmass and as measured with respect to the ground is equal. If wind is present, the aircraft travels a different distance over ground than it travels through the airmass. The convention adopted here, is that the ”glide ratio through the airmass” is adjusted for wind effects so that glide ratio always refers to the ratio of altitude lost, over ground distance travelled.

From knowledge of the wind speed and direction at the selected landing site, the course of the downwind leg, base leg and final leg, as well as the nominal bank angle during turns, autoland calculates the expected glide ratio on each segment of the holding pattern and the
traffic pattern. The segments of the holding pattern are: Turn to outbound leg, outbound leg, turn to inbound leg and inbound leg. The segments of the traffic pattern are: Downwind leg, Turn 1, Base leg, Turn 2 and Final leg, where Turn 1 represents the turn from the downwind leg to the base leg and Turn 2 represents the turn from base leg to final leg. The location of each leg is illustrated in Figure 2-7. The expected glide ratios on each of the traffic pattern

Figure 2-7: Naming convention for holding pattern legs and traffic pattern legs

segments are referred to by $L/D_d$, $L/D_{t1}$, $L/D_b$, $L/D_{t2}$ and $L/D_f$ in this order. The energy dissipated on the base leg segment for example becomes:

$$\Delta E_b/W = \frac{1}{L/D_b} \cdot L_B$$

(2.6)

where $L_B$ denotes the length of the base leg segment. The energy dissipated on the remaining segments can be calculated in a similar manner. Assuming constant glide ratio along each individual segment, energy dissipation along each segment is linear with ground distance travelled.

The complete "energy vs. range-to-go until the planned touchdown point" curve for the rectangular traffic pattern therefore becomes a piecewise linear, continuous function as shown in Figure 2-8. The currently available energy $E/W$ at the current range to go until the target

46
touchdown point, $R$, is shown on the right hand side of the the figure. The desired energy at touchdown, $E_{td}/W$, with zero range to go, is shown on the left hand side. The required length of the traffic pattern is found from matching the length of the downwind leg, base leg and final leg to the energy available to dissipate, $\Delta E/W = E/W - E_{td}/W$. The length of the turn segments are a result of the choice of the traffic pattern shape and length of the pattern legs and cannot be chosen freely for energy matching.

In the holding pattern, the length of the pattern is adjusted to match the energy difference between the current energy level $E/W$ and the energy required to fly the traffic pattern $E/W_{tp,0}$. $E/W_{tp,0}$ can be derived from knowledge of the length of the traffic pattern segments and the expected glide performance on each segment:

$$E/W_{tp,0} = \frac{L_D}{L/D_d} + \frac{r\pi}{2L/D_{t1}} + \frac{L_B - 2r}{L/D_b} + \frac{r\pi}{2L/D_{t2}} + \frac{L_F - r}{L/D_f}$$

(2.7)

The length of the holding pattern is adjusted by shortening or elongating the length of the inbound leg and outbound leg and number of loops in the pattern. Details on the adjustment
of the length of the holding pattern and traffic pattern to match energy targets are discussed in chapter 2.1.9.

2.1.7 Performance Prediction using Online Performance Estimation

In order to determine the energy dissipation with distance over ground flown for each traffic pattern segment, autoland requires information on the expected glide ratio of the aircraft on each segment. The Cirrus SR-22 manual states that the glide ratio at best glide speed is 9.6:1 with flaps retracted, which could potentially be used as an input for performance prediction. However, there is no guarantee that the aircraft will be in a clean configuration when autoland is initiated. Furthermore, the propeller could be windmilling or be stopped, which would make a large difference in observed glide ratio. Therefore, autoland monitors the aircraft’s performance as the aircraft glides at best glide speed and uses the deduced glide ratio for performance prediction on the following traffic pattern legs. For this purpose, autoland first determines an instantaneous glide ratio, $L/D_i$, from the change in energy, $\Delta E/W$, and ground distance travelled, $R$, within one sampling interval. Autoland deduces the change in energy from the change in altitude and airspeed, and deduces the distance travelled from the change in position as shown in Figure 2-9.

This instantaneous glide ratio contains the effect of wind and any non-zero bank angle, which both modify the measured glide ratio from the glide ratio that would be observed in still air with wings level. However, this no-wind, wings-level glide ratio is required for performance prediction on subsequent legs. Hence, any wind and bank angle effects are removed from the measured, instantaneous glide ratio, $L/D_i$, to yield $L/D_{i,0}$. The wind effect is broken up in crosswind and tailwind effect, with the tailwind having the strongest influence on the glide ratio. A 25% tailwind results in a 25% increase in glide ratio. The crosswind effect is much smaller but still yields a change in glide ratio of 3% for a 25% crosswind. Since glide ratio estimates are expected to be accurate to within 5%, this effect
needs to be included. In addition to wind, the glide ratio is affected by the aircraft’s bank angle during a turn. For any non-zero bank angle, the aircraft’s glide performance is reduced, since a part of the lift vector is now used to keep the aircraft in the circular motion. For a 30° bank angle, the glide ratio reduces by 20%, which clearly is a non-negligible effect.

Since the measurement of $L/D_i$, wind and bank angle are assumed to be noisy, autoland uses a simple moving average filter to smooth the samples of $L/D_{i,0}$ to arrive at the final output $L/D_0$. This $L/D_0$ can now be used to predict the performance of the aircraft on each of the subsequent pattern legs, $L/D_{exp}$. From knowledge of the course of the leg to be
flown, $\chi$ and the wind speed and direction, the crosswind and tailwind components can be calculated and their effect included from knowledge of the aircraft’s nominal airspeed $V_a$ and sink rate $V_{sink}$ as shown in Figure 2-10. For a detailed description of the equations involved in this algorithm, refer to the appendix, chapter A.2.2 and A.2.3.

![Diagram of glide ratio correction](image)

Figure 2-10: Calculation of the expected glide ratio, $L/D_{exp}$, from $L/D_0$

### 2.1.8 Aircraft Energy Monitoring and Energy Error Detection

As the aircraft is following the traffic pattern, autoland compares the aircraft’s energy and range to go until the planned touchdown point with the stored energy vs. range to go curve as was presented in Figure 2-8. If autoland determines that the energy is higher than the reference energy, the trajectory will need to be elongated to match the available excess energy. If the energy is lower, the trajectory will need to be shortened. Figure 2-11 illustrates this concept for the case that the aircraft has excess energy on the downwind leg. The red and blue boundaries are the excess energy and lack of energy tolerance bounds respectively, which are chosen in such a way that the aircraft is allowed to have 5% more energy than needed to reach the start of turn 2 and 3% less energy than need to reach the turn. These values were chosen as a tradeoff between noise tolerance and the desire to update as often as the FMS allows, to keep energy errors as small as possible. Noise added to the measurement of aircraft energy and range-to-go should not cause the measurement to cross the tolerance bounds and cause an update. Preliminary analysis shows that tolerance bounds of $+3\%$ and...
-3% meet this requirement. In addition to that, the operation of the FMS limits the frequency of trajectory updates to greater than every 5 s. For expected L/D estimation errors on the order of 10%, this consideration can be met by tolerance bounds of +2% and -1%. Hence, the tolerance bounds imposed by measurement noise become the active constraint. Since, there is little concern about the capability to correct a positive energy error, the upper bound was increased from the minimum of 3% to 5% to improve performance in a more noisy environment. For details on how the energy vs. range curve-to-go curve and its boundaries are stores is discussed in detail in the appendix, chapter A.2.9.

Using the trajectory updating strategies as outlined in chapter 2.1.9, the legs of the trajectory are elongated or shortened so as to match the available aircraft energy to the energy expected to be dissipated along the updated trajectory. The reason for the aircraft deviating off the expected energy vs. range to go curve is either due to the actual aircraft performance differing from the expected performance, or due to measurement errors of wind speed or direction, $V_w$ and $\chi_w$, or aircraft glide performance, $L/D_0$.

In order to generate a new trajectory that matches the actual aircraft performance, updated performance estimates will need to be used and the slopes of the reference energy vs. range-to-go curve adjusted to the actual performance observed. Therefore, autoland continuously runs the aircraft performance estimator and updates the reference energy vs.
range-to-go curve as soon as new estimates are received. This concept is illustrated in Figure 2-12. In this case, the aircraft performance estimator determined that the actual $L/D_0$ is lower than previously expected and the wind speed and direction has changed. This example is certainly exaggerated, but clearly shows the consequence. If the aircraft’s energy was right on the expected energy vs. range-to-go curve before the update, the aircraft’s energy would be well below the energy vs. range curve after the update.

This is one mechanism through which a trajectory update is triggered. The other mechanism is the aircraft falling below the energy vs. range curve without a difference in actual and expected performance having been detected. This scenario is possible for example, if the wind speed and direction from the wind estimator is in error.
2.1.9 Trajectory Updating

Trajectory updating in order to remove energy errors from an overestimation or underestimation of the expected glide ratio, $L/D_{\text{exp}}$, is performed while in the holding pattern, while on the downwind leg and while on the base leg. The current design does not provide energy error correction capability on the final leg, but absorbs energy errors on the final leg by appropriate placement of the target touchdown point. Since the autoland/FMS interface does not allow for trajectory updating while the FMS is capturing a previously updated course, autoland suspends trajectory updating for the duration of the course capture. This includes the instances where the FMS is capturing a new course after an unscheduled trajectory update and the scheduled transitions from downwind to base leg in turn 1 and base leg to final leg in turn 2.

Holding Pattern

The holding pattern flown while waiting for the power to be cut or while depleting excess energy consists of two half turns with variable length straight line segments in between. The goal of this function is to bring the aircraft to exactly the energy state required to fly the baseline length traffic pattern, $E/W_{tp,0}$. In order to achieve this, autoland adjusts the number of the turns and the length of the straight line segments to match the energy to be depleted, $\Delta E/W = E/W - E/W_{tp,0} - E/W_{id}$, where $E/W$ is the current aircraft energy. With this method, the length of the straight line segments will range from 0 to equal the length of a half turn. The holding pattern direction will be either right or left depending on the direction of the traffic pattern used for the final approach. The convention used here is that the aircraft first aligns with the course of the downwind leg at the initial point and then proceeds to turn away from the traffic pattern into the holding pattern as shown in Figure 2-13 for a left traffic pattern.

With this convention, the aircraft would perform right turns in the hold for a left traffic pattern and left turns in the hold for a right traffic pattern. The requirement to first
align with the downwind leg at the initial point is included to allow direct transition to the traffic pattern if the energy by the time of arrival to the initial point has dropped below the maximum allowable energy to start the traffic pattern as shown in the state transition diagrams in chapter 3.2.2. Due to the fact that the aircraft cannot deplete less than the energy required for one 360° turn with zero length straight line segments, \( E/W_{\text{loop}} \), there must be a range of energies at which the traffic pattern may be started. \( E/W \) can be calculated from knowledge of the turn radius, \( r \), and expected glide ratio within the turn \( L/D_t \). For a baseline glide ratio of 10:1, a 30° bank angle during the turn reduces the glide ratio by 20% to 8:1. Hence, the energy dissipated during one full turn becomes:

\[
E/W_{\text{loop}} = \frac{2r\pi}{L/D_t} = \frac{7,000 \, \text{ft}}{8} = 875 \, \text{ft}
\]  

(2.8)

To spread the distribution of energies at which the traffic pattern is started equally in both directions, the maximum allowable energy to start the traffic pattern is chosen as

---

Figure 2-13: Direction and sizing of holding pattern
\[ E/W_{tp,max} = E/W_{tp,0} + \frac{1}{2} \cdot E/W_{loop} + E/W_{td} \]. With this choice, the minimum energy at which the traffic pattern is started becomes \[ E/W_{tp,min} = E/W_{tp,0} - \frac{1}{2} \cdot E/W_{loop} + E/W_{td} \]. Hence, if the aircraft’s energy is just above the \( E/W_{tp,max} \) threshold on alignment with the downwind leg at the initial point, it will enter the holding pattern and fly one loop with zero length straight line segments. On passing the initial point the next time, its energy would have dropped to just above \( E/W_{tp,min} \) and it will exit the pattern. If the aircraft’s current energy is higher than \( E/W = E/W_{tp,0} + E/W_{loop} + E/W_{td} \), autoland adjusts the number of loops and length of the straight line segments to bring the aircraft to an energy state equal to \( E/W_{tp,0} \) by the time of passing the initial point after an integer number of loops according to the following method: The number of loops to be performed after completion of the current loop, \( n \), is found from the integer number of loops with zero straight line segments that can be performed with the current energy level \( E/W \).

\[
    n = \left\lfloor \frac{E/W - E/W_{tp,0} - E/W_{td} - E/W_{ip}}{E/W_{loop}} \right\rfloor \tag{2.9}
\]

where \( \lfloor x \rfloor \) is the greatest integer \( y \) such that \( y \leq x \) (rounding down). The term \( E/W_{ip} \) is the energy required to complete the current loop, which can be found from knowledge of the current location within the loop and the expected glide performance on the straight line segments and the turns.

Due to the discretization of energy levels by steps of magnitude \( E/W_{loop} \), there may be an excess of energy \( \Delta E/W \) of magnitude \([0, E/W_{loop}]\). This excess of energy is allocated to each loop by equal amounts and absorbed with equal amounts by each loop by elongating the straight line segments. \( \Delta E/W \) can be written as:

\[
    \Delta E/W = E/W - E/W_{tp,0} - E/W_{ip} - E/W_{td} - n \cdot E/W_{loop} \tag{2.10}
\]

The length of each straight line segment, \( L \), then becomes:

\[
    L = \frac{\Delta E/W}{n(L/D_d + L/D_f)} \tag{2.11}
\]
where \( L/D_d \) and \( L/D_f \) are the glide ratios expected on the inbound leg (equal to the course of the downwind leg) and the outbound leg (equal to the course of the final leg). As soon as \( n = 0 \), autoland waits until the aircraft has turned onto the inbound leg and signals the FMS to exit the hold. This causes the FMS to proceed to the downwind leg assuming it is already programmed.

**Downwind Leg**

While on the downwind leg, autoland elongates and shortens the trajectory as shown in Figure 2-14. The blue curve represents the planned trajectory, whereas the red curve represents the trajectory after the update.

In case of excess energy, autoland moves the base leg fix, \( F_B \), outwards along the course \( \chi_{c1} \), which is angled at 45° to the course of the baseline leg and downwind leg, as shown. At the same time, autoland moves the final leg fix, \( F_F \), away from the runway threshold in such a way that the course of the base leg is maintained. Similarly, for lack of energy, autoland moves \( F_B \) inwards along \( \chi_{c1} \) by the amount \( d \) and moves \( F_F \) towards the runway threshold along the course of the final leg by amount \( d \cdot \cos(45^\circ) \). The shift of the base leg parallel to the baseline base leg reduces the amount that \( F_B \) is required to move and hence reduces the course change from the downwind leg to the base leg compared to a scheme where only

![Figure 2-14: Trajectory updating method on the downwind leg](image-url)
the base lex fix, F_B is moved.

The value of $d$ for a given energy error $\Delta E/W$, can be found from knowledge of the distance to the base leg fix, $L$, the angle that $\chi_{c1}$ is making with the course of the downwind leg, $\alpha_p$, and the average glide ratio expected on the remaining legs, $L/D_{av}$. $\alpha_p$ will be 45° for the first update but will be different to this value after the first update. $\Delta E/W$, $L$ and $\alpha_p$ affect $d$ non-linearly and in such a way that an explicit expression for $d$ in terms of the former parameters cannot be found. Therefore, autoland uses lookup tables as described in more detail in the following section to retrieve $d$ from a given $\Delta E/W$, $L$ and $\alpha_p$ using interpolation between pre-computed values. $L/D_{av}$, however, acts as a simple linear scaling factor, which will be used to adjust the output from the lookup table. In summary, in order to find $d$, autoland first retrieves $d_0$, which specifies the magnitude and direction by which F_B would have to be moved if $L/D_{av} = 10$, using the lookup tables and subsequently adjusts $d_0$ for the actual $L/D_{av}$ to arrive at $d$:

$$d_0 = f(\Delta E/W, L, \alpha_p) \tag{2.12}$$

with

$$\alpha_p = \chi_{c1} - \chi_d \tag{2.13}$$

Then,

$$d = d_0 \cdot \frac{L/D_{av}}{10} \tag{2.14}$$

with

$$L/D_{av} = \frac{L(L/D)_d + L_B(L/D)_b + L_F(L/D)_f}{L + L_B + L_F} \tag{2.15}$$

**Base Leg**

While the aircraft is on the base leg, autoland uses a similar updating strategy to the method on the downwind leg. The updating strategy is shown in Figure 2-15. Since the final leg cannot be moved parallel to its current location, as was done with the base leg for trajectory
updating on the downwind leg, autoland needs to insert another leg into the traffic pattern trajectory in between the base leg and the final leg, which will be termed *final intercept leg*. The final intercept leg creates a new fix referred to as F\textsubscript{FI} which will be located at the intersection between the base leg and the new final intercept leg. The final leg fix is moved to a point 500 ft out from the runway threshold, which is the latest position, the final leg may be intercepted after an update on the base leg. 500 ft was chosen, since this is equivalent to the distance required to intercept the final leg from a 45° angle, which is the maximum course difference that can occur with the trajectory updating strategy on the base leg. This point will become the new final leg fix, F\textsubscript{F} and will remain at this position during all subsequent updates to the trajectory, while on the base leg. Only fix F\textsubscript{FI} will be moved inwards or outwards along course χ\textsubscript{c2}, which is angled at 45° to the final leg course.

As with the updating strategy on the downwind leg, the amount \( d \) that the final intercept fix F\textsubscript{FI} should be moved, depends on the distance to go until the fix, \( L \), the angle \( α_p \) that the aircraft course makes with course \( χ_{c2} \) and the average glide ratio expected on the remainder of the trajectory \( L/D_{av} \). In addition to that, \( d \) depends on the distance \( d_{total} \) between the initial location of the final intercept leg fix, F\textsubscript{FI,0}, and current location of F\textsubscript{FI} (i.e. before the update). This is due to the fact that the angle between \( χ_{c2} \) and the course of the final intercept leg, \( χ_{fi} \) does not remain constant over the region of trajectory updates as for the downwind leg but changes with change in \( d_{total} \). In Figure 2-15 \( d_{total} \) is equal to zero. Figure 2-16 shows a non-zero \( d_{total} \) for the case that the trajectory had previously been elongated (blue curve) and the next update intends to shorten the trajectory (red curve).
The distance $d$ by which $F_{FI}$ needs to be moved to remove energy error $\Delta E/W$ is

$$d_{total} = |F_{FI,0}, F_{FI}|$$

Figure 2-16: Example of trajectory updating on the base leg after first update

calculated in the same manner as for the downwind leg using look-up tables. The only difference is the added dimension of the look-up table due to the dependence on $d_{total}$.

$$d_0 = f(\Delta E/W, L, \alpha_p, d_{total}) \quad (2.16)$$

with

$$\alpha_p = \chi c2 - \chi b \quad (2.17)$$

Then,

$$d = d_0 \cdot \frac{L/D_{av}}{10} \quad (2.18)$$

with

$$\frac{L/D_{av}}{10} = \frac{L(L/D)_b + L_{FI}(L/D)_{fi}}{L + L_{FI}} \quad (2.19)$$

The length of the final intercept leg $L_{FI}$ depends only on the current location of the final intercept leg fix, $F_{FI}$. The location of the final leg fix $F_{FI}$ is fixed at 500 ft from the runway threshold. Hence, the total length of the final leg $L_F$ is 1,500 ft after adding the
displacement of the target touchdown point from the runway threshold as discussed in more detail in chapter 2.1.11.

Use of Lookup Tables for Trajectory Updating on the Downwind Leg and Base Leg

Since no explicit expression can be written for the amount $d$ that the leg fixes should be moved to correct a given energy error $\Delta E/W$ at any location along the trajectory and for any updates that have previously occurred, autoland uses pre-populated lookup tables to approximate $d$ for given $\Delta E/W$, $L$, $\alpha_p$, and $d_{total}$, if applicable. This approach was chosen over using any optimization function to solve the implicit equation on the fly, since the lookup table provides a more reliable and faster solution than an optimization function, which is of great importance when it comes to the final implementation of the system on an embedded processor. Furthermore, any level of accuracy can be achieved with the lookup table by reducing the step size, assuming memory is not restricted. Since the slopes of the functions $d_0 = f(\Delta E/W, L, \alpha_p)$ and $d_0 = f(\Delta E/W, L, \alpha_p, d_{total})$ change slowly, however, memory requirements for the lookup tables are expected to be limited.

Figure 2-17 shows a conceptual view of the implementation of a 3D lookup table for $d = f(\Delta E/W, L, \alpha_p)$. The values of $d$ for each combination of entries from vectorized $\Delta E/W$, $L$ and $\alpha_p$ are found by sweeping across the entries of one vector, while keeping the other two values constant. For constant $\alpha_p(1)$ and $L(1)$ for example, the values of $d$ for a vector of $\Delta E/W$ can be stored in a one-dimensional array. Continuing this process creates a three-dimensional array, whose entries can be imagined to be pinned to a point in 3D space. For equal spacing of the intervals in all three dimensions, the arrangement can be visualized by a square as shown in Figure 2-17. The point in space is defined by the x-, y-, and z-coordinates representing $\Delta E/W$, $L$ and $\alpha_p$ and the value at this point is $d$. In order to find $d$ for a combination of values of $\Delta E/W$, $L$ and $\alpha_p$ that falls in between the grid of stored values, autoland uses a simple linear interpolator to interpolate the function in 3D. A linear interpolator is considered sufficient due to the slow change in slope of function
Equations

\[ d = f(\Delta E/W, L, \alpha_p) \]

<table>
<thead>
<tr>
<th>( \alpha_p(1) )</th>
<th>( \Delta E/W(1) )</th>
<th>( \Delta E/W(2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>L(1)</td>
<td>( d_{11} )</td>
<td>( d_{12} )</td>
</tr>
<tr>
<td>L(2)</td>
<td>( d_{21} )</td>
<td>( d_{22} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \alpha_p(2) )</th>
<th>( \Delta E/W(1) )</th>
<th>( \Delta E/W(2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>L(1)</td>
<td>( d_{11} )</td>
<td>( d_{12} )</td>
</tr>
<tr>
<td>L(2)</td>
<td>( d_{21} )</td>
<td>( d_{22} )</td>
</tr>
</tbody>
</table>

Figure 2-17: Concept of a 3D lookup table for determination of \( d \) on downwind leg

\[ d = f(\Delta E/W, L, \alpha_p). \]

Final Leg

Up until the aircraft begins the turn onto the final leg, the trajectory can be updated using the strategies just presented. As soon as the aircraft begins to turn onto the final leg, however, there is no means to shorten the trajectory anymore, even though elongation of the trajectory would still be possible.

One means to counteract the problem of not being able to shorten the trajectory on the final leg would be to plan to turn onto the final leg with excess energy and depleting the excess energy through S-turns on the final leg by default if the aircraft performed nominally. A straight final leg would then only be flown if the glide performance is equal to the worst glide performance designed for. However, S-turns or other trajectory elongation strategies require a non-significant amount of length of the straight line final leg to perform, since the aircraft will need to be realigned with the course of the final leg. Assuming that the turn radius is fixed, as would be the case when interfacing to the existing autopilot without modification, a 5% elongation of the trajectory would require two turn radii of distance to
complete as shown in Figure 2-18. This is equivalent to correcting a 5% error in energy on

![Graph showing percent change in length of trajectory and amount of final leg length required for range of angles turned away from trajectory.]

Figure 2-18: Percent change in length of trajectory and amount of final leg length required for range of angles turned away from trajectory

the final leg.

With a nominal turn radius of 1,100 ft and a final leg length of 6,000 ft from the final leg intercept fix F_{FI0} to the target touchdown point, there is a maximum of three turn radii of final leg length available for S-turn maneuvers after subtracting the turn from the base leg to the final leg of 1,100 ft, the intercept margin of 500 ft and the margin for the touchdown point dispersion of 1,000 ft as shown in Figure 2-6. With three turn radii, a maximum of
15% of error in energy could be corrected, which would be sufficient for the energy errors expected. However, this energy error would need to be known by the start of the final leg. As the shape of the curve in Figure 2-18 suggests, the capability to correct energy errors decreases rapidly with decreasing distance available to perform the maneuver. Since it is assumed, that significant energy errors will not be visible at the start of the final leg but appear later, the S-turn maneuver is expected to have insufficient capability to correct energy errors and hence will not be included in the current design. However, it is a means to improve performance in a second iteration of the design, if deemed necessary.

The approach chosen instead, is to place the target touchdown point in a way that expected energy errors can be carried through until touchdown, without the aircraft landing short of the runway for the maximum negative energy error expected, or running off the end of the runway for the maximum positive energy error expected.

2.1.10 Placement of the Target Touchdown Point

The energy error correction strategies on the downwind leg and base leg aim to remove any energy error before the start of the turn onto the final leg. The assumption is therefore, that only energy errors accumulated during descent along the final leg will need to be absorbed by placement of the target touchdown point. Any energy error on the final leg is due to an error in estimating the expected glide ratio of the aircraft on the final leg, $L/D_F$. This error in estimation may either be due to an error in measuring the aircraft’s current glide performance, $L/D_0$, or due to an error in predicting wind shear during the descent along the final leg. Both error sources will lead to an error in expected glide ratio, $L/D_{err}$. This error will cause overshoot of the target touchdown point if the glide ratio was overestimated and will cause undershoot if the glide ratio was underestimated by the amount $\Delta d = L_F \cdot L/D_{err}$, where $L_F$ is the length of the final leg.
Overshoot or Undershoot due to Wind Shear in the Boundary Layer

At the start of the final leg, the aircraft is on the order of 500 ft - 800 ft off the ground depending on glide performance, which is within the region of the atmospheric boundary layer. Hence, the wind speed is expected to change from the measured wind speed at the start of the final leg until touchdown. While on the downwind leg and the base leg, the boundary layer effect would also be present. However, its effect is expected to be less pronounced. Since errors in glide performance estimation can be corrected on the downwind leg and the base leg, the boundary layer effect on the wind speed is ignored until the start of the final leg.

Figure 2-19 shows a sample wind profile derived from the Canadian Global Environmental Multiscale (GEM) model, which will be considered typical [23]. The change in wind direction will be ignored for simplicity and only the change in wind speed considered. The figure shows that the wind speed changes from 12 m/s (25 kt) at 300 m (1,000 ft) to zero on the ground. The strongest change in wind speed occurs below 600 ft, which suggests the importance of including its effect for glide performance prediction on the final leg. Above 600 ft, the effect is smaller and it is expected that resulting energy errors can be corrected via trajectory updating. In order to simplify the calculations, the wind profile in the boundary layer will be approximated by a linear model as shown in Figure 2-20. The wind profile is fixed by parameters $h_0$ and $V_w$, which indicate the start height of the wind shear and the wind speed at this height respectively. Using this model, an expected glide ratio can be calculated, assuming the aircraft descends through the simplified linear profile. The average glide ratio experienced under these conditions will be denoted by $L/D_{shear}$, which can be expressed in terms of the expected glide ratio experienced without the boundary layer effect present, $L/D_f$, the length of the final leg, $L_F$ and the amount of overshoot due to the change in wind speed, $\Delta d$. $\Delta d$ is positive, i.e. is an overshoot, for a reduction in headwind. $\Delta d$ is negative, i.e. is an undershoot, for a reduction in tailwind.

\[
\frac{L}{D_{shear}} = \frac{L}{D_f}(1 + \frac{\Delta d}{L_F})
\]

(2.20)
Figure 10: Change in wind speed and direction with altitude as per GEM model data

Figure 2-19: Example of change in wind speed, S and direction, dir with altitude in the atmospheric boundary layer

The magnitude of $\Delta d$ can be calculated as follows. The wind speed $V_w$ at any altitude $h$ above the ground can be calculated from knowledge of the wind speed $V_{w,0}$ at altitude $h_0$ at
the start of the final leg.

\[ V_w(h) = \frac{V_{w,0}}{h_0} \cdot h \]  

(2.21)

The change in distance travelled over ground due to this wind shear can be found from:

\[ \Delta d = \int_0^{t(h=0)} V_w(t)dt \]  

(2.22)

Changing the integration variable to \( h \) and plugging in the equation for \( V_w(h) \) yields:

\[ \Delta d = \int_0^{h_0} \frac{1}{V_a \sin \gamma h_0} \cdot V_w h \cdot dh \quad \text{and} \quad dt = \frac{dh}{V_a \sin \gamma} \]  

(2.23)

Integrating and considering that:

\[ \sin \gamma = (L/D)^{-1} \quad L/D_f = \frac{L_F}{h_0} \]  

(2.24)

results in

\[ \Delta d = \frac{1}{2} \cdot L_F \cdot \frac{V_{w,0}}{V_a} \]  

(2.25)

Equation 2.25 shows that the magnitude of overshoot or undershoot due to a linear wind profile is linearly proportional to the length of the final leg, linearly proportional to the change in wind speed from \( h_0 \) to 0, \( V_{w,0} \) and inversely proportional to the airspeed \( V_a \).

Inserting this result into equation 2.20 gives the simple result:

\[ L/D_{\text{shear}} = L/D_f \cdot \left(1 + \frac{V_{w,0}}{2V_a}\right) \]  

(2.26)

Autoland will assume that when measuring wind speed \( V_{w,0} \) at the start of the final leg, that this wind speed will reduce to zero by the time of touchdown and assume that the average glide performance to be expected on the final leg is \( L/D_{\text{shear}} \). In case the wind speed does not reduce by this amount, the aircraft would undershoot the target touchdown point for an approach into a headwind or would overshoot the target touchdown point for an approach.
into a tailwind. The amount of overshoot or undershoot can be calculated from the error in $L/D_{exp}$ made by assuming a linear wind profile. From equation 2.26, it can be seen that the $L/D_{exp}$ adjustment due to the wind profile is given by $\frac{V_{w,0}}{2V_a}$. The error in $L/D_{exp}$ is therefore given by the magnitude of the overestimation of change in wind speed. If the wind was expected to reduce by magnitude $V_{w,0}$, but only reduces by magnitude $\Delta V_w$, the difference $V_{w,0} - \Delta V_w$ contributes to an error in glide performance estimation of $L/D_{err} = \frac{V_{w,0} - \Delta V_w}{2V_a}$.

Figure 2-21 shows the amount of $L/D$ error accrued for a range of $\Delta V_w$ if the assumed reduction in wind speed was $V_{w,0}$. Figure 2-21 assumes the wind speed at the start of the final leg is 7 m/s. This wind speed would be experienced for a final leg length of 6,000 ft, glide ratio of 10 and wind speed at 1,000 ft of 12 m/s as shown by the wind model data in Figure 2-19. If the wind speed then only reduced by 1 m/s, the error in $L/D_{exp}$ resulting from making this assumption is 6.8%.

![Figure 2-21: L/D error resulting from assuming linear reduction in wind speed of $V_{w,0}$ when wind reduces only by $\Delta V_w$](image)

Figure 2-21: L/D error resulting from assuming linear reduction in wind speed of $V_{w,0}$ when wind reduces only by $\Delta V_w$. 

67
Overshoot or Undershoot of the Target Touchdown Point due to Measurement Errors

In addition to errors in predicting boundary layer wind shear, errors in estimating the glide performance on the final leg, $L/D_f$, may originate from an error in the measurement of $L/D_0$, or in the measurement of the wind speed and direction. It is assumed that these measurement errors contribute less than 5% of error to the estimation of $L/D_f$. Figure 2-22 shows the amount of overshoot or undershoot, $\Delta d$ that results from overestimating or underestimating the expected $L/D_f$ by the percentage error $L/D_{err}$.

![Graph showing the relationship between error in estimating L/D and touchdown point dispersion](image)

Figure 2-22: Overshoot or undershoot from error in estimating expected L/D

**Selected Location of the Target Touchdown Point**

For the error in predicting boundary layer wind, a maximum error of 6.5% will be assumed and for the measurement of current glide performance, a maximum error of 5%. From
Figure 2-22, this is equivalent to overshooting or undershooting the target touchdown point by 400 ft and 300 ft respectively. Since it is assumed that the error in measurement of glide performance can be both positive or negative, margin for touchdown point dispersion is required in both directions. For the error in wind prediction, only undershoot margin is required for the headwind case and only overshoot margin is required for the tailwind case. Furthermore, bilateral margin for touchdown point dispersion is required for the expected GPS error of +/- 2 m. Assuming a flight path angle during the flare maneuver of $-1.5^\circ$ as described in chapter 2.1.11, this results in a touchdown point dispersion of +/- 300 ft. Figure 2-23 illustrates the placement of the target touchdown point for the headwind case. For the tailwind case, the 400 ft allocated to wind shear would be applied to overshoot instead of undershoot of the target touchdown point.

With this choice, the target touchdown point will be placed 1,000 ft from the runway threshold in the headwind case, in order to ensure that the aircraft touches down at the runway threshold at the earliest, if all error sources were at their extreme minimum. The
latest touchdown would occur 1,600 ft from the runway threshold if all error sources were at their extreme maximum. For the tailwind case, the target touchdown point is placed 600 ft from the runway threshold with the latest touchdown occurring also after 1,600 ft.

2.1.11 Flare

Autoland performs a flare maneuver similar to the one implemented by autoland systems for commercial aircraft, in order to reduce the aircraft’s vertical speed on the final leg to a vertical speed that is acceptable for touchdown. For an airspeed of 90 kt and an assumed range of glide ratios from 13:1 to 7:1, the aircraft’s vertical speed ranges from 10 ft/s to 20 ft/s, which is above the 10 ft/s threshold, that is considered to cause damage to the aircraft and potentially severe injuries to passengers, if a landing was attempted at this vertical speed [18]. The recommended touchdown speed ranges from 1 ft/s to 6 ft/s for this class of aircraft. In order to provide a smooth decrease in vertical speed from the start of the flare until touchdown, autoland uses the following exponential equation, that is shaped by parameters \( v_{td,0} \) and \( \tau \).

\[
v(t) = (v_0 - v_{td})e^{-\frac{t}{\tau}} + v_{td}
\]

where \( v_0 \) represents the vertical speed at the start of the flare, \( v_{td} \) the target touchdown speed at the expected touchdown point, and \( \tau \) the time constant at which the vertical speed command is decreased. \( v_{td} \) and \( \tau \) are chosen in such a way, that the vertical speed at the earliest and latest expected touchdown point due to GPS error remains within the 1 ft/s to 6 ft/s range and that the airspeed remains above the stall speed even for the latest, expected touchdown due to maximum negative GPS error, at the worst glide ratio. The onset of a stall is specified to occur at 73 kt without flaps for the Cirrus SR-22 [8]. For values \( \tau = 1 \) s and \( v_{td} = 3.3 \) ft/s, satisfactory performance could be achieved. With this choice, the flare height becomes 70 ft, which is more than twice the typical flare height of 30 ft for this size of aircraft. However, this is necessary due to the fact that the aircraft is losing altitude twice as fast as on a standard 3° glideslope, which autoland compensates for by starting the flare
at a higher altitude.

With a vertical speed of 3.3 ft/s and an airspeed of 75 kt, the flight path angle becomes 1.5°, which is used to estimate the amount of overshoot and undershoot of the target touchdown point given the uncertainty in vertical position due to GPS error. Figure 2-24 illustrates the aircraft’s flight path during the flare maneuver for the maximum positive and maximum negative expected GPS error. For a GPS error of +/- 6.6 ft (2 m), the overshoot and undershoot of the target touchdown point becomes +/- 300 ft.

Figure 2-24: Flare maneuver under uncertainty in vertical position

and undershoot of the target touchdown point becomes +/- 300 ft.

2.2 Autoland - FMS Interface

2.2.1 Trajectory Specification

The trajectory generated by autoland consists of a set of straight line segments for the path to the initial point and for the traffic pattern and constant radius semi-circles plus straight line segments for the holding pattern. Figure 2-25 shows a sample autoland trajectory in the lateral plane with the path to the initial point consisting of several, straight line segments. Even though both the path to the initial point and the traffic pattern can be represented by a set of straight line segments, the fact that the traffic pattern trajectory is updated on a frequent basis, while the path to the initial point remains constant once generated, requires that the two trajectories are communicated to the FMS differently.
The path planner, that generates the trajectory to the initial point, constructs the path as a set of waypoints as shown in Figure 2-25 by waypoint P1-P4, with P1 being the position of the aircraft at the point of autoland initiation and P4 being the initial point of the traffic patten trajectory. Hence, it is convenient to specify the trajectory to the FMS as a set of waypoints. For this purpose, so-called track-to-a-fix (TF) legs are used. TF legs require the specification of two waypoints to fully specify the leg [5]. A complete trajectory is synthesized by joining as many TF legs as needed. As long as the trajectories built from TF legs are gap-free, meaning that two subsequent legs have a waypoint in common, the FMS follows the trajectory without further input required by autoland.
By definition, the FMS will perform a "fly-by" waypoint, i.e. cutting the corner, if the course change of the trajectory from one leg to the next is less than 135°, and performs a "fly-over" maneuver, if the course change is greater than 135°. For exact calculation of the length of the trajectory to the initial point, this would need to be taken into account, but it is assumed that course changes are small and few, so that their effect on the length of trajectory can be neglected.

For the traffic pattern, TF legs are not practical to use, since updates to the current flight plan cause a very abrupt reaction, as the guidance function is reset during the re-computation and capture of the new path. For frequent trajectory updating while in the traffic pattern, fix-to-a-manual-termination (FM) legs are more suitable. These legs start at a given fix and are defined by a course that is followed until the operator manually terminates the current leg and passes a new one [5]. The manual input by the operator is replaced by automatic action of the autoland software. The fixes are the Initial Point for the downwind leg, F_B for the base leg and F_F for the final leg. In order to ensure smooth transition from the current, active leg to the next leg, autoland needs to ensure, that it passes the new leg at a distance from the next FM leg fix that is sufficient for capturing the new course without overshooting the following leg. For 90° turns, as is the case for the baseline traffic pattern, the next leg needs to be injected into the FMS guidance function one turn radius before reaching the next waypoint.

Even though not strictly enforced for FM legs, the convention that the "fly-by" waypoint maneuver is only performed for course changes smaller than 135° is adopted here to limit the amount of corner cutting. For a course change of exactly 135°, the next leg would have to be injected 2.4 times the turn radius before the next waypoint and rapidly increases with increasing course change from this point. The lead distance to the next waypoint, $t_d$, required for a smooth intercept of the next leg, as a function of change in course, $\Delta \chi$, from one leg to the next, is given by the following function:

$$t_d = r \sqrt{\frac{1 + \cos \gamma}{1 - \cos \gamma}}$$

(2.28)
where $\gamma = 180^\circ - \Delta \chi$.

Finally, the holding pattern consisting of two half-circles and variable length straight line segments, is commanded to the FMS using the *hold-to-a-manual termination (HM)* leg. The HM leg requires specification of the direction of the turn, length of the straight line segment, the location of a hold fix, which is set to the initial point, and specification of the course of the outbound leg, which is equal to the course of the final leg, $\chi_f$ [5]. When following the HM leg, the FMS will keep the aircraft in the holding pattern until autoland passes a new leg. A smooth transition from the holding pattern to the downwind leg is ensured, as long as autoland does not inject the downwind leg into the FMS before the aircraft has turned onto the inbound leg of the holding pattern, which is aligned with the course of the downwind leg.

In addition to the path to the initial point, the holding pattern and traffic pattern, which are part of the nominal autoland trajectory, autoland includes the "Fail" trajectory that consists of a straight line trajectory to an area that is marked as unpopulated on the aeronautical charts. For this fail trajectory, autoland uses the *Direct-to-a-fix (DF)* leg that only requires specification of the destination waypoint [5]. The FMS then computes the shortest path to this fix, the "great circle distance", and guides the aircraft along this path.

### 2.2.2 Use of Local and Global Coordinate Systems

In order to simplify the calculations involved in trajectory generation and updating, autoland uses a local, cartesian North-East coordinate system with the origin located at the runway threshold, and the runway heading aligned with the positive y-axis, the North axis. The left graphic in Figure 2-26 illustrates this coordinate system for a left traffic pattern. Specification of the direction of the traffic pattern plus knowledge of the length of the baseline traffic pattern is then sufficient to specify the location of the downwind leg fix, base leg fix and final leg fix as well as the course of each leg within the local coordinate frame. Using this baseline traffic pattern, the trajectory updating functions can easily derive the location of
the new leg fixes and new leg course within the local coordinate frame. Conversion to the
global coordinate frame is achieved by first rotating the traffic pattern trajectory by an
amount equal to the runway heading, and second, linearly translating each leg fix by the
latitude and longitude of the touchdown zone. The location of each leg fix in the North-East
coordinate frame after rotation is calculated using the common rotation formula adjusted for
the difference in convention of positive and negative rotations. Here, a clock-wise rotation
is considered positive.

\[
\begin{bmatrix}
P'_{north} \\
P'_{east}
\end{bmatrix} =
\begin{bmatrix}
\cos(\chi_r) & -\sin(\chi_r) \\
\sin(\chi_r) & \cos(\chi_r)
\end{bmatrix}
\begin{bmatrix}
P_{north} \\
P_{east}
\end{bmatrix}
\]

The right graphic in Figure 2-26 shows the traffic pattern after the rotation by runway
heading, \(\chi_r\). The angle, \(\chi_r\), is negative in this picture and hence produces a counter-clockwise
rotation. By definition, the autoland trajectory generation and updating functions use angles
within range \([-\pi, \pi]\), which requires that the runway heading, \(\chi_r\), returned from the FMS
database is adjusted to this range before performing any calculations with it. \(\chi_r\) is expected
to be expressed in \textit{true north}. Should \(\chi_r\) be stored in the FMS with respect to \textit{magnetic
north}, autoland expects that the correction for magnetic variation is performed internally to
the FMS and the \(\chi_r\) returned is expressed in \textit{true north}.

The latitude and longitude of each leg fix is calculated from the latitude and longitude
of the runway threshold, \(\mu_0\) and \(\iota_0\), and the displacement of the leg fixes, IP, F_B and F_F
from the origin expressed in latitude and longitude coordinates, \(\Delta \mu\) and \(\Delta \iota\). A variety
of formulas with various degrees of accuracy exist to perform this calculation. The simplest
assumes the earth is a perfect sphere with radius \(R = 6,378,137\, m\). From the definition of
latitude and longitude on the sphere, \(\Delta \mu\) and \(\Delta \iota\) can be written in terms of \(\Delta N\) and \(\Delta E\) as
follows:

\[
\Delta \mu = \arctan\left(\frac{1}{R}\right)\Delta N \\
\Delta \iota = \arctan\left(\frac{1}{R \cos(\mu)}\right)\Delta E
\]
Figure 2-26: Local North-East coordinate system before and after rotation by amount equal to runway heading

where \( \mu \) specifies the current latitude.

At 45° latitude, the error in neglecting the flattening of the earth towards the poles, results in an error in estimating \( \Delta \mu \) and \( \Delta \iota \) of less than 0.3%, which would cause an error of less than 15 ft when specifying the coordinates of the fixes for a 5,000 ft long traffic pattern leg. This is deemed acceptable. However, the error can easily be removed by using a more accurate model of the earth.

Using this information, the location of the leg fixes in global latitude and longitude coordinates becomes:

\[
\mu = \mu_0 + \Delta \mu \\
\iota = \iota_0 + \Delta \iota
\]

The resulting leg coordinates can now be used for specification of the leg fixes to the FMS. The course of each leg is deduced easiest while still in the North-East coordinate frame and is calculated in a straightforward manner by adding the runway heading to the course of each leg referenced to the local North-East coordinate system. The FMS expects the course to be specified with respect to true north within range \([0, 2\pi[\), which requires adjustment from the \((-\pi, \pi]\) range used for the autoland trajectory generation and updating functions.
2.3 User Operations

2.3.1 Autoland Initiation

As was shown in Figure 2-1, autoland is envisioned to be initiated by pressing the corresponding button. This task may be performed by a passenger or the partially incapacitated pilot. For a fully incapacitated pilot, the system would have to automatically recognize that the pilot is in distress. This automatic engagement could be achieved using a hypoxia sensor that requests the pilot to acknowledge he/she is still alert if the sensor detects blood oxygen levels below the set threshold. Autoland would be engaged automatically, if no response is received within a given time limit. For a first version of the autoland system, this design proposes to use button press initiation.

2.3.2 Increase Throttle Command

After autoland has detected the engine status, there are two cases, in which autoland would request the operator to increase the throttle. The command would be given audibly and visually. In the first case, autoland has detected an engine out scenario, and was unable to find a runway within range. The aim in this case is to avoid an unnecessary off-field landing in a situation where the throttle was set below the engine out threshold inadvertently or due to a planned, steep descent. Second, autoland will request a throttle increase if the engine power is below the threshold to fly straight and level but above the threshold to land with the power-off method. In this case, the aircraft is stuck in a situation where it cannot hold altitude and also cannot land using the autoland final approach scheme. In this situation, autoland holds the wings level and aims the aircraft at the closest, unpopulated area within range, while it is waiting for the operator to increase the throttle setting. Figure 2-27 shows a conceptual view of this command.
2.3.3 Status Display for Selected Landing Site and Planned Trajectory

After a landing site with corresponding feasible trajectory has been determined, autoland illustrates the planned trajectory and target landing site on the moving map display as shown conceptually in Figure 2-28. The aircraft’s movement along the trajectory will be illustrated on the display in order to keep the operator informed over the progress.
2.3.4 Prepare to Land Tutorial

While the aircraft is following the trajectory to the initial point, autoland will provide the operator with information on what to expect from the autoland process and what tasks they will need to perform. The tasks are communicated via the screen with graphics of the appropriate control input to provide. The first task will be to reduce the throttle to zero upon arrival at the initial point, if it is not already zero. Only if this task applies, will it be communicated to the operator. Upon touchdown, the pilot or passenger will be requested to apply brake pressure in order to assist with the deceleration of the aircraft and keeping the aircraft aligned with the runway centerline. The location and operation of the brakes will be illustrated to the operator prior to starting the final approach.

2.3.5 Request to Reduce Throttle at Initial Point

As soon as the aircraft has arrived at the initial point, autoland requests the operator to reduce the throttle to zero, if it is not zero already. The command is given via audible and visual cues. A conceptual sketch of the visual cue is shown in Figure 2-29.

![Retard throttle](image)

Figure 2-29: Conceptual visual cue for operator to reduce throttle to zero
2.3.6 Request to Apply Brakes

Once the aircraft has touched down, autoland requests the operator to apply brake pressure according to the tutorial shown while flying to the initial point. The command is given via audible and via visual cues.
Chapter 3

System Architecture

The system architecture was derived by decomposing the functionality required to complete a full autoland mission into individual modules, where each module is responsible for one specific task only. A full autoland mission consists of selection of a suitable landing site, trajectory planning to the initial point, tracking of the trajectory to the initial point, generating, tracking and updating the holding pattern and generating, tracking and updating the traffic pattern.

Following the decomposition of these functional blocks into a set of tasks, 21 modules were identified that are each responsible for one specific task. These modules are grouped into seven components, which are arranged into three layers.

Modules within the same component perform either similar tasks or are each contributing a part to perform the overall component functionality. Layers represent the level of the autoland task, with planning tasks being at the top of the hierarchy and trajectory tracking tasks at the bottom.

The execution of functions and individual modules is controlled by two state machines running in parallel. Based on the functionality implemented by the components and modules, and the information required by the state machines, external and internal interfaces are derived.
3.1 Static Decomposition

3.1.1 Autoland Layers

Autoland uses three layers Destination Planning, Trajectory Planning and Trajectory Translation as shown in Figure 3-2. The Destination Planning layer is responsible for selecting the most suitable landing site given the engine status and achievable range. For this purpose, the Destination Planning layer interfaces to the FMS databases to query runway information such as location, length and heading, emergency landing site information and terrain data.

The core of the autoland functionality is provided by a Trajectory Planning layer that determines the individual legs of the autoland trajectory and updates the trajectory using the energy management strategies presented in chapter 2.1.9.

The Trajectory Translation layer implements the communication of the trajectory to the FMS in such a way that the existing trajectory tracking capability of the FMS can be used without modification. This includes determining the correct timing for passing a new trajectory or a trajectory update to the FMS and translation of the trajectory into a format that is supported by the FMS.
In addition to that, a user interface layer would be required for the final implementation. The user interface layer will not be treated in further detail in this thesis, but is an integral part of the final design, which will require a more thorough treatment from a human factors perspective than the brief overview presented in chapter 2.3.

### 3.1.2 Autoland Components

The three layers are further decomposed into components as shown in Figure 3-2.

![Autoland system architecture](image)

**Figure 3-2: Autoland system architecture**

**Components of Layer Destination Planning**

The components of layer Destination Planning are *Engine Status Detector* and *Landing Site Selector*. The Engine Status Detector determines whether the engine is operational, has failed or whether the throttle setting is at an intermediate setting. This information is used by the Landing Site Selector to determine which search criteria to apply and how to determine the available range. In case the engine is operational, the range is limited by the
available fuel. For case engine out and engine status indeterminate, the range is limited by the amount of ground range covered over altitude lost.

Components of Layer Trajectory Planning

Layer Trajectory Planning is implemented by components *Trajectory Generator*, *Performance Estimator*, *Energy Manager* and *Trajectory Updater*. The Trajectory Generator is responsible for creating a new trajectory, either a trajectory to the initial point, the initial holding pattern or an initial traffic pattern. As the aircraft follows this initial trajectory, the Performance Estimator monitors the aircraft’s glide ratio and calculates predictions of the aircraft’s glide performance on the subsequent trajectory segments. The Energy Manager maintains information on the aircraft’s target energy for the current location and passes the deviation off the energy target to the Trajectory Updater, if tolerance limits are exceeded. The Trajectory Updater then shortens or elongates the initial trajectory in order to remove the energy error.

Components of Layer Trajectory Translation

Trajectory translation is achieved by components *FMS Interface* and *Autopilot Interface*. The FMS Interface is responsible for translating the trajectory returned by the trajectory planning layer, which is defined in a local, cartesian coordinate system into a global, latitude and longitude system that can be interpreted by the FMS. Furthermore, the FMS Interface ensures correct timing of communicating the trajectory updates to the FMS, so that the aircraft performs smooth transitions from one leg of the trajectory to the next. The Autopilot Interface sets the autopilot mode depending on the autoland phase. For example, autoland uses altitude hold while tracking to the initial point (if the engine is operational) and airspeed hold along the traffic pattern.
3.1.3 Autoland Modules

Finally, the components are divided into several modules in order to implement the component functionality as shown in Figure 3-3.

![Autoland system architecture](image)

**Figure 3-3: Autoland system architecture**

**Modules of Component Landing Site Selector**

The modules of component Landing Site Selector are the Range Calculator, Touchdown Energy Calculator, Initial Point Calculator and Landing Site Prioritizer. The Range Calculator first determines how far the aircraft can go given engine status and the current altitude, i.e. energy. The Landing Site Prioritizer then searches for landing sites within this range. For the solutions returned, component Landing Site Selector checks whether the candidate landing sites are reachable given the altitude of the touchdown zone, using the Touchdown Energy Calculator. Upon passing this test, component Landing Site Selector calculates the location of the initial point using the Initial Point Calculator. Subsequently, the Landing Site Selector attempts to find an obstacle-free trajectory to the initial point that meets the
range constraints, using the Path Planner to the Initial Point of component Trajectory Generator/Updator. If such a trajectory could be found, the Landing Site Selector selects the landing site and passes the runway heading, location of the runway threshold, the location of the initial point and traffic pattern direction to the FMS Interface component of layer Trajectory Translation.

**Modules of Component Performance Estimator**

The Performance Estimator consists of modules *Energy Calculator, L/D_0 Estimator* and *L/D_Expected Generator*, and runs whenever the aircraft cannot hold altitude, i.e. in case engine out or case engine status "Indeterminate". The Energy Calculator computes the current energy from altitude and airspeed and the change in energy between sampling intervals. This sampled change in energy is used by the *L/D_0 Estimator* to calculate an instantaneous, and subsequently, averaged $L/D_0$ under consideration of the current wind conditions and bank angle. The same wind information along with the $L/D_0$ is used by the *L/D_Expected Generator* to calculate the expected glide ratio, $L/D_{exp}$, on the following trajectory segments.

**Modules of component Energy Manager**

The energy management function is implemented by modules *Feasibility Monitor, Excess Energy Calculator, Energy Curve Manager* and *Traffic Pattern Energy Error Detector*. The Feasibility Monitor runs while the aircraft is tracking the trajectory to the initial point and monitors whether the planned destination is still reachable given the current estimation of $L/D_0$. If the maximum achievable range becomes less than the remaining distance to go, the destination is deemed infeasible.

Once the aircraft enters the holding pattern, the Excess Energy Calculator computes the amount of excess energy the aircraft has over the desired energy to start the traffic pattern. From knowledge of the current position in the holding pattern provided by the FMS Interface component if layer Trajectory Translation, the Excess Energy Calculator
deduces the amount of excess energy the aircraft will have when next passing the initial point, which will be used to adjust the length of subsequent loops using the functions of the Trajectory Generator/Updator component.

Upon generation of the traffic pattern trajectory, the Energy Curve Manager constructs and maintains the energy vs. range-to-go until touchdown curve from knowledge of energy dissipation with ground distance travelled, $L/D_{\text{exp}}$, and the length of the traffic pattern segments, $L_i$.

The Traffic Pattern Energy Error Detector then uses the mathematical description of this curve and its tolerance bounds plus knowledge of distance to go until the start of the next traffic pattern segment, in order to determine whether an energy error exists. If the current aircraft energy exceeds the tolerance bounds, the Energy Error Detector passes the energy error on to the Trajectory Generator/Updator component for trajectory adjustment.

**Modules of Component Trajectory Generator/Updator**

The Trajectory Generator/Updator functionality is implemented by modules *Path Planner to Initial Point*, *Holding Pattern Trajectory Generator/Updator*, *Traffic Pattern Trajectory Generator* and *Traffic Pattern Trajectory Updator*.

The Path Planner to the Initial Point is active during the landing site selection process in order to determine an obstacle-free path to the initial point at the candidate landing site. The trajectory is planned in such a way that the aircraft is aligned with the course of the downwind leg by the time the aircraft reaches the initial point. This allows smooth transition to either the holding pattern or traffic pattern depending on the available aircraft energy by the time the aircraft reaches the initial point.

Upon arrival at the initial point, the Holding Pattern Trajectory Generator/Updator creates a holding pattern with the number and length of hold loops that matches the energy to dissipate. The holding pattern update process operates in exactly the same way as the holding pattern generation process, since the Excess Energy Calculator always returns the expected excess energy on passing the initial point the next time, and not the total excess
energy. This was done so that only subsequent hold loops are updated, but not the current hold loop. This is to conform to the way the FMS accepts HM legs to track a holding pattern. This choice clearly allows non-zero energy error to exist upon exit the holding pattern.

The Traffic Pattern Trajectory Generator takes the current aircraft energy upon exit of the holding pattern, $E/W$, the expected glide ratio $L/D_{exp}$, on each traffic pattern leg and the length of the baseline traffic pattern trajectory, and adjusts the length of the baseline traffic pattern trajectory to match the energy to dissipate until touchdown using the trajectory updating strategy on the downwind leg as presented in chapter 2.1.9. This traffic pattern is specified in the local, North-East coordinate frame and translated to global latitude and longitude coordinates by the FMS Interface component, for tracking by the FMS.

While the aircraft is tracking the trajectory, the Traffic Pattern Trajectory Updater removes energy errors by using the updating strategy that corresponds to the current leg. The Trajectory Updater receives information about the current leg from the the Traffic Pattern Follower and switches updating strategies accordingly. No updating is performed during the turns from downwind leg to base leg and from base leg to final leg and on the final leg itself.

**Modules of Component FMS Interface**

The FMS Interface uses modules *Trajectory to Initial Point Follower, Holding Pattern Follower* and *Traffic Pattern Follower* in order to pass on the trajectory generated by components Path Planner to the Initial Point, Holding Pattern Trajectory Generator/Updater and Traffic Pattern Trajectory Generator/Updater to the FMS during the corresponding phases of the autoland trajectory.

The Trajectory to the Initial Point Follower translates the trajectory returned by the Path Planner to the Initial Point in local North-East coordinates to global coordinates and passes this trajectory to the FMS as a set of TF legs.

While in the holding pattern, the Holding Pattern Follower passes the holding pattern to the FMS as HM legs by specifying the holding pattern fix, the course of the outbound leg, the turn direction and the length of the straight line segments. The Holding Pattern
Follower is also responsible for signaling the FMS to exit the hold at the correct point in time, so that a smooth transition to the downwind leg is performed.

While in the traffic pattern, the Traffic Pattern Follower passes the traffic pattern trajectory to the FMS as a set of FM legs. The Traffic Pattern Follower passes the legs one at a time and at a distance from the intersection of two legs that allows a smooth transition from one leg to the next. Furthermore, the Traffic Pattern Follower inhibits trajectory updates passed by the Traffic Pattern Updator, while the FMS is capturing a previously updated leg.

**Modules of Component Autoilot Interface**

The Autopilot Interface consists of modules *Autopilot Mode Select* and *Flare Controller*. Autopilot Mode Select runs continuously and sets the autopilot modes airspeed hold, altitude hold, and ”Straight and Level” as determined by the two state machines discussed in chapter 3.2. The Flare Controller executes during the autoland flare phase and determines the target vertical speed based on the aircraft’s altitude above the touchdown zone elevation.

**3.2 Autoland Modes and Phases**

Autoland uses two state machines to control which of the autoland components and modules are executing under which circumstances and during which part of the autoland process. One of the state machines controls the autoland *mode*, which is directly related to the engine status. The other state machine controls the autoland *phases* which is related to the progress of the aircraft along the autoland trajectory.

**3.2.1 Modes**

Depending on the engine state, autoland selects either autopilot mode ”Altitude Hold” or ”Airspeed Hold” and turns the performance estimation functionality on or off. Figure 3-4
illustrates the autoland modes and the mode transitions.

Upon initiation, autoland sets autopilot mode "Straight and Level", in order to abort any maneuvering that might have been executed prior to autoland engagement.

If the Engine Status Detector has determined that the engine has failed, autoland enters mode "Engine Out" and sets the autopilot mode to "Airspeed Hold" in order to hold best glide speed, while the aircraft is following the autoland trajectory at power off. Furthermore, in mode "Engine Out" autoland runs the Performance Estimator to determine the glide capability of the aircraft, which is used determine feasibility of the planned trajectory to the initial point and to correct energy errors by adjusting the length of the trajectory in the hold and the traffic pattern.

Figure 3-4: Autolnd modes as determined by engine status
If the engine power is sufficient for straight and level flight, autoland enters mode "Engine Operational" and requests the autopilot to hold altitude in order to maintain altitude up until reaching the initial point. The Performance Estimator cannot collect information on the glide performance of the aircraft, while the aircraft maintains altitude and is hence not active in this mode.

If the engine state is "Indeterminate", autoland does not modify the autopilot mode from the already set "Straight and Level" mode, but runs the Performance Estimator to obtain an approximation to the current rate of altitude lost per unit of ground distance travelled, which in turn is used to determine the aircraft’s available range. The state machine provides for return from mode engine "Indeterminate", upon change of the engine status. This transition would occur, once the operator followed the command to increase throttle, as given by default if the engine status is classified as "Indeterminate".

3.2.2 Phases

During a nominal automatic landing, autoland traverses phases "Destination Selection", "Approach to Initial Point", "Hold at Initial Point", "Follow Traffic Pattern", "Flare" and "Rollout" in this order. The corresponding state machine is shown in Figure 3-5. This state machine runs in parallel with the mode control state machine of the previous chapter.

It is possible to skip phase "Hold at Initial Point", if upon arrival at the initial point, the energy is smaller than the maximum energy allowed to start the traffic pattern and the power is already off. Besides the nominal path, autoland may go into state "Fail" in the unlikely event that no feasible landing site could be found, neither a runway nor a suitable emergency landing site nor an unpopulated area. This could occur if the aircraft experienced an engine failure at low altitude above a large metropolitan area. Autoland provides for the possibility to leave the "Fail" state and re-attempt a search for a landing site in the event that the engine status has changed to "Operational".

Furthermore, engine state "Indeterminate", which cannot be handled by the nominal
autoland approach procedure is captured in phase "Direct-to unpopulated area" if a reachable unpopulated area has been found. In engine status "Indeterminate", autoland does not attempt to find a runway but rather directs the aircraft straight to an area where it is expected to encounter the least man-made obstacles and where it is expected to cause the least damage.
Figure 3-5: Autoland phases determining the lateral trajectory followed by the aircraft
3.3 Autoland Interfaces

3.3.1 External Interfaces

In order to perform the functions outlined above, autoland relies on the existing FMS to provide information from the navigation database and to track a specified trajectory. Furthermore, it relies on the autopilot to provide aircraft state information, such as airspeed and altitude and to provide limited control of the aircraft’s motion in the vertical plane through autopilot modes ”Altitude Hold” and ”Airspeed Hold”. In addition to that, autoland expects to be able to receive weather forecasts from satellite radio. Figure 3-6 shows the interfaces to the FMS and the autopilot by autoland component that uses it. For a detailed view on which module uses an external input or generates an external output refer to chapter 3.3.4.

On the other end, autoland interfaces to the operator using the display capability of the integrated flight deck and using audible and visual cues as described in chapter 2.3. However,
this chapter focuses only on the autoland interface to the autopilot and the FMS. The exact external interface to the operator will be determined upon completion of the detailed user interface design, which is left for future work.

The Engine Status Detector requires the use of external input % power to determine the engine status. The Performance Estimator uses the wind speed and direction, as well as the aircraft state, state, estimated by the autopilot to determine the current aircraft glide ratio and the glide ratio to be expected on subsequent trajectory legs. The FMS Interface infers the aircraft’s current position from the aircraft state and determines the trajectory to pass to the FMS to track. While flying to the initial point, the FMS Interface specifies the track as a set of TF legs. In the holding pattern, the FMS Interface uses HM legs and in the traffic pattern, FM legs. The DF leg is only used for the ”Fail” scenario where autoland guides the aircraft directly to an unpopulated area. The Autopilot Interface simply switches between autopilot modes ”Airspeed Hold” and ”Altitude Hold” and passes the target airspeed if needed. The Energy Manager uses the current position of the aircraft to determine the aircraft’s location on the energy vs range-to-go curve in order to deduce potential energy errors. The Landing Site Selector uses a number of inputs from the FMS’ navigation database, such as runway info, emergency landing site info and image files of aeronautical charts in order to compare and select the most suitable landing site. It uses aircraft state information and fuel range to limit the landing site search region. Furthermore, the Landing Site Selector uses measurements of wind speed and direction for the airport vicinity to determine the runway direction that yields a headwind on approach. This data is available from METARs (Meteorological Terminal Aviation Routine Weather Report ) received via satellite radio. Table 3.1 summarizes the autoland external interfaces to the FMS and the autopilot.
<table>
<thead>
<tr>
<th>Ident</th>
<th>SubIdent</th>
<th>Direction</th>
<th>Frequency</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%power</td>
<td>-</td>
<td>In</td>
<td>1 Hz</td>
<td>Estimate of engine power output</td>
</tr>
<tr>
<td>wind</td>
<td>-</td>
<td>In</td>
<td>1 Hz</td>
<td>Estimate of wind speed and direction from autopilot</td>
</tr>
<tr>
<td>loc</td>
<td>-</td>
<td>In</td>
<td>1 Hz</td>
<td>Aircraft’s location in global latitude and longitude coordinates</td>
</tr>
<tr>
<td>hdg</td>
<td>-</td>
<td>In</td>
<td>1 Hz</td>
<td>Aircraft heading with respect to true north</td>
</tr>
<tr>
<td>course</td>
<td>-</td>
<td>In</td>
<td>1 Hz</td>
<td>Direction of the aircraft’s ground track with respect to true north</td>
</tr>
<tr>
<td>V_a</td>
<td>-</td>
<td>In</td>
<td>1 Hz</td>
<td>Airspeed in kt</td>
</tr>
<tr>
<td>vspeed</td>
<td>-</td>
<td>In</td>
<td>1 Hz</td>
<td>Aircraft’s current sink rate referenced to the inertial frame</td>
</tr>
<tr>
<td>h</td>
<td>-</td>
<td>In</td>
<td>1 Hz</td>
<td>Estimate of the aircraft altitude above mean sea level (MSL)</td>
</tr>
<tr>
<td>bank</td>
<td>-</td>
<td>In</td>
<td>1 Hz</td>
<td>Aircraft bank angle</td>
</tr>
<tr>
<td>TF legs</td>
<td>-</td>
<td>Out</td>
<td>triggered</td>
<td>Trajectory specified as a sequence of &quot;Track-to-Fix&quot; legs</td>
</tr>
<tr>
<td>DF leg</td>
<td>-</td>
<td>Out</td>
<td>triggered</td>
<td>&quot;Direct-to-Waypoint&quot; leg</td>
</tr>
<tr>
<td>HM leg</td>
<td>-</td>
<td>Out</td>
<td>triggered</td>
<td>&quot;Hold with Manual Termination&quot; type holding pattern</td>
</tr>
<tr>
<td>FM leg</td>
<td>-</td>
<td>Out</td>
<td>triggered</td>
<td>&quot;Fix to Manual Termination&quot; leg</td>
</tr>
<tr>
<td>mode select</td>
<td>-</td>
<td>Out</td>
<td>triggered</td>
<td>Integer identifier specifying the autopilot mode</td>
</tr>
<tr>
<td>airspeed_cmd</td>
<td>-</td>
<td>Out</td>
<td>triggered</td>
<td>Commanded target airspeed in knots</td>
</tr>
<tr>
<td>name</td>
<td>name</td>
<td>In</td>
<td>triggered</td>
<td>Name of the emergency landing site</td>
</tr>
<tr>
<td>loc</td>
<td>loc</td>
<td>In</td>
<td>triggered</td>
<td>Latitude and longitude of emergency landing site</td>
</tr>
<tr>
<td>hdg</td>
<td>hdg</td>
<td>In</td>
<td>triggered</td>
<td>Recommended approach direction with respect to true north</td>
</tr>
<tr>
<td>elevation</td>
<td>elevation</td>
<td>In</td>
<td>triggered</td>
<td>Elevation of the landing site above MSL</td>
</tr>
<tr>
<td>aero charts</td>
<td>-</td>
<td>In</td>
<td>triggered</td>
<td>Color Image files of aeronautical charts</td>
</tr>
<tr>
<td>fuel range</td>
<td>-</td>
<td>In</td>
<td>triggered</td>
<td>Range in nm calculated from amount of initial fuel and fuel flow</td>
</tr>
<tr>
<td>name</td>
<td>name</td>
<td>In</td>
<td>triggered</td>
<td>String identifier containing the name of the airport and runway</td>
</tr>
<tr>
<td>loc</td>
<td>loc</td>
<td>In</td>
<td>triggered</td>
<td>Latitude and longitude of the location of the runway threshold</td>
</tr>
<tr>
<td>hdg</td>
<td>hdg</td>
<td>In</td>
<td>triggered</td>
<td>Runway heading with respect to true north</td>
</tr>
<tr>
<td>length</td>
<td>length</td>
<td>In</td>
<td>triggered</td>
<td>Length of the runway including displaced threshold in ft</td>
</tr>
<tr>
<td>dispThres</td>
<td>dispThres</td>
<td>In</td>
<td>triggered</td>
<td>Length of runway absorbed by displaced threshold</td>
</tr>
<tr>
<td>elevation</td>
<td>elevation</td>
<td>In</td>
<td>triggered</td>
<td>Elevation of the runway above MSL</td>
</tr>
<tr>
<td>METAR</td>
<td>-</td>
<td>In</td>
<td>0.5-1 hr</td>
<td>Aviation weather report including wind speed and direction for candidate landing sites</td>
</tr>
</tbody>
</table>

Table 3.1: Autoland external interfaces
3.3.2 Internal Interfaces

The autoland internal interfaces are used to pass data from one autoland component to another. Figure 3-7 shows the component interfaces that are active while the aircraft is tracking the traffic pattern.

The Trajectory Generator/Updator publishes the trajectory in structure ”TP” (Traffic Pattern), which consists of an array of structure ”leg” for each leg of the traffic pattern, TP.leg[], an array of glide ratios TP.L/D.exp[] for each segment of the traffic pattern, TP.L/D.exp[], an update flag, TP.updateFlag, and single values specifying the distance to go until the start of the next segment, TP.R, distance to go until the start of the next leg TP.L and the number of the traffic pattern segment the aircraft is currently on TP.currSeg. The array TP.leg[] contains the fix of each traffic pattern leg, TP.leg[].fix, the course of each leg, TP.leg[].course, and the length of each leg TP.leg[].length.

The Performance Estimator uses the information on the course of the traffic pattern legs and the course change from one leg to the next to determine the expected glide ratios on each segment of the traffic pattern, TP.L/D.exp. The expected glide ratio and the length of each traffic pattern leg is then used by the Energy Manager to construct the energy vs. range-to-go curve, that is used to determine whether an energy errors exists.

The FMS Interface uses the leg fixes and courses to synthesize FM legs that it passes to the FMS one at a time. The FMS Interface keeps track of which segment the aircraft is currently on and publishes this to the Energy Manger in TP.currSeg. As the aircraft follows the FM legs, the FMS Interface calculates the range to go until the start of the next segment, TP.R.

The Energy Manger then uses TP.currSeg and TP.R to locate the aircraft’s position on the energy vs. range-to-go curve. The Energy Manager thus determines whether an energy error, $\Delta E/W$, exists and passes the error on to the Trajectory Generator/Updator, if energy error tolerance limits are exceeded.

The Trajectory Generator/Updator then generates a new trajectory using, $\Delta E/W$, TP.currSeg and TP.L, to generate a new trajectory that removes the energy error. TP.L is required by
the trajectory updating functions, since the distance that the leg fix has to be moved to correct for the same energy error, increases with decreasing distance from the leg fix. After the Trajectory Generator/Updator has published the updated trajectory, the process repeats.

Adding the data that is passed while tracking the trajectory to the initial point and

Figure 3-7: Autoland component internal interfaces active during phase ”Follow Traffic Pattern”

while in the holding pattern yields the interface diagram in Figure 3-8. The major addition are the outputs from component Landing Site Selector that passes the location of the runway threshold, \(LS.loc\), the runway heading, \(LS.hdg\), the pattern direction, \(LS.dir\), the location of the initial point, \(LS.IP\), and the desired energy at touchdown, \(LS.E_{td}\), that has been deduced from the touchdown zone elevation. This information is bundled in structure \(LS\).

In order for the Trajectory Generator/Updator to attempt and create a feasible trajectory to the selected landing site, the Landing Site Selector passes the available range, which the
Trajectory Generator/Updator compares against the length of the best feasible trajectory it could find. If the range is larger than the length of the trajectory, the Trajectory Generator/Updator signals the Landing Site Selector that the candidate landing site is feasible through flag $LS.LS.Ok$. The added trajectory to the initial point and the holding pattern are specified in struc-
tures $TI.leg[]$ and $HP$ respectively. The Energy Manager uses knowledge of which leg of the trajectory to the initial point the aircraft is currently on, $TI.currSeg$, the distance to go until the next waypoint of the trajectory, $TI.L$ and the length of the overall trajectory, $TI.length$, to monitor whether the trajectory is still feasible. If not, it signals the Landing Site Selector using flag $LS.FP.Ok$. Zero wind speed is assumed for the trajectory to the initial point. Therefore, the feasibility monitor and the Trajectory Generator/Updater use output, $L/D.0$ from the Performance Estimator.

The Holding Pattern, $HP$, is defined by the location of the initial point, $HP.fix$, the turn direction, $HP.dir$, the course of the outbound leg, $HP.course$, the length of the straight line segments, $HP.length$, and the number of turns to still complete after the current loop, $HP.#loops$. The direction of the holding pattern is opposite of the direction of the traffic pattern, $LS.dir$, and the course of the outbound leg is equal to the runway heading, $LS.hdg$. The length of the straight line segment, $HP.L$, and the number of loops, $HP.#loops$, is supplied by the Trajectory Generator/Updater.

The FMS Interface calculates the range to go until the next segment in the holding pattern, i.e. start of the turn, start of the outbound leg, $HP.R$, which the Energy Manager uses to estimate the energy error difference between the aircraft energy at the next pass of the initial point and the desired energy to start the pattern. This energy error is then used by the Trajectory Generator/Updater to update the number of loops to remain in the holding pattern and the length of the straight line segments. Table 3.2 summarizes the autoland internal component interfaces.
<table>
<thead>
<tr>
<th>Ident</th>
<th>SubIdent</th>
<th>Origin</th>
<th>Destination</th>
<th>Frequency</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>engine status</td>
<td>-</td>
<td>ESD</td>
<td>API, LSS</td>
<td>1 Hz</td>
<td>One of the following: &quot;Out&quot;=0, &quot;Operational&quot;=1, &quot;Indeterminate&quot;=2</td>
</tr>
<tr>
<td>range</td>
<td>-</td>
<td>LSS</td>
<td>TGU</td>
<td>1 Hz</td>
<td>Achievable range from L/D_0 and available energy including 30% margin</td>
</tr>
<tr>
<td>name</td>
<td>LSS</td>
<td>HMI</td>
<td>triggered</td>
<td>String identifier containing the name of the selected landing site for future display on the user interface</td>
<td></td>
</tr>
<tr>
<td>bdg</td>
<td>LSS</td>
<td>FMI, PE</td>
<td>triggered</td>
<td>Runway heading, i.e. approach direction, with respect to true north</td>
<td></td>
</tr>
<tr>
<td>loc</td>
<td>LSS</td>
<td>FMI</td>
<td>triggered</td>
<td>Latitude and longitude of the location of the runway threshold or location of the earliest desired touchdown point for alternate landing site</td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td>LSS</td>
<td>TGU, FMI</td>
<td>triggered</td>
<td>Location of the initial point in local North-East coordinates</td>
<td></td>
</tr>
<tr>
<td>dir</td>
<td>LSS</td>
<td>FMI</td>
<td>triggered</td>
<td>Traffic pattern direction, either &quot;Right = 1&quot; or &quot;Left = -1&quot;</td>
<td></td>
</tr>
<tr>
<td>E,td</td>
<td>LSS</td>
<td>EM</td>
<td>triggered</td>
<td>Desired touchdown energy normalized by weight from touchdown zone elevation</td>
<td></td>
</tr>
<tr>
<td>LS_OK</td>
<td>TGU</td>
<td>LSS</td>
<td>triggered</td>
<td>Flag indicating whether a trajectory to the candidate landing site could be found within the available range</td>
<td></td>
</tr>
<tr>
<td>FP_OK</td>
<td>EM</td>
<td>LSS</td>
<td>triggered</td>
<td>Flag indicating whether the current flight plan is still feasible given the performance of the aircraft</td>
<td></td>
</tr>
<tr>
<td>E/W</td>
<td>-</td>
<td>PE</td>
<td>LSS, TGU, EM</td>
<td>1 Hz</td>
<td>Aircraft energy normalized by weight</td>
</tr>
<tr>
<td>L/D_0</td>
<td>-</td>
<td>PE</td>
<td>LSS, TGU, EM</td>
<td>1 Hz</td>
<td>No-wind, wings-level glide ratio estimate</td>
</tr>
<tr>
<td>leg[]_fix</td>
<td>TGU</td>
<td>FMI</td>
<td>triggered</td>
<td>Array containing the waypoints that specify the trajectory to the initial point in local North-East coordinates</td>
<td></td>
</tr>
<tr>
<td>leg[]_length</td>
<td>TGU</td>
<td>EM</td>
<td>triggered</td>
<td>Array containing the length of each leg of the trajectory to the initial point</td>
<td></td>
</tr>
<tr>
<td>currSeg</td>
<td>FMI</td>
<td>EM</td>
<td>triggered</td>
<td>Counter indicating the current leg of the trajectory to the initial point the aircraft is on (1 = first leg, ... n = nth leg)</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>FMI</td>
<td>EM</td>
<td>triggered</td>
<td>Distance to go until the next leg fix</td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>TGU</td>
<td>FMI</td>
<td>triggered</td>
<td>Length of the straight line segment of the holding pattern</td>
<td></td>
</tr>
<tr>
<td>nLoops</td>
<td>TGU</td>
<td>EM</td>
<td>triggered</td>
<td>Remaining number of loops after completion of the current loop</td>
<td></td>
</tr>
<tr>
<td>fix</td>
<td>TGU</td>
<td>EM</td>
<td>triggered</td>
<td>Location of the holding pattern fix in global coordinates</td>
<td></td>
</tr>
<tr>
<td>course</td>
<td>TGU</td>
<td>EM</td>
<td>triggered</td>
<td>Course of the outbound leg global coordinates</td>
<td></td>
</tr>
<tr>
<td>dir</td>
<td>TGU</td>
<td>EM</td>
<td>triggered</td>
<td>Holding pattern direction, either &quot;Right = 1&quot; or &quot;Left = -1&quot;</td>
<td></td>
</tr>
<tr>
<td>currSeg</td>
<td>FMI</td>
<td>EM</td>
<td>triggered</td>
<td>Counter indicating the current segment of the holding pattern the aircraft is on (1=turn to outbound leg, 2= outbound leg, 3= turn to inbound leg, 4= inbound leg)</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>FMI</td>
<td>EM</td>
<td>triggered</td>
<td>Distance to go until the start of the next segment</td>
<td></td>
</tr>
<tr>
<td>L/D_exp[]</td>
<td>PE</td>
<td>TGU, EM</td>
<td>triggered</td>
<td>Expected glide ratio on the turn segment and the straight line segments</td>
<td></td>
</tr>
<tr>
<td>leg[]_fix</td>
<td>TGU</td>
<td>FMI</td>
<td>triggered</td>
<td>Array of trajectory legs specified as set of waypoints in local North-East coordinates</td>
<td></td>
</tr>
<tr>
<td>leg[]_course</td>
<td>TGU</td>
<td>FMI</td>
<td>triggered</td>
<td>Course of each leg in local North-East coordinate system</td>
<td></td>
</tr>
<tr>
<td>leg[]_length</td>
<td>TGU</td>
<td>EM</td>
<td>triggered</td>
<td>Length of each traffic pattern segment</td>
<td></td>
</tr>
<tr>
<td>currSeg</td>
<td>FMI</td>
<td>TGU, EM</td>
<td>triggered</td>
<td>Counter indicating the current segment of the traffic pattern trajectory the aircraft is on (1 = downturn leg, 2 = turn to base, 3 = base leg, 4 = turn to final intercept leg (if applicable), 5 = final intercept leg (if applicable), 6 = turn to final leg, 7 = final leg)</td>
<td></td>
</tr>
<tr>
<td>L/D_exp[]</td>
<td>PE</td>
<td>TGU, EM</td>
<td>triggered</td>
<td>Expected glide ratio on each segment of the traffic pattern</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>FMI</td>
<td>EM</td>
<td>triggered</td>
<td>Distance to go until the start of the next segment</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>FMI</td>
<td>TGU</td>
<td>triggered</td>
<td>Distance to go until the next leg fix</td>
<td></td>
</tr>
<tr>
<td>updateFlag</td>
<td>TGU</td>
<td>FMI</td>
<td>triggered</td>
<td>Flag indicating that a trajectory update has occurred</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Autoland internal interfaces
The glide ratio stated in the Pilot Operating Handbook for an idling engine. For the SR-22, this is 9.6:1. This value will be set to 10:1 to simplify calculations.

Length of a single leg of the traffic pattern. Legs are chosen to be of equal length, so only one constant is needed. Note: The total length of the final leg is 1,000 ft longer than the downwind leg and the base leg, but these 1,000 ft constitute the undershoot margin which places the target touchdown point 1,000 ft past the start of the touchdown zone. The length of the final leg up until the he start of the touchdown zone is exactly 5,000 ft, the same as the downwind leg and the base leg.

Distance consumed by the undershoot margin. Determines the position of the target touchdown point past the runway threshold.

Value increased from the baseline 22.5° that the current autopilot uses to reduce proportion of the in the traffic pattern spent on capturing course changes.

Assumed best glide speed at which the glide ratio of 10:1 can be achieved.

Turn radius achieved at \( V_{glide} \) and \( \phi_0 \)

Length of the baseline traffic pattern

Desired airspeed at touchdown

Desired sink speed upon touchdown

Desired runway length that is deemed sufficient to perform autoland maneuver if pilot is operating the brakes

Margin applied to calculation of the achievable glide range

Table 3.3: Autoland design constants

### 3.3.3 Design Constants

In addition to the data carried by the internal and external interfaces mentioned before, the autoland modules use a number of design constants that do not change from one execution of autoland to the next. The design constants are chosen so as to match the performance of the Cirrus SR-22. However, they can be adjusted to match another airplane. Since the design constants here were chosen with the Cirrus SR-22 in mind, care must be taken in adjusting not just a single design constant, but to scale other affected constants appropriately as well.

An example would be the length of the baseline traffic pattern legs. If a larger aircraft with a larger turn radius is to be used, the length of the baseline traffic pattern legs would need to be increased to maintain the same energy error correction capability while in the traffic pattern. Table 3.3 summarizes the design constants to be used for the Cirrus SR-22.
3.3.4 Complete System Diagram

The full system design as described in the previous chapters, is summarized in Figure 3-9. This graphic shows the individual autoland modules along with their internal and external interfaces. Inputs that are derived from design constants, i.e. that are the same for each execution of autoland, are highlighted in magenta. Inputs that are constant after selection of a landing site are shown in blue. Interface lines that change value over the course of the autoland execution are drawn in black. Each module is placed in a colored background to highlight its affiliation with one component. The exception is module "Engine Status Detector", since it is module and component at the same time. Furthermore, the box color and box frame of each module was chosen to highlight their point of execution within the autoland process.
Figure 3-9: Full system diagram
Chapter 4

Simulation Environment

In order to assess the performance achievable with the proposed design of the automatic landing system, a prototype of the Trajectory Planning layer and the Trajectory Translation layer for autoland phase "Follow Traffic Pattern" was implemented in Matlab. Two separate aircraft models are used to assess this performance. A simple point mass model augmented with a trajectory tracking controller is used to evaluate the performance of autoland in terms of touchdown point dispersion. A high fidelity, 6 degree of freedom, rigid body model of the Cirrus SR-22, supplied by Avidyne, Inc., is used to evaluate autoland’s performance in terms of flare performance. Flare performance includes the sink rate on touchdown, attitude on touchdown and tolerance to GPS/WAAS vertical position error.

4.1 Aircraft Model for Evaluation of Touchdown Point Dispersion

The simulation model for evaluation of touchdown point dispersion uses a point mass model to simulate the aircraft, and a trajectory tracking controller to imitate the trajectory tracking capability of the FMS. This model of the aircraft, autopilot and FMS then runs in parallel
to a model of the Performance Estimator, and in parallel to a model of the Energy Manager and the Trajectory Updator, each in a separate thread. Both the Performance Estimator and the Energy Manager/Trajectory Updator combination are executed once every second of simulated flight time of the aircraft. Communication between the aircraft/autopilot/FMS model and the Energy Manager/Trajectory Updator model is achieved via the simulated FMS Interface component of the Trajectory Translation layer. At each simulation step, the FMS Interface component publishes the aircraft’s progress along the traffic pattern, which is then used by the Energy Manager/Trajectory Updator model to determine whether an energy error exists and hence, whether a trajectory update is required. If a trajectory update has been performed, the FMS Interface component passes the new trajectory to the aircraft/autopilot/FMS model for tracking whenever the aircraft is in a state that permits trajectory updating, i.e. while on the downwind leg or base leg, and while no previous updates are being captured.

### 4.1.1 Point Mass Model

The dynamic model for evaluation of touchdown point dispersion consists of a point-mass, which is free to move in three dimensions. Lateral motion is determined by the aircraft’s heading, which is set via yaw rate commands from the trajectory tracking controller. No roll or pitch orientation is defined and hence, their value is set to zero. Motion in the vertical plane is determined by the glide performance of the aircraft in still air, $L/D_0$, and the prevailing wind. The wind speed is referred to by $V_w$ and the wind direction by $\chi_w$. The aircraft state consists of 3D position $p$ referenced to the inertial frame, the linear velocity $v_b$ referenced to the body frame, the aircraft orientation with respect to the inertial frame $O$, which is determined by the aircraft heading $\psi$, and the angular acceleration $W$ determined
by the heading rate $\psi$. The state variables can be written as:

\[
p = \begin{bmatrix}
p_{\text{north}} \\
p_{\text{east}} \\
p_{\text{down}}
\end{bmatrix}
\]

\[
v_b = \begin{bmatrix}
V_a \cos \gamma \\
0 \\
V_a \sin \gamma
\end{bmatrix}
\]

where $V_a$ is the aircraft’s airspeed and $\gamma = \arctan(L/D_0^{-1})$, the aircraft’s flight path angle referenced to the inertial frame.

\[
O = \begin{bmatrix}
0 \\
0 \\
\psi
\end{bmatrix}
\]

\[
W = \begin{bmatrix}
0 \\
0 \\
\dot{\psi}
\end{bmatrix}
\]

The rate of change of each state variable is determined by the following equations:

\[
\dot{p} = \begin{bmatrix}
\cos \psi & -\sin \psi & 0 \\
\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix} v_b + \begin{bmatrix}
V_w \cos \chi_w \\
V_w \sin \chi_w \\
0
\end{bmatrix}
\]

where $V_w$ and $\chi_w$ are provided by the wind estimator. Since the aircraft is in a trimmed flight condition, there are no linear body accelerations:

\[
\dot{v}_b = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
\]
By definition,
\[ \dot{O} = W \]  
(4.1)

Due to trimmed flight, there are also no angular body accelerations:
\[ \dot{W} = \begin{bmatrix} 
0 \\ 
0 \\ 
0 
\end{bmatrix} \]

With this choice, the aircraft course with respect to the in the inertial frame can be found from:
\[ \chi = \arctan(\frac{\dot{p}_{east}}{\dot{p}_{north}}) \]  
(4.2)

Using this simple point mass model, the flare maneuver cannot be simulated, since the model has no knowledge of lift and drag, and hence reduction in sink rate and airspeed during a pitch up maneuver cannot be imitated. However, it is advantageous to use this model for evaluation of the touchdown point dispersion, since it greatly simplifies the simulation and allows evaluating the system performance without error sources added from a complete aircraft/FMS/autopilot model. In order to demonstrate the location of the touchdown point with the simple point mass model only, the point mass simulation stops at the flare altitude (70 ft) and positions the touchdown point at the distance from the start of the flare, required by the nominal flare maneuver, plus any additional distance covered during the flare due to non-zero wind. The nominal horizontal distance occupied by the flare maneuver is 1,450 ft. The duration of the maneuver is approximately 9 s. Hence, a 5 kt headwind, for example, would reduce the horizontal distance covered during the flare by 76 ft from 1,450 ft to 1374 ft.
4.1.2 Trajectory Tracking Controller

A trajectory tracking controller is added on top of the point mass model in order to follow the prescribed trajectory. This controller monitors the distance to go until the next waypoint provided by the FMS Interface component and initiates a constant turn rate to intercept the next leg once the distance to go to the next waypoint is equal to the amount of turn anticipation required to intercept the next leg. In the presence of crosswind, the trajectory tracking controller commands turn rate corrections to adjust the aircraft heading so that the aircraft crabs into the wind and uses the horizontal component between its airspeed vector and the ground speed vector to counteract the wind. The trajectory tracking controller uses non-linear feedback on crosstrack error using the arctan function similar to the controller implemented on Stanford’s Stanley, the Stanford Racing Teams entry in the DARPA Grand Challenge 2005, to determine a desired turn rate, $\dot{\psi}_{\text{ref}}$ [12]. The advantage of this feedback controller is that it saturates the feedback gain towards $+\pi$ and $-\pi$ as the crosstrack error increases to infinity, while maintaining close to linear feedback for small crosstrack errors. Naturally, the trajectory tracking control is implemented in a different manner in several control loops on the actual aircraft, but its behavior is assumed to be imitated well with this single control law. The crosstrack error, $e$, is found by taking the shortest distance from the current position to the desired track and adjusting the sign of the crosstrack error according to the turn direction required to intercept the track. With this convention, the crosstrack error is negative when the aircraft is located to the right of the track, whereas the crosstrack error is positive when the aircraft is to the left of the track. In addition to tracking lines, the trajectory tracking controller ensures that the turn segments are tracked in inertial space, i.e. it adjusts the turn rate, if wind attempts to blow the aircraft off of the constant radius turn. The control law is given by:

$$\dot{\psi}_{\text{ref}} = -k_1 \cdot \arctan\left(\frac{k_2 \cdot e}{V_a}\right) + \dot{\psi}_{\text{track}}$$  \hspace{1cm} (4.3)
where $\dot{\psi}_{\text{track}}$ specifies the rate of change of the course of the trajectory to follow. For a straight line, this value is zero. For a constant radius turn, as would be the case on the the turn segments of the traffic pattern, the turn rate becomes:

$$
\dot{\psi}_{\text{track}} = \frac{g \cdot \tan \phi}{V_a}
$$

(4.4)

Gains $k_1$ and $k_2$ where chosen as 0.5 and 1 for straight line segments respectively, and 1 and 3 for turn segments. With the relatively large gain $k_1$, it is possible to reach the maximum limit on the turn rate of the aircraft. The maximum limit is chosen as the turn rate achievable at a 45° bank angle at the current airspeed. For a nominal airspeed of 90 kt, the turn rate limit becomes $0.22 \text{ rad/s}$. The controller-aircraft interface is simulated in such a way that a given turn rate command is followed exactly and without delay. This simplification results in a better trajectory tracking performance in simulation, than achievable by the actual aircraft. Any inaccuracy and delay in the aircraft response, which results in the trajectory flown being shorter or longer than the planned trajectory, results in the aircraft landing long or short of the target touchdown point. Energy errors ensuing from deviations off the planned trajectory before the intercept to the final leg can be corrected for by trajectory updating. Any deviations off the planned trajectory after the intercept to the final leg, however, will lead to overshoot or undershoot of the target touchdown point by the amount the planned trajectory differs from the actual trajectory flown. For the purpose of this evaluation, it is assumed that the touchdown point dispersion due to tracking error is negligible compared to the effect of GPS error and L/D estimation error. However, this will need to be verified with the actual aircraft.
4.2 Aircraft Model for Evaluation of Flare Performance

4.2.1 High-fidelity Aircraft Model

For evaluation autoland’s touchdown performance with the proposed Flare Controller, a high fidelity, 6 degree of freedom, rigid body simulation of the Cirrus SR-22, provided by Avidyne, Inc., was used. This model experiences the aerodynamic forces of lift and drag, propulsive forces, and gravitational forces. Its motion is controlled via elevator, aileron and rudder deflection as well as throttle input. The characteristics of the aircraft are captured via non-dimensional stability and control coefficients. The numerical values of these coefficients is property of Avidyne, Inc. and are therefore not provided in this description. The coefficients are used to determine the three dimensional forces, \( \mathbf{f}_B \), and torques, \( \mathbf{\tau}_B \), acting on the aircraft by dimensionalizing the stability and control coefficients with the current airspeed, altitude and geometric properties of the aircraft and summing their contribution to the applicable forces and moments. A detailed treatment of these steps can be found in many texts on Flight Control such as *Airplane and Flight Dynamics and Automatic Flight Controls* by J. Roskam [17].

The aircraft response to the body forces and torques calculated from the stability and control coefficients is governed by the following equations of motion written in matrix form:

\[
\begin{align*}
\mathbf{\tau}_B &= \mathbf{I} \omega_B + \omega_B \times \mathbf{I} \omega_B \\
\mathbf{f}_B/m &= \dot{\mathbf{v}}_B + \omega_B \times \mathbf{v}_B - R^T g \\
\dot{\mathbf{R}} &= \mathbf{R} \times \omega_B \\
\dot{\Delta} &= \mathbf{R} \mathbf{v}_B
\end{align*}
\]

where the subscript \( B \) denotes that the vector is expressed in body coordinates. \( \mathbf{v}_B \) represents the linear velocity of the aircraft and \( \omega \) the angular velocity of the aircraft. \( \mathbf{R} \) is a 3x3 rotation matrix that transforms a vector expressed in the body frame to the inertial frame. \( \mathbf{R} \) can be calculated from knowledge of the angular position of the aircraft’s body frame with
respect to the inertial frame and vice versa. $R^T$ is the transpose of $R$ and hence converts a vector expressed in inertial coordinates to body coordinates. $g$ is a 3D vector specifying the gravitational acceleration with value $[0, 0, 9.81m/s^2]^T$. $I$ is the 3x3 inertia matrix. $\Delta$ specifies the aircraft position in the inertial coordinate system.

### 4.2.2 Flare Controller

The flare maneuver is achieved by means of the flare controller, that wraps around a vertical speed hold controller. The flare controller commands an exponentially decreasing vertical speed according to the following equation repeated from chapter 2.1.11.

$$v(t) = (v_0 - v_{td})e^{-\frac{t}{\tau}} + v_{td}$$

(4.9)

where $v_0$ represents the vertical speed at the start of the flare, $v_{td}$ the target touchdown speed at the expected touchdown point, and $\tau$ the time constant at which the vertical speed command is decreased. The flare controller passes the vertical speed target $v(t)$ to the vertical speed hold controller. The vertical speed hold controller was designed using the successive loop closure method. It uses feedback on state variables pitch rate, pitch angle and vertical speed and deduces an appropriate elevator deflection based on the error of commanded and actual pitch rate, pitch angle and vertical speed. First, the control loop is closed around pitch rate, then the pitch angle and finally vertically speed. Including feedback of the pitch angle in addition to pitch rate and vertical speed increases stability and allows the use of higher gains to speed up the aircraft response. The pitch rate controller is a simple proportional controller with feedback gain -2.7. The pitch hold controller uses proportional-integral feedback with gains 1 and 0.8 respectively. Finally, the vertical speed hold loop wraps around the pitch hold loop with proportional gain -0.03 and integral gain -0.015. Performance results of this flare controller are given in chapter 5.2.
4.3 Wind Model

The wind is modeled as a constant, lateral wind that does not change speed or direction with time. However, the lateral wind speed and direction changes with altitude, which allows modeling of the wind shear present in the atmospheric boundary layer as discussed in chapter 2.1.10. With this approach, at any given altitude, the aircraft’s motion in the inertial frame can be modeled as the sum of the translation of the wind frame with respect to the inertial frame and the translation of the aircraft with respect to the surrounding air. Vertical winds, as would be experienced in microbursts or similar phenomena are not considered, since these conditions have a strong, negative effect on the achievable autoland performance, while occurring very rarely. Hence, designing autoland for tolerance to vertical winds would unnecessarily lower its performance during nominal wind conditions and is therefore not included. However, any vertical wind has an effect on the measured $L/D_0$ and would hence be picked up by the $L/D_0$ Estimator and compensated for by the Trajectory Updater, up to the maximum possible.

Three different wind profiles are used for simulation purposes. The first profile is an idealized wind shear, where the wind speed reduces linearly from the constant value $V_{w,0}$ at 560 ft above the touchdown zone to zero at 0 ft above the touchdown zone. Figure 4-1 illustrates this wind profile. At all times, the wind direction is 180° south, so that a runway orientation of 0° north results in a tailwind on the downwind leg and a headwind on the final leg. Autoland assumes this wind speed profile for calculation of the expected glide ratio on each leg of the traffic pattern. Hence, if the wind estimator measures $V_{w,0}$ at the start of the downwind leg, this wind speed remains constant up until 560 ft above the touchdown zone and then decreases linearly to zero, the aircraft will hit the target touchdown point exactly, assuming no trajectory tracking errors and $L/D_0$ estimation errors have occurred.

The second and third wind models are taken from wind data recorded at the Dallas-Fort Worth Airport from January 1, 1998 to January 31, 1999 [24]. Profile 2, shown in Figure 4-2, is representative for a condition where the wind speed does not change significantly with a decrease in altitude. Hence, if the aircraft is flying into a headwind and the $L/D$
Figure 4-1: Wind profile for linear reduction in wind speed in boundary layer

Estimator expects the wind speed to reduce linearly, the aircraft will touch down short of the target touchdown point as will be shown and quantified in chapter 5. The third and final wind profile shown in Figure 4-3, represents an example of a low-level wind shear, where the wind changes direction close to the ground. This phenomenon may be observed in the morning hours when cold air trapped below a warmer layer of air creates an inversion weather condition.

### 4.4 Wind Estimator Model

The wind estimator is modeled as a perfect estimator that at any point in time knows the exact wind speed and direction. Any wind estimation errors present in the actual wind estimator of the Cirrus SR-22 autopilot will affect the accuracy of the calculation of $L/D_{exp}$ on the subsequent traffic pattern legs and result in an overshoot or undershoot of the target
Figure 4-2: Wind profile for near-constant wind speed in boundary layer

touchdown point. The contribution of the wind estimation error to the touchdown point dispersion was quantified in chapter 2.1.10 and is hence not included in the simulation, since the results would be redundant.

4.5 Performance Estimator Model

The Performance Estimator determines the current $L/D_0$ and calculates $L/D_{exp}$ on the subsequent legs using knowledge of the wind speed and direction and knowledge of the course of the subsequent legs to be flown. The implementation of this module follows the conceptual outline given in chapter 2.1.7. The details of the implementation are given in the appendix, chapters A.2.2 and A.2.3.
4.6 Energy Manager Model

The Energy Manager determines whether an energy error exists as outlined in chapter 2.1.8 and 2.1.9. Details of the implementation are given in the appendix, chapter A.2.9.

4.7 Trajectory Updator Model

The Trajectory Updator adjusts the length of the trajectory given a non-zero energy error and the location of the aircraft on the trajectory as described in chapter 2.1.9. Implementation details can be found in the appendix, chapter A.2.11.
Chapter 5

Results

In this chapter, the performance of the autoland prototype is evaluated in terms of touchdown point dispersion, sink rate on touchdown, attitude on touchdown and tolerance to GPS/WAAS vertical position error.

Touchdown point dispersion is evaluated using the point mass model described in chapter 4.1, under the three wind profiles as described in chapter 4.3. Uncertainty in wind shear in the boundary layer is expected to be the largest contributor to touchdown point dispersion in the final implementation on the aircraft. Other contributors are errors in measurement of $L/D_0$ and measurement of current wind, and errors in tracking the planned trajectory. The point mass model removes these error sources by providing perfect aircraft state information to the Performance Estimator (which results in perfect $L/D_0$ and $L/D_{exp}$ estimation) and second, by ensuring near perfect tracking of the planned trajectory. Hence, the effect of wind can be examined in isolation.

Sink rate on touchdown, attitude on touchdown and tolerance to GPS/WAAS vertical position error, termed the ”touchdown performance parameters”, are evaluated using the 6 degree of freedom, high fidelity aircraft model, as described in chapter 4.2.1, for a range of $L/D_0$. Variations in $L/D_0$ were chosen as test cases, since changes in $L/D_0$ have the strongest effect on the aforementioned criteria, when compared to the GPS/WAAS vertical position...
error. However, the effect of the GPS/WAAS error on the touchdown performance parameters is small compared to the effect of variations in $L/D_0$, as long as the GPS/WAAS error remains within +/- 6.6 ft (2 m). Lateral wind has no effect on the sink rate on touchdown, attitude on touchdown or tolerance to GPS/WAAS vertical position error and is therefore not included in the evaluation of the touchdown performance parameters.

### 5.1 Evaluation of Effect of Wind on Touchdown Point Dispersion

This chapter evaluates the effect of uncertainty in wind shear in the boundary layer on the location of the touchdown point. First, the operation of autoland during tracking of the traffic pattern is illustrated graphically for the linear wind profile shown in Figure 4-1. This test will be termed Test Case 1. The same procedure is then performed for the near-constant wind profile as shown in Figure 4-2 and the inversion wind profile as shown in Figure 4-3. These tests will be referred to as Test Case 2 and Test Case 3 respectively. Autoland’s performance under Test Case 2 and 3 is evaluated and discussed, however, detailed graphics as for Test Case 1 are not provided since the information would be redundant.

#### 5.1.1 Test Setup For Evaluation of Effect of Wind on Touchdown Point Dispersion

Each test case will be run on a left baseline traffic pattern consisting of downwind leg, base leg and final leg at right angles to one another and leg lengths of 5,000 ft. The waypoint locations are given in a local North-East coordinate system with the runway threshold located at the origin of the coordinate system. The trajectory is oriented in such a way that the runway heading is $0^\circ$ due north. All distances in the following graphs are given in ft. The
baseline waypoint locations are:

\[
\begin{bmatrix}
F_{D,north} & F_{D,east} \\
F_{B,north} & F_{B,east} \\
F_{F1,north} & F_{F1,east} \\
F_{F,north} & F_{F,east}
\end{bmatrix} = 
\begin{bmatrix}
0 & -5,000 \\
-5,000 & -5,000 \\
-5,000 & 0 \\
-500 & 0
\end{bmatrix}
\]

The resulting test trajectory is shown in Figure 5-1. The point marked with the red star indicates the target touchdown point. The two black dots indicate the end of the overshoot and undershoot margin respectively. The no-wind, wings-level glide ratio $L/D_0$ is taken to be 10:1, which results in a glide ratio of 8:1 during a 30° banked turn.

Figure 5-1: Left traffic pattern trajectory used to test the performance of the autoland software prototype
5.1.2 Illustration of Test Case 1 for Evaluation of Effect of Wind on Touchdown Point Dispersion

Test Case 1 operates under the linear wind profile shown in Figure 4-1. The wind speed is 5 kt in direction 180° south, decreasing linearly to zero from 560 ft. This is the exact wind profile autoland assumes. Hence, there is no error in estimating the expected glide ratio for each leg $L/D_{exp}$. Hence, assuming no trajectory tracking errors, the aircraft would land right on the target touchdown point.

The aircraft is started at altitude 1950 ft, approximately 100 ft higher than required to follow the baseline traffic pattern. This results in a first trajectory update right at the start of the traffic pattern as shown in Figure 5-2. The Trajectory Updator deduced the amount

![Updated trajectory vs Baseline trajectory](image)

Figure 5-2: Traffic pattern after update from 100 ft altitude error

to move the base leg fix and final leg intercept fix, $d$, from the distance to go until the base leg fix, $L$, the current angle of the downwind leg to the base leg, $\alpha_p$ and the average glide ratio expected on the remainder of the trajectory, $L/D_{av}$. $d$ becomes 500 ft, which results in a shift of the base leg fix by 354 ft south and 354 ft west to $F_{B,north} = -5354$ and
\( F_{B, \text{east}}' = -5354 \). The final leg intercept fix was moved 534 ft south to \( F_{F1, \text{north}}' = -5354 \) and \( F_{F1, \text{east}}' = 0 \). The energy vs. range-to-go curve before the update is shown in Figure 5-3. The single square indicates the current energy level of the aircraft at the expected range-to-go until touchdown. The blue curve is the desired energy level, whereas the red curve and green curve represent the upper and lower energy boundaries respectively. After the first update, the trajectory is elongated in such a way, that the current aircraft energy at the new range-to-go until touchdown coincides with the blue curve. This is shown in Figure 5-5. Close inspection shows that the energy level of the aircraft has remained constant at 1950 ft, while the range-to-go has been increased from 14,000 ft to 15,500 ft.

As soon as the aircraft intercepts the base leg, the aircraft’s \( L/D_0 \) is dropped by 5%. This may occur, for example, due to unexpected flap extension. The drop in \( L/D_0 \) requires a shortening of the trajectory. Since this drop occurred sufficiently early on the base leg, the full energy error resulting from this drop, can be corrected. The updated trajectory is shown in Figure 5-7 in red. The blue line on the graph indicates the actual trajectory flown by the aircraft so far. The corresponding energy vs. range-to-go curves before and after the update are shown in Figure 5-8 and 5-10. Before the update, the aircraft energy is 64 ft below the energy level required. Taking into account the distance to go until the final leg intercept fix, \( L \), the angle of the currently planned base leg to the final intercept
Figure 5-5: Energy vs. range-to-go curve after first update

Figure 5-6: Energy vs. range-to-go curve after first update, enlarged

Figure 5-7: Traffic pattern after second update due to reduction in $L/D_0$

leg, $\alpha_p$ and the average glide ratio expected on the remainder of the trajectory, $L/D_{av}$, the Trajectory Updater deduced that the final leg intercept fix should be moved by a total of $d = 540$ ft, which corresponds to a change in the North and East coordinate of 380 ft to $F_{FI,north}' = -4975$ and $F_{FI,east}' = -380$. Since no previous updates have been performed on the base leg, $d_{total} = 0$.

Due to the wind behaving exactly as expected and the trajectory tracking controller
following the prescribed trajectory precisely, the aircraft finally touches down at the target touchdown point with an error of <10 ft. Figure 5-12 summarizes the initially planned trajectory, the trajectory after the first update and after the second update, and the actual trajectory, the aircraft has flown.
Figure 5-12: Trajectory before and after each update and actual trajectory flown
<table>
<thead>
<tr>
<th>Test Case 1: Linear wind profile</th>
<th>Test Case 2: Near-constant wind profile</th>
<th>Test Case 3: Inversion wind profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial energy error, $\Delta E/W_1$</td>
<td>100 ft</td>
<td>135 ft</td>
</tr>
<tr>
<td>Distance base leg fix was moved, $d_b$</td>
<td>500 ft</td>
<td>690 ft</td>
</tr>
<tr>
<td>Energy error after change in $L/D_0$, $\Delta E/W_2$</td>
<td>-49 ft</td>
<td>-57 ft</td>
</tr>
<tr>
<td>Distance final leg intercept fix was moved, $d_{fi}$</td>
<td>-435 ft</td>
<td>-534 ft</td>
</tr>
<tr>
<td>Horizontal distance covered during flare, $x_h$</td>
<td>1,450 ft</td>
<td>1,379 ft</td>
</tr>
<tr>
<td>Overshoot/undershoot of the target touchdown point</td>
<td>7 ft</td>
<td>-398 ft</td>
</tr>
</tbody>
</table>

Table 5.1: Summary of test results

5.1.3 Summary of Effect of Wind on Touchdown Point Dispersion

Table 5.1 summarizes the touchdown point dispersion of the autoland prototype during Test Case 1 to 3. The energy errors, $\Delta E/W_1$ and $\Delta E/W_2$ incurred at the start of the test run due to excess altitude, and on the base leg due to the change in glide performance, are included for reference. Furthermore, the change in location of the leg fixes, $d_b$ and $d_{fi}$ is given to show the magnitude of the changes in the trajectory for a given energy error. As expected, $d_{fi}$ for updates on the base leg is greater, than $d_b$ for updates on the downwind leg for the same energy error. Furthermore, the horizontal distance covered during the flare, $x_h$, is included to show to effect of wind on the flare distance.

Test Case 1 was discussed in the previous section. In Test Case 2, the aircraft landed 398 ft short of the target touchdown point. This is due to the fact, that autoland assumed that the wind speed would reduce from 5 kt to 0 kt, as the aircraft descended along the final leg. Since the wind speed only reduced to 4.5 kt, however, the aircraft still experienced a tailwind of magnitude 3.9 kt upon touchdown. Refer to Figure 4-2 for verification of wind speed and direction. Stepping through the calculations outlined in chapter 2.1.10 on determining the glide ratio estimation error, results in $L/D_{err} = +6.4\%$. For a 6,000 ft long final leg, this $L/D_{err}$ results in an undershoot of the target touchdown point of -394 ft, which is close to
the value determined from simulation. The ground distance covered during the flare is less than the nominal flare distance by 71 ft, since the aircraft still experienced a headwind of 3.9 kt for the duration of the flare.

In Test Case 3, the aircraft landed 75 ft beyond the target touchdown point. Here, the wind speed initially reduced less than expected over the length of the final leg, but then abruptly changed direction from a headwind to a tailwind at 200 ft of altitude. This change in direction suddenly increased the glide ratio of the aircraft and changed the landing that would have been short, into a landing that was slightly long. The distance covered during the flare in this case was longer than nominal by 119 ft, since the aircraft experienced a tailwind of 2 kt of magnitude for the duration of the flare.

5.1.4 Discussion of Effect of Wind on Touchdown Point Dispersion

The autoland prototype met the performance specifications for Test Cases 1 and 2 by keeping the touchdown point dispersion within the unilateral tolerance limit of -400 ft for errors in predicting boundary layer wind shear in a headwind case. For an approach in a tailwind, the error margin for touchdown point dispersion would have been +400 ft.

In Test Case 3, autoland assumed an approach into a headwind. The change in wind direction, however, changed this case into an approach into a tailwind that caused an overshoot of 75 ft, which was not accounted for by the unilateral tolerance bound for wind shear. However, this small overshoot is acceptable, since the length of the runway easily absorbs this magnitude of error. If Test Case 3 would have been an approach with a tailwind, however, the change in wind direction would have caused an undershoot, that was not accounted for. If all other error tolerance bounds for GPS vertical position error, and $L/D_0$ and wind measurement error, were exhausted at their undershoot limit, the aircraft would touch down short of the runway threshold, which should be avoided. Hence, the currently unilateral tolerance bound for error in wind shear prediction should be extended to contain margin in the opposite direction as well, in order to be able to absorb changes in wind direction.
Since no GPS error and no performance estimation error due to measurement error was included in the simulation, the additional overshoot and undershoot margin of +/- 600 ft was not used.

In the actual aircraft/FMS/autopilot system, errors in estimating wind speed and direction, and \( L/D_0 \) will naturally occur. This effect is expected to produce a touchdown point dispersion of less than +/-300 ft. However, this will need to be verified in practice.

Inevitably, there will also be a non-zero GPS error, which causes the aircraft to touchdown sooner or later during the flare maneuver. The overshoot and undershoot caused by the GPS error is accounted for by bilateral tolerance bounds of +/-300 ft. The capability to absorb the GPS error in terms of tolerance on touchdown airspeed is discussed in the following chapter.

### 5.2 Evaluation of Flare Performance

In this chapter, the performance of autoland’s flare controller is evaluated with respect to sink rate on touchdown, attitude on touchdown and tolerance to GPS/WAAS vertical position error for different \( L/D_0 \). In addition to that, the horizontal distance travelled during the flare maneuver is examined.

In the absence of a vertical wind component, non-zero wind does not change the sink speed of the aircraft. Hence, the flare controller acts on an aircraft with a specific \( L/D_0 \) in the same manner, regardless of whether wind is present or not. Non-zero wind changes the distance the aircraft travels over ground during the flare maneuver. However, this effect has already been included in the previous chapter and hence zero wind is used for this test.

#### 5.2.1 Test Setup for Evaluation of Flare Performance

The performance of the flare controller is tested using the high fidelity, 6 degree of freedom, rigid body simulation using three different \( L/D_0 \): 7, 10 and 13 representing Test Case 1, 2
and 3 respectively. The three different $L/D_0$ were achieved by adjusting the zero-lift drag coefficient, $C_{D,0}$. An $L/D_0$ of 10 is expected to be the nominal glide ratio, 7 the low end of the glide ratios to be expected, and 13 the upper end. In each case, the aircraft descends from an altitude 200 ft in its trim condition at airspeed of 90 kt and zero thrust. When reaching 70 ft altitude, the flare maneuver is initiated and the flare controller turned on. The test setup is illustrated in Figure 5-13. For a detailed description of the flare maneuver,

![State Machine Diagram](image)

Figure 5-13: Test setup for performance evaluation of flare controller

refer to chapter 2.1.11.

### 5.2.2 Illustration of Test Case 1 for Evaluation of Flare Performance

The following test run shows the simulated aircraft’s behavior during the flare maneuver for Test Case 1, with an $L/D_0$ of 7, the hardest case for the flare controller to handle. This is due to the energy being lowest for an aircraft with $L/D_0$ of 7, compared to an $L/D_0$ of 10 or 13, if all else is equal. Yet, the flare controller needs to slow down the sink speed of the low energy aircraft to the same target sink speed of 3.3 ft/s as for the higher energy aircraft. This is achieved by trading airspeed for sink speed. However, airspeed can only be traded up until the airspeed has reduced to the stall speed. Below this airspeed limit, the aircraft would experience a sudden decrease in lift, resulting in a sudden increase in sink rate, opposite the desired effect. However, as discussed in chapter 2.1.11, the parameters of the flare controller were chosen so that sufficient stall margin is available even in the worst case scenario considered.
Figure 5-14 shows the sink speed command given by the flare controller and the aircraft’s response. This simulation assumes zero vertical GPS error, but considers the effect of touching down 6.6 ft (2 m) earlier or later during the summary of the results in the following chapter. The sink speed command is shown via the solid line, whereas the aircraft response is shown via the dotted line. The flare maneuver is started after 5.5 s, after the aircraft has sunk to 70 ft altitude. From this point on, the flare controller commands an exponentially decreasing sink speed from 20.8 ft/s to 3.3 ft/s. The aircraft follows the sink speed command reasonably well and manages to establish a sink rate of 3.8 ft/s upon touchdown. The swing in aircraft sink speed at the bottom of the exponential sink speed command is due to the aircraft entering the ground effect, which is turned on as a step input at 33 ft of altitude. Figure 5-15 shows the aircraft’s motion in the vertical plane during the flare maneuver. At around 40 ft, the decrease in sink rate becomes noticeable. The descent into the ground effect at 33 ft of altitude is again noticeable as a small bump in the vertical profile.

Figure 5-16 shows the horizontal and vertical position, as well as the horizontal and vertical speed of the aircraft during the flare maneuver. The plot of horizontal speed vs. time shows that the aircraft touched down with an airspeed of 75 kt, which is above the zero flaps stall speed of 73 kt [8].
Figure 5-15: Altitude vs. horizontal distance travelled during flare maneuver

Figure 5-16: Change in aircraft position and velocity during flare maneuver

Figure 5-17 shows the evolution of the angle of attack, $\alpha$, the flight path angle $\gamma$, and the pitch angle $\theta$ during the flare maneuver. For an $L/D_0$ of 7 with zero wind, the aircraft descends at an angle of $-8.1^\circ$. By the time of touchdown, $\gamma$ has increased to $-1.5^\circ$, which is
the flight path angle that determines the overshoot or undershoot of the target touchdown point, for a non-zero GPS vertical position error. With this flight path angle, the assumed GPS error of +/- 6.6 ft (2 m), produces an overshoot and undershoot of +/- 300 ft as previously illustrated in chapter 2.1.10. The angle of attack required to create the lift that balances the aircraft weight at this flight path angle is 6.8°. During the flare maneuver, $\alpha$ increases to 9°, which provides good margin to the stall angle of attack of 14°. The aircraft’s pitch angle is slightly negative at the start of the flare, meaning, the aircraft is in a slight nose-down attitude. Due to the increase in flight path angle and decrease in airspeed during the flare maneuver, however, the pitch angle increases to a pitch-up attitude of 7.5°. This attitude is deemed sufficient to avoid nose-wheel barrowing on touchdown.

5.2.3 Summary of Flare Performance Test Results

Table 5.2 summarizes the flare controller’s touchdown performance for $L/D_0$ of 7, 10 and 13 in terms of the touchdown sink rate $V_{v,td}$, the touchdown airspeed, $V_{a,td}$, the pitch attitude, $\theta_{td}$, the available stall margin, $m$ and the horizontal distance travelled during the flare, $x_h$. 
\[
\begin{array}{|c|c|c|c|}
\hline
 & L/D_0 = 7 & L/D_0 = 10 & L/D_0 = 13 \\
 & (C_{D,0} = 0.05) & (C_{D,0} = 0.02) & (C_{D,0} = 0) \\
\hline
\text{Pitch angle at start of flare, } \theta_0 & -1.3^\circ & 1.1^\circ & 2.4^\circ \\
\text{Sink rate at start of flare, } V_{v,0} & 20.8 \text{ ft/s} & 15.5 \text{ ft/s} & 11.6 \text{ ft/s} \\
\text{Pitch angle at touchdown, } \theta_{td} & 7.5^\circ & 7.2^\circ & 6.7^\circ \\
\text{Sink rate at touchdown, } V_{v,td} & 3.8 \text{ ft/s} & 3.1 \text{ ft/s} & 2.9 \text{ ft/s} \\
\text{Airspeed at touchdown, } V_{a,td} & 75 \text{ kt} & 78 \text{ kt} & 81 \text{ kt} \\
\text{stall margin, } m & 8.25 \text{ ft (2.5 m)} & 13.2 \text{ ft (4.0 m)} & 23 \text{ ft (6.9 m)} \\
\text{Horizontal distance covered during the flare, } x_h & 1,450 \text{ ft} & 1,508 \text{ ft} & 1,566 \text{ ft} \\
\hline
\end{array}
\]

Table 5.2: Summary of flare performance test results

The pitch attitude, \( \theta_0 \), and sink rate, \( V_{v,0} \), at the start of the flare are provided for reference.

### 5.2.4 Discussion of Flare Performance Test Results

The flare controller met the required touchdown performance in each case. As discussed in chapter 2.1.11, the sink speed on touchdown should remain within 1 ft/s to 6 ft/s to avoid damage to the aircraft and passengers. The flare controller managed to keep this parameter within \( \pm 0.5 \) ft/s from the target 3.3 ft/s. Each time, the aircraft landed with a pitch up attitude of \( > 5^\circ \), which is deemed sufficient to prevent the nose wheel from touching down first. The airspeed remained above the stall speed of 73 kt, and provided a stall margin of \( > 6.6 \) ft (2 m) in each case to absorb the expected vertical GPS error. For a positive GPS error, i.e. the aircraft touches down 6.6 ft (2 m) higher in altitude than expected, inspection of the plot of vertical speed vs. time and flight path angle vs. time from chapter 5.2.2 reveals, that the touchdown sink speed and flight path angle would only be slightly higher than in the non-zero GPS error case and is hence well within the the acceptable region.

Due to the differences in sink speed at the start of the flare for each \( L/D_0 \), the horizontal distance travelled during the flare is different in each case. The aircraft with the worst glide ratio of 7:1 touched down 50 ft earlier than the aircraft with the nominal glide ratio of 10:1. Conversely, the aircraft with the best glide ratio of 13:1, touched down 58 ft later than the nominal aircraft. Compared to the touchdown point dispersion from errors in
expected glide ratio, $L/D_{exp}$, as discussed in the previous chapter, changes in $L/D_0$ add only little touchdown point dispersion. Hence, the design choice is to absorb this small effect by assuming a nominal flare length of 1,450 ft, i.e. for the worst assumed glide ratio. Any deviation off this flare length then results in a small overshoot of the target touchdown point, which is tolerable.
Chapter 6

Conclusion and Future Work

6.1 Conclusion

The design for an autoland system for general aviation aircraft presented, aims to address the emergency scenarios of partial or complete pilot incapacitation and loss of power. The scope of the proposed concept encompasses automatic landing site selection and guidance to the selected landing site in addition to the usual scope of an autoland system, which only covers the final approach to land. The challenge arising from lack of auto-throttle and lack of radar altimeter, as typically encountered in general aviation aircraft, was alleviated by performing the final approach at power-off and placing the target touchdown point, so as to provide overshoot and undershoot margin for expected aircraft performance and GPS position errors.

The theoretical analysis presented and performance results gathered in simulation show that an automatic landing of a general aviation sized aircraft is possible on a 5,000 ft runway, if the error in $L/D_{exp}$ prediction from measurement errors is kept within +/- 5% and the error from predicting wind shear is kept within -6.5% for a headwind approach and +6.5% for a tailwind approach. The vertical GPS error should not exceed +/- 6.6 ft (2 m) to remain within the allotted overshoot/undershoot margin and the stall margin provided by the flare.
controller.

Touchdown point dispersion from errors in trajectory tracking performance were deemed to be negligible. However, any deviation off the planned trajectory, starting from the turn onto the final leg, will contribute to overshoot or undershoot of the target touchdown point by the amount the actual trajectory flown differed from the trajectory planned.

Assuming perfect trajectory tracking and an approach into a headwind, the earliest touchdown would occur at the runway threshold, whereas the latest touchdown would be 1,600 ft past the runway threshold, if each error source caused the maximum positive and maximum negative allocated error respectively. On a 5,000 ft runway, this leaves 3,400 ft of rollout distance, for a range of rollout distance from 1,000 ft - 2,000 ft required by a trained pilot for the Cirrus SR-22 [8]. Given that the system is intended to be used by an untrained passenger in case of pilot incapacitation, the additional margin is deemed necessary for a successful landing by an untrained pilot.

6.2 Future Work

The simulated autoland performance results appear promising, however, the results would need to be verified during flight testing. Furthermore, performance enhancement of the current design may be necessary in order to compensate for difficulties with tight trajectory tracking, or in order to land on runways shorter than 5,000 ft. The performance of the autoland design presented can be improved in two ways: via operational means by adding more complex trajectory updating, or via removal of one or more hardware constraints. Both means are briefly discussed.
Performance Enhancement through Addition of More Complex Trajectory Updating

The current trajectory updating methodology in the traffic pattern allows for trajectory updating only on the downwind leg and base leg using two straightforward schemes. During turns and on the final leg, trajectory updating is inhibited. However, for maximum performance, trajectory updating would need to be possible at all times. This could be achieved by incorporating the capability to change the course of a previously updated traffic pattern leg, while the aircraft is in the process of intercepting the updated leg. Furthermore, more complex maneuvers that allow a change in turn radius or turn direction, while executing a maneuver could be added to control to a desired energy vs. range-to-go state at all times. The current design locks the decision about the shape of a maneuver at the start of the maneuver and only allows updates after the completion of the maneuver.

Performance Enhancement from Removal of Hardware Constraints

The current design was developed with the hardware restrictions typically found on general aviation aircraft in mind. These constraints are: lack of automatic throttle control, lack of auto-rudder, lack of auto-brakes and lack of radar altimeter. However, if these limitations were removed, the performance of autoland could be improved. Figure 6-1 repeats Figure 1-1 from chapter 1.4 with the added automatic control lines shown as dashed lines. Also, auto-flaps are included in the figure. Flaps are not typically driven by a conventional autoland system. However, addition of this capability would offer automatic glide ratio modulation, which would be of use to the autoland system proposed.

If automatic throttle control were to be added to the aircraft, the limitation on the achievable flight path angles and airspeed would be removed. Hence, any flight path angle and airspeed within the performance envelope of the aircraft could be flown, meaning the standard −3° flight path angle could be flown and power adjustments could be used to hold the aircraft on this glide path even under changing wind and aircraft performance.
Addition of automatic rudder, would provide two benefits. First, it allows for the execution of a "de-crab" maneuver during a crosswind landing, which aligns the nose of the aircraft with the runway centerline just before touchdown. Second, it adds the possibility to control the steepness of the glide path without adding airspeed, by manipulating the aerodynamic efficiency of the aircraft. This would be achieved by performing a deliberate slip, in which the aircraft is banked using the ailerons, and the resulting tendency to turn counteracted by the application of opposite rudder. The amount of flight path angle control available with this maneuver depends on the rudder authority available. The rudder needs to counteract the horizontal component of the lift vector that tends to pull the aircraft into a turn, in order to maintain the current aircraft heading. As soon as maximum rudder deflection is reached, further steepening the bank angle, results in the aircraft turning.
Automatic flap deployment could be used to deliberately modulate the aircraft’s glide ratio. Since extended flaps lower the glide ratio, this control input could be used in a similar manner as the slip via the automatic rudder. In this respect, automatic flaps are redundant to the automatic rudder. However, automatic flaps could be used to extend the glide ratio modulation range of the rudder, whenever rudder authority has been exhausted.

Addition of a radar altimeter would remove the touchdown point dispersion incurred from the GPS vertical position error, since the measurement error of the radar altimeter is negligible compared to the GPS position error. This would also allow autoland to control to a softer touchdown sink rate than 3.3 ft/s, since no airspeed margin is required for a potentially late touchdown due to GPS error.
Appendix A

Module Descriptions

A.1 Modules of Layer Destination Planning

A.1.1 Engine Status Detection

Inputs

- % power, \( P \): estimation of engine power derived from manifold pressure, indicated air speed, outside air temperature, pressure altitude, engine speed, and fuel flow within the Cirrus SR-22 and made available on the autopilot data bus by the integrated flight deck.

Outputs

- engine status: engine status, 0 = ”Out”, 1 = ”Operational”, 2 = ”Indeterminate”

Trigger

Initiation of autoland
Description

The engine status will be classified as either "Engine Operational", "Engine Out" or "Indeterminate". The engine status is deduced to be "Engine Operational", if the engine output power is equal to or higher than the power required to maintain straight and level flight. This setting is denoted by $p_u$. For the SR-22, this value is about 20\%. The engine is deduced to be "Out", if power output is smaller than the maximum power setting at which a successful power-on approach can be performed. This value is referred to by $p_l$. For the SR-22, this is about 5\%. In case the power setting is in between $p_u$ and $p_l$, the engine status is set to "Indeterminate".

The logic for output "Engine Status" becomes:

$$
\text{Engine Status} = \begin{cases} 
0, & \text{if } P < p_l \\
1, & \text{if } P \geq p_u \\
2, & \text{if } p_l \leq P < p_u 
\end{cases}
$$

A.1.2 Range calculator

Inputs

- **Engine status**: engine status output from Engine Status Detector
- **Fuel range, $R_f$**: achievable range in nautical miles calculated internally to the SR-22 from amount of initial fuel (set manually) and fuel flow
- **E/W**: current aircraft energy normalized by weight (input used if engine status = "Out" or "Indeterminate")
- **L/D_POH**: expected glide ratio of the aircraft for an idling engine, design constant
- **L/D_0**: estimated no-wind, wings-level glide ratio of the aircraft from the Performance Estimator
- LS.E\_td, \( E_{td}/W \): desired energy on touchdown normalized by weight, design constant

- % margin, \( m \): percent margin to be used over maximum achievable range as safety factor

**Outputs**

**Range, \( R \):** achievable range radius

**Trigger**

called by higher level component, Landing Site Selector

**Description**

The Range Calculator uses input "engine status" to decide whether to route input Fuel range, \( R_f \), or a calculated glide range, \( R_g \) to output \( R \). If the engine is operational, \( R \) is assigned \( R_f \). In the remaining cases, \( R \) is assigned \( R_g \):

\[
R = \begin{cases} 
R_f, & \text{if engine status = "Out"} \\
R_g, & \text{if engine status \neq "Out"} 
\end{cases}
\]

The glide range, \( R_g \) including margin, \( m \) can be calculated from the simple formula:

\[
R = \frac{1}{1 + m} (E/W - E_{td}/W) \cdot L/D \quad (A.1)
\]

where L/D is either the L/D\_0 returned from the Performance Estimator or the average expected glide ratio from manufacturer specifications, L/D\_POH. L/D\_0 is used whenever available. However, the Performance Estimator requires \( 10 \) s to compute a stable L/D estimate. Hence, a measured glide ratio may not be available by the time the Range Calculator executes on the first run of the Landing Site Selector. The Range Calculator ignores the
current aircraft heading and calculates a perfectly circular footprint, but this is deemed acceptable since the included margin exceeds the error made in ignoring the heading by an order of magnitude.

A.1.3 Landing Site Prioritizer

Inputs

- **criteria level, c**: criteria level under which landing site search is performed (1-4)

- **state.loc**: latitude and longitude of current aircraft position

- **Engine status**: engine status from the Engine Status Detector, either ”Operational”, ”Out” or ”Indeterminate”

- **L_rwy**: desired runway length for case engine ”Operational” to search for a runway that fulfills level 1 criteria

- **range, R**: reachable range as returned by Range Calculator

- **rwy info**: information on location, heading, length and displaced threshold of the selected runway

- **emergency landing site info**: information on location and heading of a selected emergency landing site

- **aero charts**: color image files of aeronautical charts for automatic detection of unpopulated areas

- **METARS**: aviation weather information from built-in XM-satellite radio containing information on wind speed in knots and wind direction in true north for the candidate landing sites
Outputs

- **LS[].loc**: array of latitude and longitude of the runway threshold or location of the earliest desired touchdown point for alternate landing site
- **LS[].hdg**: runway heading, i.e. approach direction, with respect to true north
- **LS[].elevation**: elevation of the touchdown zone of the selected landing site above mean sea level

Trigger

called by higher level component, Landing Site Selector

Description

The landing site prioritizer is used by the landing site selection process to compile a prioritized list of landing sites meeting the criteria specified by input *criteria level*. The landing site selection process is outlined in chapter 2.1.3. Figure A-2 illustrates how the landing site prioritizer compiles an ordered list of candidate landing sites for criterial level 1, which is the entry point for landing site selection when the engine is operational. Figure A-3 shows the flow of functions of the landing site prioritizer for criteria levels 2-4. Criteria level 2 is the entry point for landing site selection in case *engine out*, whereas criteria level 4 is used for engine status *indeterminate*.

If the engine is operational, the achievable range can easily amount to one to two thousand miles. Hence, not all runways within range should be retrieved and compared to find the very best. Rather, autoland will search for a ”good enough” runway within increasing search radii and stop the search process as soon as a runway is found that meets the ”good enough” criteria. This ensures that runways closest to the aircraft’s current position are prioritized. This is illustrated in Figure A-1.

The hard stop set by the Range Limit would realistically not be reached within the conti-
Increasing search radii

Range Limit
Increasing search radii

Figure A-1: Runway search in engine operational case

In the event that no runway fulfilling the "good enough" criteria could be found within the range limit, autoland relaxes the criteria level to level 2, "any runway" and selects the longest runway from the set of runways within range. The good enough criteria applied here is runway length exceeding 5,000 ft. Once at least one runway has been found that meets this criteria, the search process is stopped. If more than one runway is found, runways are prioritized by length. The approach direction will be chosen in such a way, that the aircraft experiences a headwind on approach. The search process is illustrated in Figure A-2.

In case engine out, the landing site selector initializes the criteria level to level 2, "any runway". At this level, the Landing Site Prioritizer attempts to find a reachable runway using the runway database of the FMS. If there is more than one solution, the Landing Site Prioritizer orders the candidate runways by length. It would have been possible to include a multitude of additional criteria such as wind conditions, runway surface, lighting, availability of emergency equipment at airport, etc. in the prioritization scheme, but since only a small number of runways is likely to be returned in the engine out case and runway length is the overarching criteria for probability of a successful landing, only runway length is included.
in the prioritization scheme. However, if both ends of the longest runway within range can be reached, autoland will select the end of the runway that results in a headwind on approach. This approach direction can be deduced from XM weather information (METAR) received at regular intervals from satellite data and runway orientation contained in the runway database. If no runway can be found, the landing site selection process will relax the criteria level to level 3 and request the Landing Site Prioritizer to attempt to find a known emergency landing site from the emergency landing site database. Since there is generally no information on the size of the emergency landing site, the landing site prioritizer orders the emergency landing sites within range by distance to go, in the event that more than one solution was found. The emergency landing site database is not included by default with the FMS, but databases, such as the "Worldwide Soaring Turnpoint Exchange" maintained
by the glider community, may be uploaded and stored permanently within the existing FMS [11]. If the Landing Site Prioritizer also cannot find an emergency landing site within range, it lowers the criteria level to 4 and attempts to find an unpopulated area from images of the area maps. Unpopulated regions are depicted with a light green color on aviation maps, which can automatically be searched for using color detection algorithms. MATLAB’s Image Processing Toolbox contains routines that can be used for this purpose [20]. For landings in unpopulated areas, distance to go is used as the ordering criteria with the approach direction chosen as the current heading of the aircraft irrespective of wind conditions.
A.1.4 Initial Point Calculator

Inputs

- **LS.loc**: latitude and longitude of the runway threshold or location of the earliest desired touchdown point for alternate landing site

- **LS.hdg**: runway heading, i.e. approach direction, with respect to true north

- **state.loc**: latitude and longitude of current aircraft position

Outputs

- **LS.dir**: direction of the traffic pattern, either right or left

- **LS.IP**: location of the initial point in local North-East coordinates

Trigger

called by higher level component, Landing Site Selector

Description

Given the current aircraft position, state.loc, the location of the runway threshold, LS.loc, and the runway heading, LS.hdg, this function first decides whether the traffic pattern is to be flown right or left. This can be deduced from the cross-product of the vector from the current position to the runway threshold position and the runway heading vector. A positive sign of the result indicates the pattern should be flown left, a negative sign means the pattern should be flown right. This can be verified using the right-hand rule on vectors V1 and V2 in Figure A-4.
Figure A-4: Pattern direction from cross product of V1 and V2

\[ V1 \times V2 = \begin{cases} 
-ve & \text{for } "Left" = 1 \\
+ve & \text{for } "Right" = 0 
\end{cases} \]

Since the initial point, LS.IP, is given in the local North-East coordinate frame, there is only two options for the location of the initial point:

\[ \text{LS.IP(North, East)} = \begin{cases} 
(0, -5,000 \text{ ft}), & \text{if } \text{LS.dir} = "Left" \\
(0, 5,000 \text{ ft}), & \text{if } \text{LS.dir} = "Right" 
\end{cases} \]

Refer to chapter 2.2.2 for a description of the local coordinate system.

A.1.5 Touchdown Energy Calculator

Inputs

- **LS.elevation**, \( h_{td} \): elevation of the touchdown zone of the selected landing site above mean sea level

- \( V_{a,t_d} \): desired airspeed on touchdown, design constant

- **W**: aircraft weight, design constant

Outputs

- **LS.E_{td}**, \( E_{td}/W \): desired energy on touchdown normalized by weight, design constant
Trigger

called by higher level component, Landing Site Selector

Description

The touchdown energy can simply be deduced from the elevation of the touchdown zone and the desired airspeed on touchdown.

\[
E_{td}/W = h_{td} + \frac{V_{a,td}^2}{2g}
\]

(A.2)

with \( g = 9.81 m/s^2 \).
A.2 Modules of Layer Trajectory Planning

A.2.1 Energy Calculator

Inputs

- \texttt{state.h}, \( h \): current altitude of the aircraft above mean sea level

- \texttt{state.V.a}, \( V_a \): current airspeed of the aircraft

Outputs

- \( E/W \): current aircraft energy normalized by weight

Trigger

Sampling clock, every 1 s

Description

The aircraft energy can be found from the sum of potential and kinetic energy.

\[
E/W = h + \frac{V_a^2}{2g}
\]  

(A.3)

A.2.2 L/D\_0 estimator

Inputs

- \texttt{state.loc}, \( P \): latitude and longitude of the current aircraft position

- \texttt{state.V}, \( V_a \): current airspeed of the aircraft
• **state.course**, \( \chi \): angle of the current aircraft track with respect to true north

• **state.wind**: magnitude, \( V_w \), and direction, \( \chi_w \), of the wind

• **state.hdg**, \( \psi \): current aircraft heading with respect to true north

• **state.vspeed**, \( V_{sink} \): aircraft vertical speed in m/s

• **state.bank**, \( \phi \): current bank angle of the aircraft

• \( E/W, E_i/W \): current aircraft energy normalized by weight

**Outputs**

• \( L/D_0 \): aircraft no-wind, wings-level glide ratio

**Trigger**

called by higher level function, Performance Estimator, executes every 1 s

**Description**

This module measures an instantaneous, apparent glide ratio \( L/D_i \) and corrects this value for bank angle and wind effects to arrive at the no-wind, wings-level glide ratio, \( L/D_{0,i} \). \( L/D_i \) is calculated from the change in energy, \( \Delta E/W \), and distance travelled over ground in a given time interval, \( \Delta R \). After correction for wind and bank angles effects, the samples of \( L/D_{0,i} \) are passed through simple moving average filter with a sample depth of 10 to arrive at smoothed measurement of the no-wind, wings-level glide ratio \( L/D_0 \). With a sampling time of 1 s, the first \( L/D_0 \) is therefore available after 10 s of the initiation of the Performance Estimator. The change in energy, \( \Delta E/W \), over one sampling interval becomes:

\[
\Delta E/W = E_i/W - E_{i-1}/W
\]  

(A.4)
The distance, $\Delta R$, travelled within one sampling interval becomes:

\[
\Delta R = \sqrt{(P_{\text{north},i} - P_{\text{north},i-1})^2 + (P_{\text{east},i} - P_{\text{east},i-1})^2}
\]  

(A.5)

The instantaneous, apparent glide ratio, $L/D_i$ can be written as:

\[
L/D_i = \frac{\Delta R}{\Delta E/W}
\]  

(A.6)

This $L/D_i$ is now corrected for wind and any non-zero bank angle to arrive at the wings-level, no wind glide ratio $L/D_{0,i}$. A non-zero wind has the effect of changing the distance travelled over ground when compared to a no-wind situation. Hence, it needs to be removed to arrive at a no-wind glide ratio. The approach taken here is to break up the wind into a crosswind and tailwind component and remove their effect on $L/D_i$ separately.

Note: Vertical wind speeds strongly affect the glide ratio. However, since this wind cannot be inferred by the wind estimator, its effect cannot be removed. Therefore, vertical wind effects are absorbed in $L/D_{0,i}$.

Tailwind increases the no-wind, wings-level glide ratio from $L/D_{0,i} \approx \frac{V_a}{V_{\text{sink}}}$ in a no-wind situation to $L/D_{w,t} \approx L/D_{0,i} + \frac{V_{w,t}}{V_{\text{sink}}}$ in case of a perfect tailwind. The magnitude of tailwind $V_{w,t}$ can be deduced from the course angle, $\chi$, (angle between the inertial north axis and the ground speed vector), the wind direction, $\chi_w$, and the magnitude of the wind speed returned by the wind estimator as illustrated in Figure A-5. The equation for the tailwind component can be written as:

\[
V_{w,t} = V_w \cdot \cos(\chi_w - \chi)
\]  

(A.7)

If $\chi$ and $\chi_w$ differ by $180^\circ$, $V_{w,t}$ is a perfect headwind denoted by a negative sign. Non-zero crosswind results in a non-zero crab angle, $\chi_c$. This non-zero crab angle causes the distance flown over ground within a unit of time to be larger than the distance flown in the airmass within the same time interval. From Figure A-6, the crab angle, $\chi_c$, is the difference between the aircraft course, $\chi$, and the aircraft heading, $\psi$, i.e. $\chi_c = \chi - \psi$. Hence, $L/D_i$ uncorrected for crosswind effects will be larger than $L/D_{0,i}$. The correction factor becomes $\cos(\chi_c)$ since
\[ V_a = V_g \cdot \cos(\chi - \psi) \]. The effect of non-zero bank angle is to reduce the wings-level, no-bank

glide ratio to \( L/D_{0,i} \cdot (\cos \phi)^{3/2} \). Including tailwind, crosswind and bank angle effect, yields the following formula for \( L/D_{0,i} \) from \( (L/D)_i \).

\[ L/D_{0,i} = [(L/D)_i - \frac{V_w \cdot \cos(\chi - \chi_w)}{V_{sink}}] \cdot \frac{\cos(\chi - \psi)}{(\cos \phi)^{3/2}} \]  

(A.8)

The tailwind component has the largest effect on the change in glide ratio. A 20 kt tailwind increases the observed glide ratio by 22\% for an airspeed of 90 kt. A 20 kt crosswind for a 90 kt airspeed creates a 13\° crab angle, which increases the apparent glide ratio by only 3\%. A 30\° bank angle as nominally held during the turn segments decreases the glide ratio by 20\%. After having applied the corrections for wind and bank angle effects, the sampled \( L/D_{0,i} \) are
averaged using \( n = 10 \) samples to arrive at the estimated aircraft glide ratio \( L/D_0 \):

\[
L/D_0 = \frac{\sum_{i=1}^{n} (L/D)_{0,i}}{n}
\]  

(A.9)
A.2.3 L/D_expected Generator

Inputs

- $L/D_0$: aircraft no-wind, wings-level glide ratio
- $\text{LS.hdg}, \chi_r$: runway heading, i.e. approach direction, with respect to true north
- $\text{TP.leg}[\cdot].\text{course}$: array of courses of the traffic pattern legs in local North-East coordinates
- $V_{\text{glide}}, V_{bg}$: best glide airspeed as commanded to the autopilot, design constant
- $\text{state.wind}$: wind speed, $V_w$, and direction, $\chi_w$, as returned by the autopilot’s wind estimator
- nominal bank angle $\phi_0$: nominal bank angle to be used for turning

Outputs

- $\text{TP.L/D\_exp}[]$: array of L/D_expected, one per traffic pattern segment
- $\text{HP.L/D\_exp}[]$: array of L/D_expected, one per holding pattern segment

Trigger

called by higher level function, Performance Estimator, executes every 1 s

Description

This module takes the $L/D_0$ estimate returned from the L/D_0 Estimator and estimates the expected glide ratio for each leg of the traffic pattern, TP.L/D\_exp[], or holding pattern, HP.L/D\_exp[], using the expected crosswind and tailwind components, plus nominal bank
angle for turn segments. For this purpose, the L/D$_{\text{expected}}$ generator first calculates the course of each traffic pattern leg in global coordinates, which by rotating the courses given in TP.leg[].course by runway heading, $\chi_r$, a function that is performed by library module "convertToGlobalCoords". For the holding pattern, the course of the outbound leg is known to coincide with runway heading, $\chi_r$, and the inbound leg being the opposite direction, i.e. $\chi_r - 180^\circ$ normalized to range $[ -180^\circ, 180^\circ ]$. Then for each traffic pattern segment and holding pattern segment, the L/D$_{\text{Expected}}$ Generator calculates $\mathbf{L}.L/D_{\text{exp}}$ and $\mathbf{H}.L/D_{\text{exp}}$ respectively using the expected wind and bank angle along the segment. For turn segments, a nominal bank angle of $\phi_0$ is assumed and zero bank angle for straight line segments. For calculation of the crosswind and tailwind components during the turn segments, the center course between the course at the start of the turn and the finish of the turn is used. The crosswind manifests itself as a non-zero crab angle, $\chi_c$, which can be calculated from the wind speed, $V_w$ and direction $\chi_w$ from the following equation:

$$
\chi_c = \tan^{-1}\left(\frac{V_w \cdot \sin(\chi - \chi_w)}{V_a}\right)
$$

where the term $V_w \cdot \sin(\chi - \chi_w)$ denotes the magnitude of the crosswind. The tailwind component adds to the forward speed of the aircraft and increases the glide ratio by the term $\frac{V_w \cdot \cos(\chi - \chi_w)}{V_{\text{sink}}}$, with $V_{\text{sink}} \approx \frac{V_a}{L/D_0}$. The bank angle is included through term $(\cos \phi_0)^{3/2}$. In summary, L/D$_{\text{exp}}$ can be written as:

$$
L/D_{\text{exp}} = L/D_0 \cdot \frac{(\cos \phi_0)^{3/2}}{\cos(\chi_c)} + \frac{V_w \cdot \cos(\chi - \chi_w)}{V_{\text{sink}}}
$$

In order to include the effect of wind shear in boundary layer, an "effective" wind speed $V_{\text{w}}$ can be calculated for the traffic pattern legs, where wind shear is expected to have an impact. For a wind profile, where the wind speed changes linearly with altitude with no change in wind direction, the effective wind speed can be calculated from the wind speed expected at the end of the leg, $V_{w,2}$ and the expected change in wind speed from the start of the leg to the end of the leg, $\Delta V_w = V_{w,1} - V_{w,2}$. Both $V_{w,1}$ and $V_{w,2}$ can be inferred from the
assumed wind profile and the altitude of the aircraft at the start and end of the given leg. This altitude is known from the planned trajectory and the expected aircraft performance. The effective wind, $V_w$, becomes:

$$V_w = V_w,2 + \Delta V_w$$

(A.12)

### A.2.4 Path planner to Initial Point

**Inputs**

- **LS.IP**: location of the initial point in local North-East coordinates
- **LS.loc**: latitude and longitude of the runway threshold or location of the earliest desired touchdown point for alternate landing site
- **state.loc**: latitude and longitude of current aircraft position
- **range, R**: achievable range
- **L_tp,0**: length of the baseline traffic pattern
- **LS.hdg**: runway heading, i.e. approach direction, with respect to true north
- **engine status**: engine status

**Outputs**

- **LS.LS.Ok**: flag indicating whether the provided candidate landing site is feasible given the range, 1=success, 0=fail
- **TI.leg[].length**: array containing the length of each leg of the trajectory to the initial point
- **TI.leg[].fix**: array containing the waypoints that specify the trajectory to the initial point in local North-East coordinates
Trigger

called by higher level function, Landing Site Selector

Description

The task of the Path Planner to the Initial Point is to generate a feasible path from the current position, state.loc, to the initial point, LS.IP, so that the aircraft is aligned with the course of the downwind leg by the time of arrival at the initial point. The path planning is performed in the local North-East coordinate system. This is justified since the typical distances to a landing site are expected to be 100 miles or less, which creates an error of less than 0.002%, or 10 ft in total. This error is not noticeable in comparison to the margin included in the achievable range, R. With this approach, the path planner first calculates the aircraft’s current position in local North-East coordinates using library function "convertToLocalCoords". The initial point, LS.IP, is already provided in local coordinates. If there are no obstacles along the straight line path from the current position to the initial point, the Path Planner to the Initial Point chooses the simple straight line trajectory and inserts an additional leg that allows the aircraft to align with the downwind leg upon arrival at the initial point. If there are obstacles along the straight line path, a segmented trajectory will need to be synthesized. For this purpose, the path planner uses a digital terrain map. If the engine status = "Operational", the aircraft is capable of holding altitude so that the 3D problem reduces to a 2D problem. From this 2D map, the path planner deduces terrain elevations greater than the aircraft’s current altitude and represents these elevations as polygonal obstacles. In order to deduce an obstacle-free path from the aircraft’s current position to the selected destination, a representation of feasible trajectory segments are stored in form of a roadmap. This is most commonly done via a visibility graph. This graph connects each node of a polygon with the node of another polygon, given that the nodes are within line of sight, i.e. do not cross an obstacle. Figure A-7 illustrates this concept. Since the full visibility graph contains many edges that are not part of the shortest path solution,
the full visibility graph can be reduced to a *reduced visibility graph*. From this point, search algorithms, such as $A^*$, can be used to find the shortest trajectory from the start to the goal by setting the cost of each edge equal to the length of each edge. For the cases engine status "Out" or "Indeterminate", the simple 2D map will not suffice. A 3D representation will have to be generated taking the ceiling of each obstacle into account as opposed to only the obstacles that exceed the aircraft’s current altitude. Meuleau, Plaunt and Smith propose to use an *extended tangent graph*, which includes additional edges that take into account the aircraft’s capability to overfly an obstacle [3]. The path planner passes the final trajectory, as a list of waypoints in local coordinates via TI.leg[].fix. It stores the length of each leg in TI.leg[].length. If the total length of the synthesized trajectory is greater than the achievable range $R$, the path planner sets flag $LS.LS.Ok$ to 0 to indicate failure to find a feasible trajectory. Otherwise, the path planner set the flag equal to 1.

### A.2.5 Flight Plan Feasibility Monitor

**Inputs**

- **LTp,0**: total length of the baseline traffic pattern, design constant

- **TI.leg[].length**: array containing the length of each leg of the trajectory to the initial point

- **TI.L**: distance to the next waypoint
• **TI.currSeg:** integer identifying the current leg the aircraft is on

• **range, R:** achievable range as returned by the Range Calculator

**Outputs**

**LS.FP.Ok:** flag denoting whether the target touchdown point can still be reached given the current energy, E/W and the glide ratio estimate L/D₀, 1=OK, 0=NOK

**Trigger**

started and stopped by higher level state machine, executes every 3 s while running

**Description**

The Flight Plan Feasibility Monitor compares the achievable range, R, to the remaining distance to fly until the target touchdown point, which is simply the sum of the remaining distance to fly to the initial point plus the distance to fly in the pattern. The distance to fly to the initial point can be calculated from knowledge of the distance remaining on the current leg, TI.L, the length of the remaining legs stored in TI.leg[].length. The remaining legs start at index, (TI.currSeg + 1) = k, in array TI.leg[].length. The number of elements in array TI.leg[].length is denoted by n:

\[
L_{ip} = TI.L + \sum_{i=k}^{n} TI.leg[k].length \\
\text{\text{(A.13)}}
\]

This calculation neglects the fact, that the aircraft cuts the corner between two subsequent legs to perform a smooth intercept to the next leg. This can be neglected for small course changes from one leg to the next (< 45°) since it is expected that these errors are offset by cross-track errors the aircraft will inevitably incur along the trajectory. Overall, this approach overestimates the length of the actual trajectory, so that it errs on the conservative
side. For trajectories including many and many sharp turns, it is advisable to adjust this scheme in a subsequent design iteration. Hence, the flag LS.FP.Ok is set according to the following conditions:

\[
FP_{OK} = \begin{cases} 
  \text{NOK} = 0 & \text{if } R < L_{ip} + L_{tp,0} \\
  \text{OK} = 1 & \text{if } R \geq L_{ip} + L_{tp,0}
\end{cases}
\]

A.2.6 Excess Energy Calculator

Inputs

- **E/W**: current aircraft energy normalized by weight
- **HP.R, R**: range to go until start of the next segment of the holding pattern
- **HP.curSeg**: integer specifying the segment of the holding pattern, the aircraft is currently on (1=turn to outbound leg, 2 = outbound leg, 3 = turn to inbound leg, 4 = inbound leg)
- **LS.E_td, E_{td}/W**: desired aircraft energy on touchdown, design constant
- **r**: nominal turn radius, design constant
- **L**: length of a single leg of the traffic pattern.
- **HP.L/Dexp[]**: expected glide ratio on each segment of the holding pattern
- **HP.length, L**: length of the current inbound and outbound leg
- **TP.L/D_exp[]**: expected glide ratio on each segment of the traffic pattern
- **L**: length of a single leg of the traffic pattern
- **L_m**: runway distance consumed by undershoot margin

163
Outputs

- **HP.dE/W, ΔE/W**: difference in energy between the estimated aircraft energy upon passing the initial point next, and the energy required to start the baseline traffic pattern adjusted for the current aircraft performance.

Trigger

started and stopped by higher level state machine, executes every 1 s; paused during turn segments.

Description

The Excess Energy Calculator estimates the excess energy, above the energy required to start the traffic pattern, upon completion of the current loop. This excess energy is used to determine the length of the inbound leg and outbound leg of the holding pattern, and the number of loops to perform in the hold after completion of the current loop. For this purpose, it first computes the energy required to complete the current loop, $E_{ip}/W$. Calculation of the excess energy is halted during the turn segments. Hence, there are only two cases to consider: 1) The aircraft is on the inbound leg or 2) the aircraft is on the outbound leg.

It would have been possible to schedule only one calculation of the excess energy estimate by the time the aircraft starts the inbound leg. However, since there is the possibility of a zero length inbound leg, which causes the holding pattern update to arrive too late for a smooth transition onto the downwind leg should the update have commanded an exit from the holding pattern.

If the aircraft is on the inbound leg, $E_{ip}/W$ can simply be calculated from:

$$E_{ip}/W = R \cdot (L/D)_{exp,A} \tag{A.14}$$
where \( L/D_{exp,4} \) represents the estimated glide ratio on the inbound leg. If the aircraft is on the outbound leg, \( E_{ip}/W \) can simply be calculated by adding the turn from the outbound to the inbound leg and the inbound leg:

\[
E_{ip}/W = R \cdot (L/D)_{exp,2} + \pi \cdot (L/D)_{exp,3} + L \cdot (L/D)_{exp,4}
\]  

(A.15)

where \( L/D_{exp,2} \) represents the estimated glide ratio on the outbound leg and \( L/D_{exp,3} \), the expected glide ratio during the turn onto the inbound leg. The excess energy \( \Delta E/W \) hence becomes:

\[
\Delta E/W = E/W - E_{ip}/W - E_{tp}/W - E_{td}/W
\]  

(A.16)

with

\[
E_{ip}/W = \frac{L - r}{L/D_{exp,1}} + \frac{\pi}{2 \cdot L/D_{exp,2}} + \frac{L - 2r}{L/D_{exp,3}} + \frac{r \pi}{2 \cdot L/D_{exp,4}} + \frac{L - r}{L/D_{exp,5}}
\]  

(A.17)

where \( E_{tp}/W \) is the energy required to fly the baseline traffic pattern with the current expected glide performance.

### A.2.7 Holding Pattern Generator/Updater

**Inputs**

- \( HP.dE/W, \Delta E/W \): difference in energy between the estimated aircraft energy upon passing the initial point next, and the energy required to start the baseline traffic pattern adjusted for the current aircraft performance
- \( HP.L/D_{exp}[] \): expected glide ratio on each segment of the holding pattern
- \( r \): nominal turn radius, design constant
- \( \phi_0 \): nominal bank angle, design constant
- \( L/D_0 \): estimated no-wind, wings-level glide ratio
Outputs

- **HP.#loops, n**: number of loops to spend in holding pattern after completion of the current loop

- **HP.length, L**: length of the straight line segments of the inbound and outbound leg of the holding pattern

Trigger

started and stopped by higher level state machine, executes every 1 s; paused during turn segments

Description

The Holding Pattern Generator/Updater recalculates the number of holding pattern loops to follow, \( n \), and length of the straight line segments, \( L \), every time it receives an update on the energy that is to be depleted in the holding pattern after completion of the current loop, HP.dE/W. The current loop of the holding pattern is not altered, since the holding pattern generator cannot update the length of the current hold loop, while the FMS is tracking it. The integer number of loops, \( n \), that can be completed with the excess energy, \( \Delta E/W \), can be found from:

\[
 n = \left\lfloor \frac{\Delta E/W}{E_{\text{loop}}/W} \right\rfloor \tag{A.18}
\]

where

\[
 E_{\text{loop}}/W = \frac{2r \pi}{(\cos \phi_0)^{3/2} L/D_0} \tag{A.19}
\]

and \( \lfloor x \rfloor \) represents rounding down. The length of the straight line segments, \( L \), becomes:

\[
 L = \frac{\Delta E/W - n \cdot E_{\text{loop}}/W}{n(LL_{exp,2} + L/L_{exp,A})} \tag{A.20}
\]
A.2.8 Traffic Pattern Trajectory Generator

Inputs

- **E/W**: current aircraft energy
- **LS.E_td, \ E_{td}/W**: desired touchdown energy
- **r**: nominal turn radius, design constant
- **L**: length of the baseline traffic pattern leg, design constant
- **L_m**: undershoot margin, determines the position of the target touchdown point past the runway threshold, design constant
- **LS.dir**: direction of the traffic pattern, either right or left
- **TP.L/D_exp[]**: expected glide ratio for each segment of the traffic pattern in the order of downwind leg, turn 1, base leg, turn 2, final leg
- **r**: nominal turn radius

Outputs

- **TP.leg[].fix**: array of the leg fixes in local North-East coordinate system
- **TP.leg[].course**: course of each leg in local North-East coordinate system
- **TP.leg[].length**: length of each traffic pattern leg in m

Trigger

called by higher level state machine, runs once upon exit of holding pattern
Description

This function takes the energy error between the current energy level, \( E/W \), and the energy required to fly the baseline traffic pattern, \( E_{tp}/W \), and elongates or shortens the trajectory by moving the base leg fix along a line angled at 45° to the base leg by magnitude \( d \), which is dependent on energy error \( \Delta E/W \) and \( L/D_{av} \). The energy error is:

\[
\Delta E/W = E/W - E_{tp}/W - E_{td}/W
\]

(A.21)

where

\[
E_{tp}/W = \frac{L - r}{L/D_{exp,1}} + \frac{r \pi}{2 \cdot L/D_{exp,2}} + \frac{L - 2r}{L/D_{exp,3}} + \frac{r \pi}{2 \cdot L/D_{exp,4}} + \frac{L - r + L_m}{L/D_{exp,5}}
\]

(A.22)

where \( L/D_i \) are the expected glide ratios on the downwind leg, turn to base leg, base leg, turn to final leg and final leg from array TP.L/D\_exp[]. The average glide ratio for the complete traffic pattern can be found from:

\[
L/D_{av} = \frac{(L - r) \cdot L/D_d + (L - 2r) \cdot L/D_h + (L - r + L_m) \cdot L/D_f + \frac{1}{2} r \pi \cdot (L/D_{tl} + L_T)}{3L - 4r + L_m + r \pi}
\]

(A.23)

The amount, \( d \), by which the base leg fix should be moved is given by:

\[
d = 5.0 \cdot \frac{L/D_{av}}{L/D_{POH}} \cdot \Delta E/W
\]

(A.24)

This formula was derived from a graph of the \( d \) vs \( \Delta E/W \) for various \( L/D_{av} \) as shown in Figure A-8. The graph results from comparing the length of the baseline traffic pattern with the length of the traffic pattern achieved for \( d \) ranging from -3,000 ft to + 3,000 ft. For the detailed equations, refer to module Traffic Pattern Trajectory Updater in section A.2.11.

For \( L/D_{av} = 10 \), the maximum and minimum energy error, with which the holding pattern can be exited is \( + \frac{1}{2} E_{loop}/W = +433 \text{ ft} \) or \( - \frac{1}{2} E_{loop}/W = -433 \text{ ft} \) respectively. From equation A.24, the distance to move the base leg fix for energy error +/- 433 ft and \( L/D_{av} = 10 \) is
Figure A-8: Distance, $d$, to move base leg fix to correct energy error $\Delta E/W$

$\pm 2200$ ft.

Figure A-9 shows the updated trajectory (red curve) compared to the baseline trajectory (blue curve) for a runway oriented due East.

Figure A-9: Baseline and updated trajectory for max. and min. energy error expected (rotated by $-90^\circ$ from local coordinate system)

After having deduced the amount, $d$, that the base leg fix should be moved, the Traffic Pattern Trajectory Generator fills array TP.leg[] with information on the updated location of each fix, TP.leg[].fix, the course of each leg, TP.leg[].course, and the length of each leg, TP.leg[].length, in the local coordinate system. The Traffic Pattern Trajectory Generator calculates the location of the updated leg fixes based on knowledge of the location of the leg
fixes of the baseline traffic pattern, the direction of the traffic pattern and the distance that the base leg fix is moved, \( d \). The result is:

\[
\begin{bmatrix}
F_{D,north} & F_{D,east} \\
F_{B,north} & F_{B,east} \\
F_{F,north} & F_{F,east}
\end{bmatrix}
= \begin{bmatrix}
0 & -5,000 \\
-5,000 - d \cos(45^\circ) & -5,000 - d \sin(45^\circ) \\
-5,000 - d \cos(45^\circ) & 0
\end{bmatrix}, \text{ if LS.dir = "Left"}
\]

\[
\begin{bmatrix}
F_{D,north} & F_{D,east} \\
F_{B,north} & F_{B,east} \\
F_{F,north} & F_{F,east}
\end{bmatrix}
= \begin{bmatrix}
0 & 5,000 \\
-5,000 - d \cos(45^\circ) & 5,000 + d \sin(45^\circ) \\
-5,000 - d \cos(45^\circ) & 0
\end{bmatrix}, \text{ if LS.dir = "Right"}
\]

The downwind leg fix, \( F_D \) coincides with the initial point and is not changed during the traffic pattern trajectory generation process.

The course of each baseline traffic pattern leg is known from the choice of the local North-East coordinate system. They are:

\[
\chi_{d,0} = 180^\circ \text{ for LS.dir = "Right" or "Left"}
\]

\[
\chi_{f,0} = 0^\circ \text{ for LS.dir = "Right" or "Left"}
\]

\[
\chi_{b,0} = \begin{cases} 
90^\circ & \text{for LS.dir = "Left"} \\
-90^\circ & \text{for LS.dir = "Right"}
\end{cases}
\]

The course of the base leg and final leg are unchanged by the trajectory generation process and hence coincide with the course of the baseline traffic pattern base leg and final leg. The new course of the downwind leg can be derived from the location of the new base leg fix and the location of the downwind leg fix using the tangent function and considering the quadrant:

\[
\chi_d' = \arctan\left( \frac{F_{B,east}' - F_{D,east}}{F_{B,north}' - F_{D,north}} \right)
\]  

(A.25)
Hence,

\[
\text{TP.leg[].course} = \begin{bmatrix}
\chi_d \\
\chi_b \\
\chi_f \\
\end{bmatrix} = \begin{bmatrix}
\chi_d' \\
90^\circ \\
0^\circ \\
\end{bmatrix}, \text{ if LS.dir = "Left"}
\]

\[
\text{TP.leg[].course} = \begin{bmatrix}
\chi_d \\
\chi_b \\
\chi_f \\
\end{bmatrix} = \begin{bmatrix}
\chi_d' \\
-90^\circ \\
0^\circ \\
\end{bmatrix}, \text{ if LS.dir = "Right"}
\]

The length of each leg can be found from geometry:

\[
\text{TP.leg[].length} = \begin{bmatrix}
\sqrt{(5,000 + d \cos(45^\circ))^2 + (d \cos(45^\circ))^2} \\
5,000 + d \cos(45^\circ) \\
6,000 + d \cos(45^\circ) \\
\end{bmatrix}
\]

A.2.9 Energy Curve Manager

Inputs

- TP.leg[].length: length of each traffic pattern leg
- \( r \): nominal turn radius, design constant
- \( \text{LS.E_td, } E_{td}/W \): desired energy upon touchdown
- TP.L/D_exp[]: expected glide ratio for each segment of the traffic pattern

Outputs

M: array of a struct containing both slope and intercept of each local energy vs. range-to-go-until-next-segment function in the following format. Refer to Figure A-10 for usage of
the symbols.

\[ M = \begin{bmatrix}
  b_D & a_D \\
  b_{T1} & a_{T1} \\
  b_B & a_B \\
  b_{T2} & a_{T2} \\
  b_F & a_F 
\end{bmatrix} \]

\[ M_u = \begin{bmatrix}
  b_{u,D} & a_{u,D} \\
  b_{u,T1} & a_{u,T1} \\
  b_{u,B} & a_{u,B} \\
  b_{u,T2} & a_{u,T2} \\
  b_{u,F} & a_{u,F} 
\end{bmatrix} \]

\[ M_l = \begin{bmatrix}
  b_{l,D} & a_{l,D} \\
  b_{l,T1} & a_{l,T1} \\
  b_{l,B} & a_{l,B} \\
  b_{l,T2} & a_{l,T2} \\
  b_{l,F} & a_{l,F} 
\end{bmatrix} \]

Trigger

started and stopped by higher level state machine, executes every 1 s on receipt of updated TP.L/D.exp and TP.leg[].length

Description

The slope \( b \) and intercept \( a \) specify slope and intercept of a local energy vs. range to go curve as shown in Figure A-10. Each local, linear equation can be written as:

\[ f_i(R) = b_i \cdot R_i + a_i = E/W \]  \hspace{1cm} (A.26)
where $b$ is the reciprocal of the expected glide ratio, $\text{TP.L}/\text{D}_{\exp}[i]$, on the leg specified by subscript, $i$. For example, $b_F$, the slope on the final leg can be found from $L/D_{\exp,f} = \text{TP.L}/\text{D}_{\exp}[1]$ using the array indices as defined in table 3.2.

$$b_F = \frac{1}{L/D_{\exp,f}} \quad (A.27)$$

$a$ denotes the desired energy at the end of the leg specified by the subscript, $i$, and can be found by working from left to right on the E/W vs. R curve starting at the desired touchdown energy, $E_{td}/W$ as follows:

$$a_F = (E_{td}/W) \quad (A.28)$$

$$a_{T2} = (E_{td}/W) + b_F \cdot L_F \quad (A.29)$$

and so on. The slopes, $b_i$, and intercepts $a_i$ are illustrated in Figure A-10.

![Local energy vs. range curve function representation](image)

Figure A-10: Local energy vs. range curve function representation

In addition to the energy vs. range curve itself, the Energy Curve Manager also maintains a representation of the upper and lower energy boundaries which are shown in Figure A-11. The tolerance bounds are set in a way that the aircraft may at any point in time have $x=5\%$ more energy than needed to reach the turn onto the final leg and $y=3\%$ less energy than
required to reach this turn. The tolerance bounds are stored similarly to the target energy vs. range-to-go curve as an array of slopes and intercepts. The subscript $u$ refers to the "upper" boundary curve, and the subscript $l$ refers to the "lower" boundary curve.

$$f_u(R) = b_{u,i}R + a_{u,i} \quad (A.30)$$

$$f_l(R) = b_{l,i}R + a_{l,i} \quad (A.31)$$

### A.2.10 Traffic Pattern Energy Error Detector

**Inputs**

- **TP currSeg**: counter indicating the current segment of the traffic pattern trajectory the aircraft is on

- **E/W**: aircraft energy normalized by weight

- **M**: array of slopes and intercepts for each leg as returned by "Energy Curve Manager"
• **M_u**: array of slopes and intercepts for the upper energy boundary as returned by "Energy Curve Manager"

• **M_l**: array of slopes and intercepts for the lower energy boundary as returned by "Energy Curve Manager"

• **TP.R**: range to go until the start of the next segment

**Outputs**

• **ΔE/W**: energy error in m

**Trigger**

called by higher level function, Energy Management, executes every 1 s

**Description**

This function uses input TP.currSeg to decide, which array element of M_u and M_l is applicable to determining a potential energy error. After having extracted the correct slopes and intercepts from M, M_u, and M_l, the Energy Error Detector determines if the current energy is above or below one of the boundaries. If the current energy is in within the boundaries, the Energy Error Detector returns \( \Delta E/W = 0 \), which signals the higher level Energy Management function that no update is required. If the energy is above the upper boundary or below the lower boundary, the Energy Error Detector returns the magnitude and sign of the error, negative representing lack of energy, in output \( \Delta E/W \).

\[
\Delta E/W = \begin{cases} 
0 & \text{if } b_{l,i}R + a_{l,i} \leq E/W \leq b_{u,i}R + a_{u,i} \\
E/W - (bR + a) & \text{if } E/W > b_{u,i}R + a_{u,i} \\
E/W - (bR + a) & \text{if } E/W < b_{l,i}R + a_{l,i}
\end{cases}
\]
A.2.11 Traffic Pattern Trajectory Updator

Inputs

- **TP.currSeg**: current segment the aircraft is on
- **ΔE/W**: energy error
- **TP.L/D.exp[]**: expected glide ratio for each segment of the traffic pattern
- **r**: nominal turn radius, design constant
- **TP.leg[]**: fix, course and length of each traffic pattern leg
- **TP.L, L**: distance to go until the next leg fix

Outputs

- **TP.leg[]**: updated fix, course and length of each traffic pattern leg
- **TP.updateFlag**: sets this flag, if a trajectory update has been performed to signal the Traffic Pattern Follower to update the active leg within the FMS

Trigger

called by higher level function, Energy Management; executes whenever there is a non-zero energy error from the Energy Error Detector

Description

This function takes input TP.currSeg and checks whether a trajectory update is currently allowed. Trajectory updating is possible only on the downwind leg and the base leg with the current design.
Energy error correction on downwind leg

While on the downwind leg, energy errors are corrected by moving the downwind leg fix along course $\chi_{c1}$ which is placed at a 45° angle to the course of the base leg. The trajectory adjustment strategy while on the downwind leg was illustrated in Figure 2-14 in chapter 2.1.9. In order to correct energy error $\Delta E/W$, the base leg fix, $F_B$ would have to be moved along course $\chi_{c1}$ by magnitude $d$. The direction in which $F_B$ is moved depends on the sign of $d$. A positive $d$ moves the base leg fix outwards, which corrects a positive energy error. The final leg fix is consequently moved by $d \cdot \cos(45^\circ)$ along the opposite direction of the final leg course.

The sign and magnitude of $d$ can be found from the energy error $\Delta E/W$, the distance to the base leg fix, $L$, the angle the downwind leg is making with $\chi_{c1}$, $\alpha_p$ and the average glide ratio to be expected on the remaining trajectory $L/D_{av}$. As mentioned previously, no explicit form for $d$ in terms of $\Delta E/W$, $L$ and $\alpha_p$ can be found. However, the implicit equation can be solved for $\Delta E/W$ for vectors of $d$, $L$ and $\alpha_p$ and the result stored in a lookup table. The implicit equation for $d$ can be found from comparing the energy required to fly the baseline trajectory to an elongated or shortened trajectory $E/W_{tp}$ through $d$.

$$\Delta E/W = E/W_{tp} - E/W_{tp,0} = \frac{L' + L_B + 2d \cos(45^\circ) - 2t_d + r \cdot \Delta \chi}{L/D_{av,tp}} - \frac{L + L_B - 2t_{d,0} + r \cdot \Delta \chi_0}{L/D_{av,tp0}} \tag{A.32}$$

with $L'$ being the distance from the current position to the new base leg fix, $t_d$ the distance before the base leg fix that the turn to intercept is started, and $\Delta \chi$ and $\Delta \chi_0$ being the updated and baseline course change from the downwind leg to the base leg. To simplify the calculation, it is assumed that $L/D_{av,tp} \approx L/D_{av,tp0} = L/D_{av}$. Hence,

$$\Delta E/W = \frac{1}{L/D_{av}} (L' - 2t'_d + r \cdot (135^\circ - \alpha) + 2d \cos(45^\circ) - (L - 2t_d + r \cdot (135^\circ - \alpha_p))) \tag{A.33}$$
$L/D_{av}$ can be written as:

$$L/D_{av} = \frac{L(L/D)_{d} + L_B(L/D)_b + L_F(L/D)_{f}}{L + L_B + L_F} \quad (A.34)$$

$t_d'$ and $t_d$ can be found from $\alpha$ and $\alpha_p$ respectively:

$$t_d = r \sqrt{\frac{1 + \cos \gamma}{1 - \cos \gamma}} \quad (A.35)$$

where $\gamma = 180^\circ - \Delta \chi = 135^\circ - \alpha$

$\alpha$ can be found from $\alpha_p$:

$$\alpha = \arcsin\left(\frac{L \sin(\alpha_p)}{L'}\right) \quad (A.36)$$

where $L' = \sqrt{L^2 + d^2 - 2Ld \cos(180^\circ - \alpha_p)} = \sqrt{L^2 + d^2 + 2Ld \cos(\alpha_p)}$

The result can be graphed on $d$ vs. $E/W$ plots, while first holding $\alpha_p$ constant and substituting a discrete set of $L$. This is illustrated in Figure A-12.

Figure A-12: $d$ vs. $E/W$ plot for various $L$ with $\alpha_p = \text{const} = 45^\circ$
For the second plot, \( L \) is held constant and \( \alpha_p \) takes on a set of discrete values. This is illustrated in Figure A-13. All of this information is stored in a lookup table for \( L/D_{av} = 10 \).

Figure A-13: \( d \) vs. E/W plot for various \( \alpha_p \) with \( L = \text{const} = 5,000 \text{ ft} \)

To find \( d \) for a specific combination of \( \Delta E/W \), \( L \) and \( \alpha_p \), the data in the lookup table is interpolated using a linear interpolation function that is called by passing the known vectors of \( \Delta E/W \), \( L \) and \( \alpha_p \), the lookup table, which is stored as a 3D matrix and the combination of \( \Delta E/W \), \( L \) and \( \alpha_p \) for which \( d \) should be found. The value \( d_0 \) returned from the table is scaled for an \( L/D_{av} \) of 10. Hence, the actual \( d \) for the actual \( L/D_{av} \) can be found from:

\[
d = d_0 \frac{L/D_{av}}{10}
\]  

(A.37)

When calculating \( d \) to move the base leg fix, it is necessary to observe whether the fix can be moved by the amount required without violating the condition that the course of the base leg remains unchanged. For a positive \( d \), i.e. elongating the trajectory, this is not a problem. When shortening the trajectory, however, the magnitude of \( d \) is limited by \( d_{max} = \frac{L(\sin(135^\circ) - \alpha_p)}{\sin(45^\circ)} \) from geometry. Any remaining energy error will therefore be carried over to the base leg, where it can be corrected for using the following strategy. Refer to function ”Traffic Pattern Generator” for details on how to update the traffic pattern leg
fixes, TP.leg[].fix, the courses, TP.leg[].course and the length of each leg, TP.leg[].length.
Energy error correction on base leg

While on the base leg, energy errors are corrected by inserting a fourth leg, called the "Final Leg Intercept Leg" as shown in Figure 2-15 in chapter 2.1.9 and moving the final leg intercept fix, $F_{FI}$ along course $\chi_{c2}$, which is placed at a 45° angle to the course of the base leg. The final leg fix, $F_F$ is moved to a fixed position 500 ft off the runway threshold. During future updates, only the final leg intercept fix is moved while the location of the final leg fix remains unchanged as illustrated in Figure A-14, which repeats Figure 2-16 from chapter 2.1.9 for convenience. Due to the added leg, the distance that the final leg intercept fix should be moved, $d$, is not only dependent on $L$ and $\alpha_p$ as was the case for the downwind leg, but also dependent on the distance that the final leg intercept fix has already been moved from the location of the baseline final leg intercept fix $F_{FI,0}$, termed $d_{total}$. Similarly to the downwind

$$d_{total} = |F_{FI,0} - F_F'|$$

![Figure A-14: Geometry used to determine difference in path length between the current and updated trajectory](image)

The energy error that can be corrected on the base leg is determined by the difference in
the length of trajectory:

\[
\Delta E/W = \frac{1}{L/D_{\text{av}}}(L' + L'_{FI} - 2t'_i + r \cdot (180^\circ - (\alpha + \beta))) - (L + L_{FI} - 2t_f + r \cdot (180^\circ - (\alpha_p + \beta_p)))
\]  

(A.38)

\[
L/D_{\text{av}} \text{ can be written as:}
\]

\[
L/D_{\text{av}} = \frac{L(L/D)_b + L_{FI}(L/D)_{fi}}{L + L_{FI}}
\]  

(A.39)

t'_{fi} \text{ and } t_{fi} \text{ can be found from } \alpha \text{ and } \alpha_p \text{ respectively:}

\[
t_{fi} = r \sqrt{\frac{1 + \cos \gamma}{1 - \cos \gamma}}
\]  

(A.40)

where \(\gamma = 180^\circ - \Delta \chi = 180^\circ - (\alpha + \beta)\)

\(\alpha\) can be found from \(\alpha_p\):

\[
\alpha = \arcsin\left(\frac{L \sin(\alpha_p)}{L'}\right)
\]  

(A.41)

where \(L' = \sqrt{L^2 + d^2 - 2Ld \cos(\alpha_p)}\)

\(\beta\) can be found from \(\beta_p\):

\[
\alpha = \arcsin\left(\frac{L_{FI} \sin(\beta_p)}{L'_{FI}}\right)
\]  

(A.42)

where \(L'_{FI} = \sqrt{L^2_{FI} + d^2 - 2L_{FI}d \cos(\beta_p)}\) with \(L_{FI} = \sqrt{L^2_I + d^2_{\text{total}} - 2L_{FI}d_{\text{total}} \cos(135^\circ)}\) and \(\beta_p = \arcsin\left(\frac{L_{FI} \sin(135^\circ)}{L_{FI}}\right)\).

The result can be graphed on \(d\) vs. \(E/W\) plots, while first holding \(d_{\text{total}}\) and \(\alpha_p\) constant and substituting a discrete set of \(L\). This is illustrated in Figure A-15.

For the second plot, \(L\) and \(d_{\text{total}}\) are held constant and \(\alpha_p\) takes on a set of discrete values. This is illustrated in Figure A-16.

Finally, \(L\) and \(\alpha_p\) are held constant while \(d_{\text{total}}\) takes on a set of discrete values. Figure A-17 illustrates this.
For the trajectory updating strategy on the base leg, it is possible that the course change from the updated base leg to the final intercept leg is greater than 135°. This will cause the FMS to cap turn anticipation in favor of overshooting the subsequent leg. This avoids...
excessive corner cutting. Figure A-18 shows the effect of this scheme on the required \( d \) for a given \( \Delta E/W \). While \( d \) increases rapidly with increasing \( \Delta E/W \) before turn anticipation is capped, it increases noticeably slower after the capping, which makes intuitive sense from the illustration shown on the right of Figure A-18. On the other end, a large negative \( d \) will at some point cause an elongation of the trajectory rather than a shortening, which occurs as soon as the course of the updated base leg becomes equal to the course of the updated final leg intercept leg. This point signifies the maximum lack of energy, \( \Delta E/W \), that can be corrected using the trajectory updating strategy on the base leg. As with the downwind leg, all this data is stored in a multi-dimensional array, which acts as a lookup table, and is used to calculate an interpolated value of \( d \) for any combination of \( \Delta E/W \), \( L \), \( \alpha_p \) and \( d_{total} \). The value returned, \( d_0 \) is again scaled for an \( L/D_{av} \) of 10, so that \( d \) can be found from:

\[
d = d_0 \frac{L}{D_{av}} \frac{10}{10} \tag{A.43}
\]
From knowledge of $d$ and the location of the current leg fixes, the leg fixes can be updated as follows:

$$
\text{TP.leg}[\text{fix}'] = \begin{bmatrix}
    F'_{D,\text{north}} & F'_{D,\text{east}} \\
    F'_{B,\text{north}} & F'_{B,\text{east}} \\
    F'_{FI,\text{north}} & F'_{FI,\text{east}} \\
    F_{F,\text{north}} & F_{F,\text{east}}
\end{bmatrix}
= \begin{bmatrix}
    0 & -5,000 \\
    P_{\text{north}} & P_{\text{east}} \\
    F_{FI,\text{north}} - d \cos(45^\circ) & F_{FI,\text{east}} + d \sin(45^\circ) \\
    0 & -500
\end{bmatrix}, \text{ if LS.dir = "Lef}
$$

$$
\text{TP.leg}[\text{fix}'] = \begin{bmatrix}
    F'_{D,\text{north}} & F'_{D,\text{east}} \\
    F'_{B,\text{north}} & F'_{B,\text{east}} \\
    F'_{FI,\text{north}} & F'_{FI,\text{east}} \\
    F_{F,\text{north}} & F_{F,\text{east}}
\end{bmatrix}
= \begin{bmatrix}
    0 & 5,000 \\
    P_{\text{north}} & P_{\text{east}} \\
    F_{FI,\text{north}} - d \cos(45^\circ) & F_{FI,\text{east}} - d \sin(45^\circ) \\
    0 & -500
\end{bmatrix}, \text{ if LS.dir = "Rig}
$$

The base leg fix, $F_B$ will be changed to the current position. The final intercept leg fix, $F_{FI}$ is moved along course $\chi_{c2}$ as shown in Figure A-14 and the final leg fix, $F_F$ remains
unchanged.

Note: If this is the first update from the baseline traffic pattern trajectory, the final leg intercept leg needs to be inserted first and \( F_F \) moved to the fixed position 500 ft off the runway threshold.

The updated course of the base leg and final intercept are derived from the location of the new base leg fix, \( F_B' \), the final leg intercept fix, \( F_{FI}' \) and the final leg fix, \( F_F \). The final leg course is unchanged by the trajectory generation process. The downwind leg course is irrelevant at this point and left as is for simplicity. In summary,

\[
\text{TP.leg}[][\text{course}] = \begin{bmatrix}
\chi_d \\
\chi_b \\
\chi_{fi} \\
\chi_f
\end{bmatrix} = \begin{bmatrix}
\chi_d \\
\text{arctan} \left( \frac{F_{FI,\text{east}} - F_{B,\text{east}}}{F_{FI,\text{north}} - F_{B,\text{north}}} \right) \\
\text{arctan} \left( \frac{F_{F,\text{east}} - F_{FI,\text{east}}}{F_{F,\text{north}} - F_{FI,\text{north}}} \right) \\
0^\circ
\end{bmatrix}
\]

The length of each leg can be found from geometry:

\[
\text{TP.leg}[][\text{length}] = \begin{bmatrix}
L_D \\
\sqrt{(F_{fi,\text{east}} - P_{\text{east}})^2 + (F_{fi,\text{north}} - P_{\text{north}})^2} \\
\sqrt{(F_{f,\text{east}} - F_{fi,\text{east}})^2 + (F_{f,\text{north}} - F_{fi,\text{north}})^2} \\
1,500
\end{bmatrix}
\]
A.3 Modules of Layer Trajectory Following

A.3.1 Trajectory to Initial Point Follower

Inputs

- TI.leg[].fix, \( P \): array of waypoints specifying the trajectory to the initial point in the local North-East coordinate system

- LS.hdg, \( \chi - r \): runway heading, i.e. approach direction, with respect to true north

- LS.loc, \( \mu_0, \nu_0 \): latitude and longitude of the runway threshold or location of the earliest desired touchdown point for alternate landing site

- state.loc: latitude and longitude of current aircraft position

Outputs

- TF legs/ DF leg: track-to-fix leg for trajectory to the initial point, direct-to-fix leg for direct flight to an unpopulated area

- TI.L: distance to go until the next leg fix

- TI.currSeg: counter indicating the current leg of the trajectory to the initial point the aircraft is on

Trigger

started and stopped by higher level state machine, executes continuously while active
Description

The function of the Trajectory to Initial Point Follower is to translate the trajectory given in local North-East coordinates to the global latitude-longitude coordinate system using the runway heading, $\chi_r$, and location of the runway threshold, $(\mu_0,t_0)$. Library function "convertToGlobalCoords" is used to perform this task. Furthermore, this function keeps track of the aircraft’s progress along the trajectory. It passes TI.currSeg, indicating the array index of the current leg, and TLL, the distance to go until the start of the next leg, to the Feasibility Monitor. Counter TI.currSeg is incremented every time the aircraft initiates a turn to intercept the next leg. This initiation can be detected by checking whether the aircraft has passed the line drawn perpendicular to the current course at the point where the FMS initiates the turn to allow a smooth intercept to the next leg. This perpendicular line is found using library function ”getTurnEntryLine”, whereas passing the line is checked by library function ”lineCrossed”. TLL is calculated using library function ”calcDistToGo”, which averages cross-track and along-track error with the straight line distance to the next waypoint to estimate the length of the actual trajectory to the next waypoint.

A.3.2 Holding Pattern Follower

Inputs

- **LS.IP**: location of the initial point in local North-East coordinates
- **LS.loc**: latitude and longitude of the runway threshold or location of the earliest desired touchdown point for alternate landing site
- **LS.hdg**: runway heading, i.e. approach direction, with respect to true north
- **LS.dir**: direction of the traffic pattern, either right or left
- **state.loc**: latitude and longitude of current aircraft position
- **HP.length**: length of the straight line segments of the holding pattern
- **HP.#loops**: number of loops to remain in the holding pattern after completion of the current loop
- **r**: nominal turn radius, design constant

**Outputs**

- **HP.R**: range to go until the end of the straight line segment, outputs zero during turn segments
- **HP.currSeg**: counter indicating the current segment of the holding pattern the aircraft is on (1=turn to outbound leg, 2=outbound leg, 3=turn onto inbound leg, 4=inbound leg)
- **HM leg**: specification of leg "hold-until-manual-termination" consisting of initial point, HM.fix in latitude and longitude coordinates, holding pattern direction, HM.dir, course of the outbound leg, HM.hdg, and length of the straight line segment, HM.length
- **exit**: flag indicating to the FMS to exit the holding pattern and to continue onto the downwind leg, which has been pre-stored

**Trigger**

started and stopped by higher level state machine, executes continuously when active

**Description**

The function of the Holding Pattern Follower is to translate the inputs describing the required holding pattern into the HM leg format required by the FMS and to initiate the "manual" termination once the Holding Pattern Generator/Updator signals exit from the
holding pattern by setting HP.#loops=0. The holding pattern fix, HM.fix is equal to the initial point, LS.IP. The FMS requires this fix to be specified in global latitude and longitude coordinates, which is performed using library function ”convertToGlobalCoords”. The course of the outbound leg, HM.hdg is simply equal to the course of the runway heading, so that HM.hdg=LS.hdg. The holding pattern direction is opposite the traffic pattern direction, so that HM.dir = -LS.dir. The length of the straight line segment is provided as an input so that HM.length = HP.length.

The distance to go until the start of the next turn segment can be found from the current location, state.loc and the knowledge of the two end-points of the straight line segments, IP and O as shown in Figure A-19. Point IP is known from LS.IP. Point O can be found from knowledge of the traffic pattern direction, the length of the straight line segment L = HP.length and the turn radius, r.

For a left traffic pattern, the coordinates of O in the local North-East coordinate frame become:

\[
OP_{i,north} = L \tag{A.44}
\]

\[
OP_{i,east} = -(5,000 \text{ ft} + 2r) \tag{A.45}
\]

Figure A-19: Geometry used to determine location of IP and OP
For a right traffic pattern, the coordinates are:

\[
OP_{r,north} = L \quad \text{(A.46)}
\]

\[
OP_{r,east} = 5,000 \text{ft} + 2r \quad \text{(A.47)}
\]

The distance to go to IP or OP while the aircraft is on the inbound or outbound leg respectively, HP.R, can be calculated using library function ”calcDistToGo” after the aircraft position has been converted to local coordinates using library function ”convertToLocalCoords”. The Holding Pattern Follower keeps track of the segment, the aircraft is currently on by monitoring whether the aircraft has passed point IP, OP, I, and O using library functions ”getTurnEntryLine” and ”lineCrossed”, which first draw a line perpendicular to the current course through the point of interest and subsequently checks the aircraft’s position with respect to this line, to determine whether the aircraft has crossed the line. Finally, the Holding Pattern Follower sets flag ”exit”, which emulates the operator to manually terminate the holding pattern, once input HP.#loops equals 0, and the aircraft has intercepted the inbound leg. The logic can be written as follows:

\[
\text{exit} = \begin{cases} 
0 & \text{if } \text{HP.#loops} \neq 0 \\
1 & \text{if } \text{HP.#loops} = 0 \& \text{HP.currSeg} = 4
\end{cases}
\]

### A.3.3 Traffic Pattern Follower

**Inputs**

- **LS.loc:** latitude and longitude of the runway threshold or location of the earliest desired touchdown point for alternate landing site

- **LS.hdg:** runway heading, i.e. approach direction, with respect to true north

- **TP.leg[]:** array of trajectory legs containing waypoints, TP.leg[].fix and course of each leg, TP.leg[].course
• **TP.updateFlag**: flag indicating whether the Traffic Pattern Trajectory Updator has changed the current trajectory

• **state.loc**: latitude and longitude of current aircraft position

• **r**: nominal turn radius, design constant

**Outputs**

• **FM leg**: specification of single ”fix-to-manual termination” legs which require information of the course to track, FM.hdg, and the start fix, FM.fix

• **TP.R**: range to go until the start of the next segment

• **TP.currSeg**: counter indicating the current segment the aircraft is on (1=downwind leg, 2=turn to base, 3=base leg, 4=turn to final intercept leg (if applicable), 5=final intercept leg (if applicable), 6=turn onto final, 7=final leg)

• **TP.L**: distance until the next leg fix

**Trigger**

started and stopped by higher level state machine, runs continuously while active

**Description**

This function passes one FM leg at a time to the FMS for tracking and times the switching from one leg to the next, so that a smooth intercept of the next leg is possible without overshoot. The distance from the next fix, at which the turn should be initiated, is determined by the course change from the current to the next leg, $\Delta \chi$, which can be found from TP.leg[].course and the nominal turn radius, r. This value is found using library function ”calcTurnDist” and is termed $t_d$. Using, $t_d$, library functions ”getTurnEntryLine” and
"lineCrossed" are used to determine whether the aircraft has passed the point, at which the turn should be initiated. Furthermore, the Traffic Pattern Follower passes updates of the current leg, should the Traffic Pattern Trajectory Updater raise the TP.updateFlag.

Note: The energy error tolerance bounds were chosen in such a way that updates do not occur more frequently than every 3-5 s if the performance estimation error remains within +/-10%, which is deemed sufficient time for the FMS to capture an updated leg. If, however, updates occur more frequently, this function will need to include a timer that is started after each update and expires after 3-5 s. During this time, updates are discarded.

In addition to that, the Traffic Pattern Follower passes TP.R, the distance to go until the start of the next segment, and TP.L, the distance to go until the next waypoint to the Energy Curve Manager and Traffic Pattern Trajectory Updater respectively. TP.R is found using library function ”calcDistToGo” by passing the current location, state.loc, converted to local coordinates and point Z, which is located at distance $t_d$ from the next fix, $F$, along the opposite of the current course, $\chi$ as shown in Figure A-22. The coordinates of Z can be found as follows:

\[
Z_{north} = F_{north} - t_d \cos(\chi) \quad (A.48)
\]
\[
Z_{east} = F_{east} - t_d \sin(\chi) \quad (A.49)
\]

TP.L is also found using function ”calcDistToGo” by passing the current location, state.loc, and the location of the next fix, $F$, both in local coordinates. Finally, the Traffic Pattern Follower specifies the current leg the aircraft is on through counter TP.currSeg. If no final intercept leg exists yet, since no updates on the base leg have been performed, there are only five traffic pattern trajectory segments, downwind leg, turn to base, base leg, turn to final and the final leg. The first update on the base leg inserts the final leg intercept leg that adds two new segments: turn to final intercept leg and the final intercept leg itself. The Traffic Pattern Follower increments TP.currSeg every time the aircraft passes the line perpendicular
to the point at which the turn to the next leg is initiated, i.e. at the point where the Traffic Pattern Follower passes the new leg, and at the point where the aircraft has completed the turn and intercepted the new leg. This can be detected by checking whether the aircraft has passed the line drawn perpendicular to the current course at a distance $t_d$ from the current fix along the current course. This perpendicular line is found using library function "getTurnEntryLine", whereas passing the line is checked by library function "lineCrossed". TP.currSeg is used by the Energy Error Detector to index the array of energy vs. range to go curve and boundaries in M, M_u and M_l. These arrays are automatically updated to include the final intercept leg by the Energy Curve Manager once this function receives an updated array of TP.leg[] from the Traffic Pattern Follower.

A.3.4 Flare Controller

Inputs

- $h$: altitude above touchdown zone

Outputs

- $v_{sink}$: target vertical speed to be passed to autopilot vertical speed hold controller

Trigger

started and stopped by higher level state machine, runs continuously while active

Description

Autoland performs a flare maneuver similar to the one implemented by autoland systems for commercial aircraft. It commands a smooth decrease in vertical speed from the start of the flare until touchdown, with the difference that the final target sink speed does not approach
zero, but a constant $v_{td}$. Autoland uses the following equation for the target sink speed, $v$ in terms of time elapsed, $t$:

$$v(t) = (v_0 - v_{td})e^{-\frac{t}{\tau}} + v_{td} \quad (A.50)$$

where $v_0$ represents the vertical speed at the start of the flare, $v_{td} = 3.3\, ft/s$ the target touchdown speed at the expected touchdown point, and $\tau = 1\, s$ the time constant at which the vertical speed command is decreased. With this choice, the flare height becomes 70 ft, which is the height above ground at which the higher level state machine switches to the "flare" state and invokes this function.

### A.4 Selected Library Functions

#### A.4.1 convertToGlobalCoords

**Inputs**

- **origin, O**: latitude and longitude of the origin of the local North-East coordinate frame
- **heading, $\chi$**: orientation of the local North-East coordinate frame with respect to true north of the global coordinate system
- **pos, P**: North and East coordinates of the point P in the local coordinate frame

**Outputs**

- **loc, $\mu$, $\nu$**: latitude and longitude of point P in global coordinate frame

**Trigger**

Function call
Description

This function first rotates the local coordinate system to the orientation described by course $\chi$ to convert $P$ to $P'$:

$$
\begin{bmatrix}
P_{north}' \\
P_{east}'
\end{bmatrix} =
\begin{bmatrix}
\cos(\chi) & -\sin(\chi) \\
\sin(\chi) & \cos(\chi)
\end{bmatrix}
\begin{bmatrix}
P_{north} \\
P_{east}
\end{bmatrix}
$$

Subsequently, this function calculates the change in latitude, $\Delta \mu$ and longitude, $\Delta \iota$ between point $P'$ and the origin of the local North-East coordinate system, at specific latitude $\mu_0$ using $\Delta N$ and $\Delta E$, which represent the difference in North and East coordinates of $P'$ and the origin.

$$
\Delta \mu = \arctan\left(\frac{1}{R_N}\right) \Delta N
$$

$$
\Delta \iota = \arctan\left(\frac{1}{R_M \cos(\mu_0)}\right) \Delta E
$$

where $R_N$ specifies the curvature of the earth in the prime vertical and $R_M$ the curvature of the earth in the prime meridian at latitude $\mu_0$, which can be found from approximation formulas or tabulated data. As an approximation, $R_N$ and $R_M$ may be set to the average radius of the earth $R = 6,378,137 m$. For a distance of 30 miles, as is assumed to be a typical distance to a runway, the error from neglecting the flattening of the earth would amount to 300 ft at 45° latitude, which is considered acceptable considering the included range margin of 30%. Finally, the location of $P$ in the global coordinate system is found by offsetting $\Delta \mu$ and $\Delta \iota$ by $\mu_0$ and $\iota_0$.

$$
\mu = \mu_0 + \Delta \mu
$$

$$
\iota = \iota_0 + \Delta \iota
$$

A.4.2 getTurnEntryLine

Inputs

$\texttt{turnDist, t}$: distance to waypoint at which turn should be initiated to avoid overshoot
\( P \): location of waypoint in local North-East coordinates

\( \chi \): course of the current leg

**Outputs**

\( I \): intercept point at distance \( t \) from waypoint \( P \) on the current leg

\( Q \): point at (arbitrarily chosen) distance 10 from intercept point \( I \) on line perpendicular to the current leg in the direction that yields a right-handed coordinate system if the x-axis points along the course of the current straight line segment

**Description**

Finds two points describing the turn entry line. The turn entry line is the line perpendicular to the current course at a distance \( t \) from the targeted waypoint as shown in Figure A-20. \( I \), the intercept point and \( Q \) are found as follows.

![Diagram showing turn entry line](image)

**Figure A-20: Calculation of turn entry line**
\[
I_{east} = P_{east} - t \cdot \sin(\chi)
\]
\[
I_{north} = P_{north} - t \cdot \cos(\chi)
\]

Since the slope of the line perpendicular to the current course is defined by \(m_2 = -\frac{1}{m_1}\) and \(m = \tan(\alpha)\)

\[
m_1 = \tan(90^\circ - \chi) = \frac{1}{\tan(\chi)}
\]
\[
m_2 = -\tan(\chi)
\]
\[
\alpha = \arctan(-\tan(\chi)) = -\chi
\]

In order to ensure that \(Q\) is always in the direction from \(I\) that forms a right-handed coordinate system, need to distinguish between \(\text{abs}(\chi) \leq \frac{\pi}{2}\) and \(\text{abs}(\chi) > \frac{\pi}{2}\).

For \(\text{abs}(\chi) \leq \frac{\pi}{2}\):

\[
Q_{east} = I_{east} + 10 \cdot \cos(\chi)
\]
\[
Q_{north} = I_{north} + 10 \cdot \sin(\chi)
\]

For \(\text{abs}(\text{crs}) > \frac{\pi}{2}\):

\[
Q_{east} = I_{east} - 10 \cdot \cos(\chi)
\]
\[
Q_{north} = I_{north} - 10 \cdot \sin(\chi)
\]
A.4.3 lineCrossed

Inputs

P: current location

I: intercept of turn entry line with current straight line segment as returned by function "getTurnEntryLine"

Q: second point on turn entry line returned by function "getTurnEntryLine"

Inputs

b: returns -1 if turn entry line has been crossed at position P, 1 if the line is still ahead, 0 if P is on the turn entry line

Description

This function uses the vector cross product of vector $V_1$ and $V_2$ to determine whether the line defined by points $I$ and $Q$ has been crossed. Figure A-21 illustrates the vector orientations. The vector cross product is formed in a way, so that the resulting vector has negative magnitude if $P$ is behind the turn entry line (i.e. the turn entry line has been crossed) and has positive magnitude if the turn entry line is still ahead.

$$V_1 = \overrightarrow{PI}$$

$$V_2 = \overrightarrow{IQ}$$

$$b = sign(V_2 \times V_1)$$
Aircraft position, \( P \) current leg

• \( P \): current aircraft position in local North-East coordinates

• \( Z \): location of the point to which the distance to go should be estimated in local North-East coordinates

• \( \chi \): course of the current leg

**Outputs**

**R**: distance to go from the current position \( P \) point \( Z \)

**Trigger**

Caller function (this is a library function to be used by several main modules)
Description

This function is used to calculate the “Range-to-go-until the next segment”, R, which is used by the Traffic Pattern Follower and the Energy Error Detector to time their switching to the next leg or next segment of the traffic pattern respectively. In this case, the higher level functions pass the current position, P and point Z as shown in Figure A-22. Furthermore, this function is also used to determine the distance to go from the current position to the next waypoint by the Traffic Pattern Follower and the Trajectory to Initial Point Follower. In this case, the higher level functions simply pass the next leg fix, F in place of Z. Generally, some small cross-track error c will always exist and hence the course connecting points P and Z will not equal the desired course \( \chi \). Since the autopilot will attempt to zero the cross-track error by returning the aircraft onto the line from point I to P prior to point Z, the range to go is not simply the distance denoted by \( s \). Rather, distance \( s \) denotes a lower bound on the range to go until the start of the next segment. Distance \( c + d \) on the other hand denote an upper bound to the range to go until the start of the next segment at point Z. In order to arrive at a reasonable estimate of the range to go given the lack of knowledge of the exact path that will be flown, distances \( s \) and \( c + d \) will be averaged. Note: It is necessary to include the cross-track error \( c \) in the distance to go estimation since typical cross-track error cause an elongation of the actual trajectory that causes energy errors on the order of the errors that the energy management function is trying to fix. If, for example, \( c = 60 \) ft and \( d = 1,500 \) ft, \( s \) differs by 4% from \( c + d \). Using only \( s \) instead of the average of \( s \) and \( c + d \) results in a difference of 2%, which is non-negligible considering that the energy manager attempts to keep the energy error below 5%. The estimate for Range to Go from point P until the start of the next segment at Z becomes:

\[
R = \frac{s + c + d}{2}
\]  
(A.55)

All distances are calculated using the euclidean distance between the end points of their connecting lines. Points P and Z are given as inputs. The intercept point, I can be calculated
using library function "calcInterceptPoint".

![Diagram of calculation of Range-To-Go until start of the next segment]

Figure A-22: Calculation of Range-To-Go until start of the next segment

### A.4.5 calcInterceptPoint

**Inputs**

**P**: current position of the aircraft in the local North-East coordinate frame

**P1, P2**: North and East coordinates of two points specifying the line to which the cross-track error is to be found; for trajectory legs, the two waypoints specifying the leg can be used

**Outputs**

**d**: absolute value of the cross-track error

**I**: North-East coordinates of the intercept point, I of the line perpendicular to the line defined by P1 and P2 through point P
Description

This function calculates the absolute value of the distance from point P to a line defined by P1 and P2. Figure A-23 illustrates the geometry of the problem. The line is extrapolated past the two points and an intercept point, I is found at a course perpendicular to the course of the line. It is irrelevant whether an intercept in between the two points given is possible. The cross-track error, d is the distance from point P to intercept I. A is a point on the line through p1 and p2 that has the same North coordinate as P. B is a point on the line through P1 and P2 that has the same East coordinate as P. m1, t1, m2, t2 designate slope and intercept for the original and intercept line respectively. d and point I are found as follows.

\[ d \text{ and point } I \text{ are found as follows.} \]

\[ \begin{align*}
\text{Line} & \\
P_1 & \\
P_2 & \\
\text{intercept, I} & \\
\phi & \\
h & \\
da & \\
v & \\
\text{aircraft position, P} & \\
\end{align*} \]

**Figure A-23:** Calculation of distance from line

203
\[ m_1 = \frac{P_{2\text{north}} - P_{1\text{north}}}{P_{2\text{east}} - P_{1\text{east}}} \]
\[ t_1 = P_{1\text{north}} - m_1 \cdot P_{1\text{east}} \]
\[ A_{\text{east}} = (P_{\text{north}} - t_1) \]
\[ h = P_{\text{east}} - A_{\text{east}} \]
\[ \phi = \arctan(m_1) \]
\[ d = h \cdot \sin(\phi) \]
\[ m_2 = -\frac{1}{m_1} \]
\[ t_2 = P_{\text{north}} - m_2 \cdot P_{\text{east}} \]

Since the intercept point is on both the original and the perpendicular line:

\[ I_{\text{east}} = \frac{t_2 - t_1}{m_1 + \frac{1}{m_1}} \]
\[ I_{\text{north}} = m_1 \cdot I_{\text{east}} + t_1 \]

A.4.6 calcTurnDist

Inputs

\( \chi_1 \): course of the current straight line segment to be flown in radians from \(-\pi\) to \(\pi\).

\( \chi_2 \): course of the next straight line segment to be flown in radians from \(-\pi\) to \(\pi\).

\( r \): turn radius to be assumed for the turn in m
Outputs

t: absolute value of the distance to the waypoint to start the turn at in m

Trigger

Function call

Description

This function calculates the distance to the waypoint to start the turn given the change in course to be achieved. The geometry used for the calculation is shown in Figure A-24. The calculation uses the fact that $\phi = 2\tau$. Applying the law of cosines to the two triangles described by $r$ and $s$ and $d$ and $s$ to find an expression for $s$ allows writing the following equality:

$$r^2(1 - \cos(\pi - \gamma)) = d^2(1 - \cos \gamma) \quad (A.56)$$

Figure A-24: Geometry used to determine the lead distance to the waypoint at which a turn should be initiated
Ignoring the negative solution (since both \( d \) and \( r \) are positive quantities), yields the following expression for the ratio of lead distance to turn \( d \) and turn radius \( r \):

$$\frac{d}{r} = \frac{\sqrt{1 - \cos \gamma}}{\sqrt{1 + \cos \gamma}}$$  \hspace{1cm} (A.57)

where \( \gamma = \pi - |(\chi_1 - \chi_2)| \).
Bibliography


208