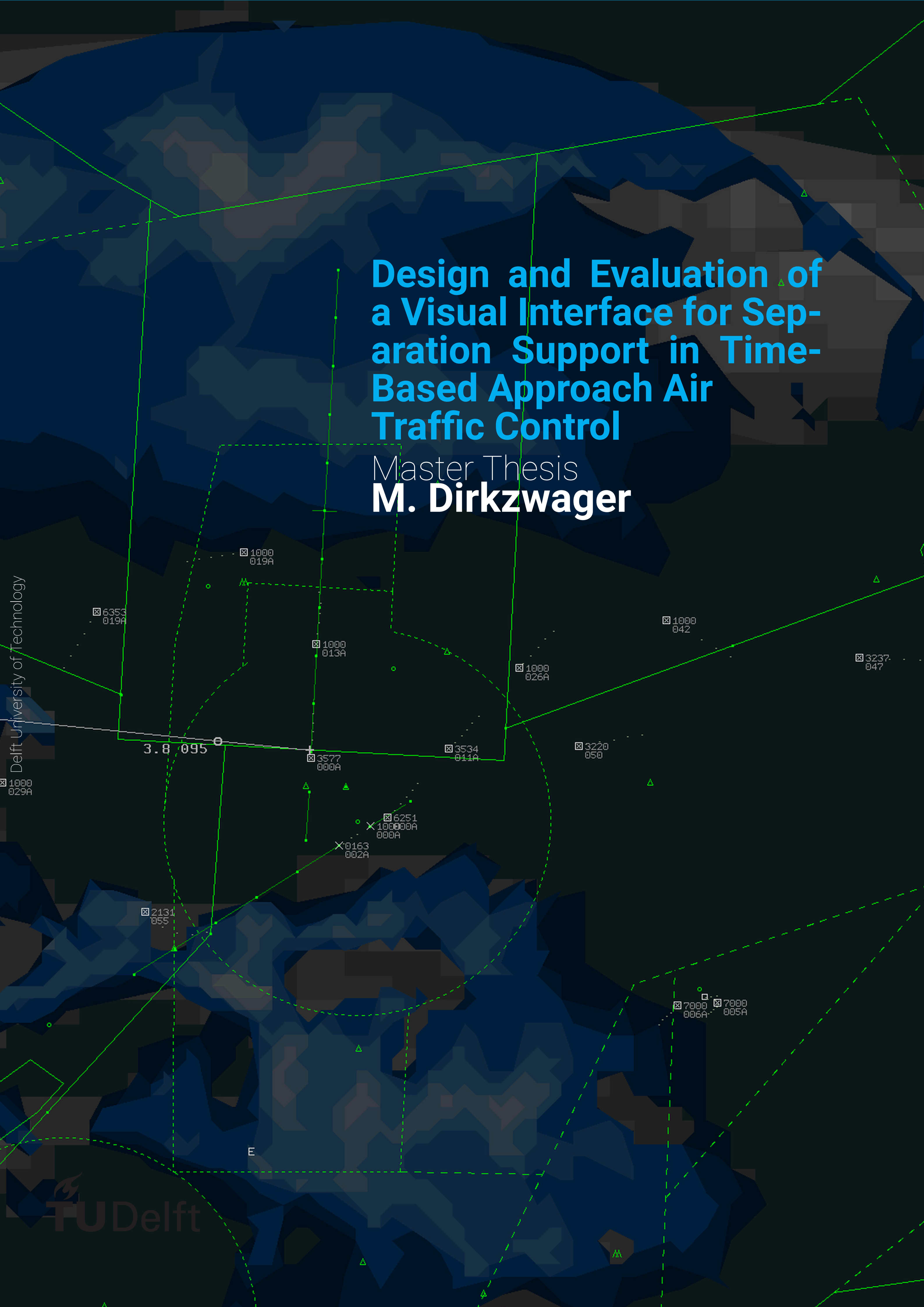

1000
019A
6353
019A
1000
013A
1000
026A
1000
042
3237
047
3.8 095
3577
000A
3534
011A
3220
050
1000
029A
6251
1000
000A
000A
0163
002A
2131
055
7000
006A
7000
005A
E

Design and Evaluation of a Visual Interface for Separation Support in Time-Based Approach Air Traffic Control

Master Thesis

M. Dirkzwager

Delft University of Technology

Design and Evaluation of a Visual Interface for Separation Support in Time-Based Approach Air Traffic Control

Master Thesis

by

M. Dirkzwager

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4206223	
Project duration:	May 23, 2018 – June 3, 2019	
Thesis committee:	Dr. Ir. C. (Clark) Borst,	TU Delft, First supervisor
	Prof. Dr. Ir. M. (Max) Mulder,	TU Delft, supervisor
	Dr. Ir. M. M. (René) van Paassen,	TU Delft, Supervisor
	F. (Ferdinand) Dijkstra,	LVNL, Supervisor
	Dr. Ir. O. O. (Otto) Bergsma,	TU Delft, External Committee Member

An electronic version of this thesis will be made available at `http://repository.tudelft.nl/`.

Preface

This report states the steps taken for a master thesis at the Delft Technical University, Faculty of Aerospace Engineering in the Control and Operations division. The thesis project was started on 23-05-2018 and will be finished on 03-06-2019.

I would like to take this moment to thank a few people who have contributed to this project. Firstly I would like to thank my supervisors from the TU Delft, Dr. Ir. Clark Borst, Prof. Dr. Ir. Max Mulder and Dr. Ir. René van Paassen, guiding me through this thesis project with all their useful insights and motivating enthusiasm. Next I would like to thank Ferdinand Dijkstra for providing the link between the TU Delft and the LVNL, and for providing me with all the practical knowledge I need concerning air traffic management. Finally, I would like to thank the LVNL for giving me the opportunity to perform a master graduation internship at the LVNL, enabling me to make the link between academics and practice.

Mats Dirkzwager
24-05-2019

Contents

I Thesis Paper **1**

II Literature Review and Preliminary Research *(Graded under AE4020)* **23**

III Part 3: Book of Appendices **111**

1 Experiment Manual **113**
- 1.1 Test Setup . . . 113
- 1.2 Approach Control in Sector X . . . 114
  - 1.2.1 Goal of the Simulation . . . 114
  - 1.2.2 Difference in Aircraft Symbol and Label . . . 114
  - 1.2.3 Difference in Simulator Control . . . 115
  - 1.2.4 Practice Scenario 1 . . . 116
  - 1.2.5 Approach Controller Guidelines . . . 117
  - 1.2.6 Distance-Based Separation . . . 118
  - 1.2.7 Practice Scenario 2 . . . 118
- 1.3 Time-Based Separation and the NATS Display . . . 119
  - 1.3.1 Time Based Separation . . . 119
  - 1.3.2 The NATS Display . . . 119
  - 1.3.3 Practice Scenario 3 & 4 . . . 120
- 1.4 The Ideal Turn-In Point Display . . . 120
  - 1.4.1 The Ideal TTB Point . . . 121
  - 1.4.2 The TTI Display . . . 126
- 1.5 Final Tips & Tricks for the ITIP Display . . . 126
  - 1.5.1 Practice Scenario 4 & 5 and Test Runs . . . 126

2 Additional Results **129**
- 2.1 Numerical Results . . . 129
- 2.2 Survey Results . . . 129
  - 2.2.1 Survey Questions . . . 129
  - 2.2.2 Results of Survey Rating Questions for Test Participants . . . 132
  - 2.2.3 Results of Survey Open Questions for Test Participants . . . 135
  - 2.2.4 Results of Survey Rating Questions for Professional Controllers . . . 140
  - 2.2.5 Results of Survey Open Questions for Professional Controllers . . . 140
  - 2.2.6 Discussion of Survey Results . . . 141

3 Code Layout **143**
- 3.1 Aircraft Class . . . 143
  - 3.1.1 ITIP Data Update Methods . . . 143
  - 3.1.2 updateTP(dt_sec) . . . 144
  - 3.1.3 updateNATSMarkerData() . . . 144
  - 3.1.4 ILS Related Methods . . . 144
  - 3.1.5 getLeg Methods . . . 145
  - 3.1.6 Path Integration Methods . . . 145
  - 3.1.7 Miscellaneous Methods . . . 146
- 3.2 ILS class . . . 147
  - 3.2.1 Landing Order Methods . . . 147
  - 3.2.2 Miscellaneous Methods . . . 148

3.3 GLILS class 148
3.4 GLITIP class. 148
3.5 Wind class 150

I

Thesis Paper

Design and Evaluation of a Visual Interface for Separation Support in Time-Based Approach Air Traffic Control

M. Dirkzwager. Supervisors: C. Borst, M. Mulder, M.M. van Paassen, F. Dijkstra

***Abstract*—On final approach, an Approach (APP) Air Traffic Controller (ATCo) is responsible for keeping sufficient separation between aircraft lining up on the ILS. The current industry standard is to separate these aircraft with a minimum distance, called Distance-Based Separation or DBS. European regulation requires all European airports to implement separation based on time (Time-Based Separation or TBS) before 2024. Due to this implementation, effectively changing the APP ATC task from a geometrical to a time-based problem, and because of further complications such as the European Re-categorisation of aircraft types, experts fear that the theoretical gains attainable by using TBS will not be fully realised. In this research, a display tool concept to aid APP ATCos in realising the full potential of TBS, the Ideal Turn-In Point (ITIP) display is designed and evaluated with respect to the current state of the art. The ITIP display assists controllers in selecting optimal approach strategies starting from the moment aircraft enter the Terminal Control Area. The display aims to assist the operator by showing the possibilities and restrictions in the system rather than giving (restricting) advisories. In an initial proof-of-concept experiment, comparing the ITIP display to the current industry state of the art display, promising results were found; the ITIP display was shown to maintain safety and increase efficiency, whilst maintaining controller workload. The current industry state of the art display is a tool designed by the National Air Traffic Services, currently operational at London Heathrow Airport.**

***Index Terms*—Time-Based Separation, Air Traffic Control, Display Tool, Approach**

NOMENCLATURE

4DME	4 NM Distance Measuring Equipment
ATC	Air Traffic Control
ATCo	Air Traffic Controller
APP	Approach
CDU	Command Display Unit
CV	Control Variable
DBS	Distance-Based Separation
DV	Dependent Variable
EID	Ecological Interface Design
FL	Flight Level
G/S	Glideslope, part of ILS
IAS	Indicated Airspeed
ICAO	International Civil Aviation Organisation
IFR	Instrument Flight Rules
ILS	Instrument Landing System
ITIP	Ideal Turn-In Point
IV	Independent Variable
kts	Knots
LOC	Localiser, part of ILS
LVNL	'Luchtverkeersleiding Nederland' (Dutch ATC)
MTOW	Maximum Take-Off Weight
NATS	National Air Traffic Services
NM	Nautical Miles
ORD	Optimised Runway Delivery
SRK	Skill-, Rule- and Knowledge based behavior
TBS	Time-Based Separation
TMA	Terminal control area
TTB	Turn To Base
TTI	Turn To ILS

I. INTRODUCTION

In the final phase of flight of an aircraft, an Approach (APP) Air Traffic Controller (ATCo) is responsible for guiding aircraft to the runway threshold. This is accomplished by vectoring the aircraft toward the Instrument Landing System, which consists of a Localizer (LOC) for lateral and Glideslope (G/S) for vertical guidance, together forming the Instrument Landing System (ILS). In this process, the ATCo controller must try to be as efficient as possible; a high landing rate means more capacity for the airport. However, the ATCo must also maintain safety; he/she must keep certain separation buffers between aircraft at all times. Next to a general separation buffer for aircraft which are close to each other, an extra separation buffer should be included when aircraft are flying close behind one another. The size of this last separation margin between an aircraft pair varies from 3 to 8 NM, depending on the type of aircraft. Its exact value per aircraft pair is specified by the International Civil Aviation Organisation (ICAO) [1]. The current industry standard is to provide these margins as pre-defined distances in nautical miles, called Distance-Based Separation (DBS). In these specifications, aircraft are divided into classes based on a.o. their Maximum Take-off Weight (MTOW). Currently, only four categories exist in these specifications. However, the newer European Wake Vortex Re-categorisation (RECAT-EU) project, will distinguish between as much as six categories, increasing complexity for the ATCo. While the implementation of RECAT is not mandatory, EUROCONTROL, an organisation dedicated to building a 'Single European Sky', encourages European airports to do so [2].

Another factor increasing the complexity of APP Air Traffic Control (ATC) in the future is the implementation of

Time-Based Separation (TBS). TBS is a solution to one of the flaws of DBS, which is that when large headwinds are present during final approach, the landing rate is significantly decreased when using DBS. This is due to the headwinds reducing the groundspeed of the approaching aircraft. When using TBS, the APP controller keeps a specified time between each aircraft, instead of a distance. This way, when large headwinds are present, the distance between aircraft may be reduced and the landing rate is thus maintained. As per European regulation, all European airports need to implement TBS before the 1^{st} of January 2014 [3]. The LVNL aims to implement RECAT-EU by this date as well.

In multiple analyses, the implementation of RECAT-EU and TBS have been proven to increase landing rates in large headwind conditions [4] [5]. However, experts fear that the practical gain in runway throughput will be less than the theoretical gain, since the implementation of TBS will change the APP task from a geometrical problem into a time-based one.

Separating based on time is less intuitive than separating based on distance. After all, on the map-like radar screen which the APP controller uses to monitor all aircraft states, distance can be directly observed whilst time is not a directly visible variable.

At London Heathrow Airport, TBS has been operational since 2015. In cooperation with the National Air Traffic Services (NATS) in the United Kingdom, they have developed a display tool to assist APP controllers in separating the aircraft efficiently on the ILS [6] [7]. While this tool provides a mark on the ILS where the ATCo can ‘aim’ at in order to safely separate aircraft, the tool does not provide any information on which strategy should be used to get the aircraft on that mark at the right time earlier on in the approach phase. Since the ATCo has limited control options available when aircraft are lined up with the ILS, experts fear that the theoretical gain attainable when using TBS might not be realised, as support is only available in the last, least flexible stage of the approach phase.

In order to aid APP controllers in their task, and to ensure as much gain as possible from the implementation of TBS is realised, demand arose for a display tool assisting APP controllers in selecting a flight path for an aircraft, such that the aircraft will be safely and efficiently separated on the ILS. This support tool should ideally provide support starting from the moment the APP controller is in control of the aircraft, i.e., when the aircraft enters the TMA.

The goal of this research is to develop a first concept of such a display. Since TBS has to be implemented by 2024 already, one of the major requirements for the display is that it can be implemented in the APP controller work environment at the Dutch ATC, the ’Luchtverkeersleiding Nederland’ (LVNL), with relatively few adaptations and that the display should be designed such that APP controllers at the LVNL will accept the display. In this research, such a display tool is proposed and evaluated. The display will be designed in a medium-fidelity ATC simulator developed at Delft University of Technology, called Sector X.

In this paper, the design process of the tool will be explained. Some detailed problem background will be provided and the display created by Heathrow and the NATS will be investigated. Subsequently, the design considerations for the display tool will be laid out, after which the results of an initial proof-of-concept experiment will be discussed. Finally, the results and implications for implementation will be discussed.

II. Problem Background

For this research, it is important to have a basic understanding of the APP task and the TBS concept. In this section, these topics are treated in more detail.

A. The APP controller Task

The APP controller’s task is to guide aircraft to the runway threshold in an efficient manner, whilst maintaining separation. Here, efficiency is measured by the number of aircraft that can land in a certain amount of time. The APP controller should always keep 3 NM distance between aircraft, when they are flying at the same altitude. If the aircraft have at least an altitude difference of 1,000 ft, the lateral separation may be smaller than 3 NM. An additional separation margin should be included when aircraft fly close behind one another, this separation is treated in Section II-B

The controller guides the aircraft to the runway by steering it towards the LOC and G/S of the ILS. These are predefined paths in the airspace preceding a runway which the aircraft can intercept using radio receivers. The LOC plane aligns the aircraft with the runway centerline, the G/S defines the perfect altitude path for an aircraft to follow (3 degrees from the touchdown point). The ILS is visualised on the APP controller radar screen by a green line with dots at every 2 NM. In Figure 1, a typical representation of an ILS mapping on an electronic radar display. An aircraft is pictured west of the ILS; in this paper all aircraft will be visualised with their heading angle vector. In all pictures and examples in this paper, the runway is located at the bottom of the ILS.

The APP controller can control aircraft in his/her sector by vectoring, i.e., giving speed, heading and altitude instructions to the pilot of the respective aircraft via radio communication. For vectoring of aircraft, the ATCo needs to adhere to some restrictions and guidelines. First of all, speed and altitude increasing commands should be avoided, since all aircraft need to decrease their potential and kinetic energy in an approach scenario. Furthermore, APP controllers must strive to let aircraft approach the G/S from below, since a G/S catch from above is more complex and results in more workload for the pilot. After the crash of the Turkish Airlines flight on the 25^{th} of February 2009 near Schiphol Airport, for which a catch-glideslope-from-above maneuver was partly responsible

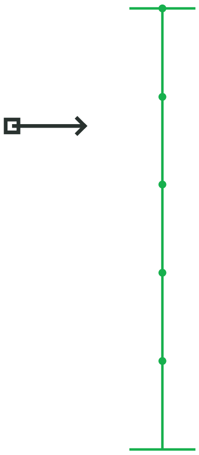


Fig. 1. The ILS with an aircraft, as typically visualised on an electronic radar display

[8], it is considered malpractice to let aircraft catch the G/S from above at the LVNL.

Next to these restrictions, the APP controller also has some guidelines, defining what an 'ideal' approach should be. The LVNL teaches these guidelines to their student controllers. In Figure 2, an approach trajectory for an aircraft approaching from the south is given. Four points of interest are depicted. For each of those points, certain ideal commands are defined.

The first decision point for an APP controller is the point where an aircraft starts its turn toward the ILS. This point is called the Turn To Base (TTB) point. The location of the TTB point and the heading angle the aircraft is given at that point greatly influences the approach path of the aircraft and thus its landing time. For an ideal approach path, this TTB point is located such that the aircraft will travel toward the ILS at approximately 10 NM from the runway threshold with a heading angle of 90° relative to the ILS is given, as depicted in Figure 2.

The second decision point the controller encounters is the Turn To ILS (TTI) point. At this point, the aircraft starts its turn into the ILS with a maximum heading angle of 30° relative to the ILS. In the example of Figure 2, the heading angle the aircraft would be given at the TTI point would be 150°. It is at this point that the ATCo clears the aircraft to intercept the ILS. The aircraft can now initiate its instrument landing procedure; ATCo heading and altitude vectoring is no longer needed. Ideally, the TTI point lies 2 NM from the ILS and a heading angle command of 30° relative to the ILS is given.

The third point of interest is the point at which the aircraft intercepts the ILS. Ideally, this point should lie about 8 NM from the runway threshold. From this point on, the aircraft heading and altitude is governed by the aircraft avionics interacting with the ILS.

The last point of interest lies at 4 NM Distance Measuring Equipment (4DME) from the runway threshold. From that moment, pilots are free to decelerate toward the landing speed as they see fit. Here, the APP controller does not control the aircraft anymore and transfers the aircraft communication to the tower of the airport.

For each part in the approach path of an aircraft, certain speeds and altitudes are seen as ideal. An altitude of 2,000 ft (FL20) is considered to be ideal for the entire path up until the ILS interception, where the altitude is decreased automatically by the aircraft interacting with the ILS. As such, the APP controller should strive to decrease the altitude of incoming aircraft toward FL20 as soon as possible. It is considered ideal for the aircraft to have an Indicated Airspeed (IAS) of 220 kts at the TTB point, 180 kts at the TTI point and 160 kts at the 4DME point.

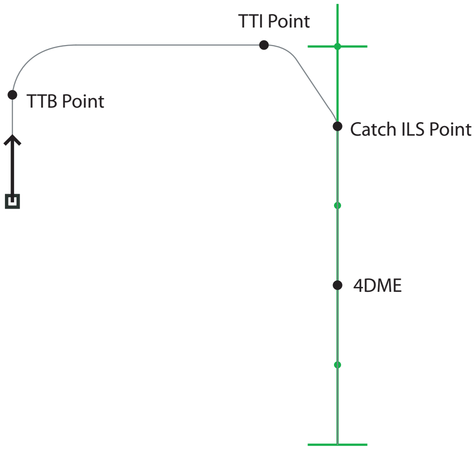


Fig. 2. An aircraft with its ideal approach trajectory, including four points of interest

B. Separation on Final Approach

When two aircraft fly closely behind one another, the follower aircraft should maintain sufficient separation with the leading aircraft to stay clear of wake vortices. The minimum size of this separation is specified in regulations by the International Civil Aviation Organisation (ICAO). The separation can be given in either a distance (Distance-Based Separation or DBS) or a time (Time-Based Separation or TBS). Currently, most airports use DBS. However, as per European Regulation, all European airports need to implement TBS before the 1^{st} of January 2024 [3].

When using DBS, the landing rate at a runway can be significantly reduced if strong headwinds are present, causing a reduction of aircraft groundspeed. If the distance kept between the aircraft is constant, strong headwinds will result in a reduced landing rate for that runway. The use of TBS solves this problem; when using a minimum time as separation buffer, aircraft can fly closer together in large headwinds. At Heathrow Airport, TBS is operational as of 2015. The British National Air Traffic Services (NATS) and intergovernmental

TABLE I
DISTANCE-BASED SEPARATION MINIMA FOR DIFFERENT WAKE PAIRS, IN NAUTICAL MILES [1], [10]

	Follower			
Leader	**Super**	**Heavy**	**Medium**	**Light**
Super	3	6	7	8
Heavy	3	4	5	6
Medium	3	3	3	5
Light	3	3	3	3

TABLE II
TIME-BASED SEPARATION MINIMA FOR DIFFERENT WAKE PAIRS, LOW HEADWIND CONDITIONS (5 KTS), IN SECONDS WITH 130 KTS AS AVERAGE LANDING IAS

	Follower			
Leader	**Super**	**Heavy**	**Medium**	**Light**
Super	86	173	202	230
Heavy	86	115	144	173
Medium	86	86	86	144
Light	86	86	86	86

European aviation organisation EUROCONTROL have proven that the use of TBS does not increase the chance of follower aircraft encountering wake turbulence from leading aircraft [9]. While the use of TBS, in combination with RECAT-EU, can result in a permanent landing rate increase of 5-10% [5], it will increase complexity for the APP task, since time, unlike distance, is not directly observable on the radar screen.

On final approach, separation between an aircraft pair will be at a minimum at the time the leading aircraft touches down. This is due to the compression effect, the phenomenon where the separation between an aircraft pair decreases due to the leading aircraft starting its deceleration to landing speed earlier. As such, the APP controller must make sure that the distance between the follower aircraft and its leading aircraft is larger than or equal to the prescribed minimum at the moment the leading aircraft lands.

To obtain the minimum separation distance in a TBS scenario, the DBS separation minimum is first converted to a time. This is accomplished by taking the DBS separation minimum for the aircraft pair from Table I and calculating the time it would take the follower aircraft to traverse this distance, using the groundspeed derived from an average approach IAS and an average headwind of 5 kts. The average approach IAS can be taken from data, in Table II the separation times are given for an aircraft with an average approach IAS of 130 kts. This time is then converted to a distance again by taking the actual average groundspeed of the follower aircraft, derived using the same average approach IAS but with the actual headwind at that time. This results in a smaller distance when the headwind present is larger than 5 kts.

III. STATE OF THE ART

Since the spring of 2015, TBS is operational at Heathrow Airport. For Heathrow Airport, a final approach display tool was developed to facilitate TBS operations by the National Air Traffic Services (NATS). This display tool shall henceforth be referred to as the NATS display. Not much is known about this tool. However, from a few short informational movies published by the NATS, it is possible to get a conceptual understanding of the display [6].

The NATS display uses two markers; the TBS marker and the Optimised Runway Delivery (ORD) marker. For an aircraft pair, the TBS marker is placed at the minimum separation distance behind the leading aircraft. The follower aircraft should never pass this marker, as this will count as a TBS conflict. The ORD marker is a marker behind the TBS marker, visualising the aforementioned compression effect.

An example of a TBS and ORD marker can be observed in Figure 3. In this figure, the TBS marker is the red dot and the ORD marker is the blue dot. The markers are given for an aircraft pair where aircraft A is the leading aircraft with aircraft B as its follower.

In Figure 3a, the aircraft travel behind one another with the same groundspeed, since they both fly at an IAS of 160 kts and at an altitude of 2,000 ft. Since the groundspeed of both aircraft is the same, the separation between the aircraft will remain the same. The aircraft are safely separated; aircraft B is behind the red TBS marker, created behind aircraft A. The blue marker lies behind the TBS marker; the distance between the ORD and the TBS marker is equal to the distance the follower aircraft will gain on the leading aircraft.

In Figure 3b, the leading aircraft has passed the 4DME point and thus started its deceleration. The follower aircraft has not yet passed the 4DME point and is still travelling with an IAS of 160 kts. The follower aircraft is thus catching up to the leading aircraft, as depicted by the decrease in distance between the TBS and ORD markers. Note that while the distance between the TBS and ORD marker has decreased, the distance between the follower aircraft and the ORD marker has remained constant.

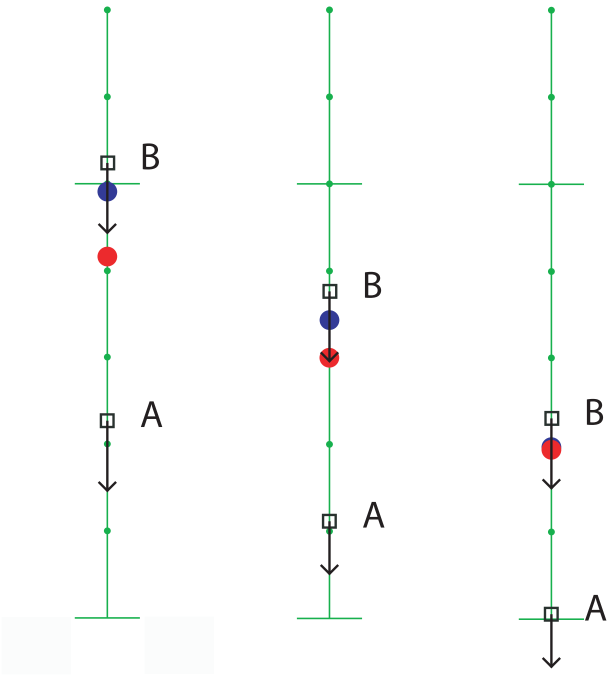
In Figure 3c, the final situation is depicted, just before the leading aircraft lands. As one can see, the ORD marker now lies almost exactly on top of the TBS marker, since the follower aircraft will not catch up to the leading aircraft before it lands anymore.

The idea behind the ORD marker is that if a follower aircraft is behind the ORD marker at any point on the ILS, no loss of separation will occur due to the compression effect. This gives ATCos a ’target’ to aim at when separating aircraft on the ILS. The ORD marker can be seen as a projection of the location of the TBS marker at the moment the leading aircraft lands, relative to the follower aircraft.

IV. PROPOSED NEW CONCEPT

A. *Design Requirements*

For the designed display, the main design drivers were as follows:



(a) (b) (c)

Fig. 3. An aircraft pair on the ILS with (a) both aircraft before the 4DME point, (b) the leading aircraft behind the 4DME point and (c) the leading aircraft moments before landing

1) The tool should help APP controllers in their task of separating aircraft based on time;
2) The tool should assist the controller in the early stages of approach control (from sector entry to ILS catch);
3) It should be possible to implement the tool in the current systems of the LVNL with minimum adaptations.

If the tool was to be implemented in the current systems of the LVNL, it should be implemented in the current work flow of the ATCos at the LVNL. The tool should be an addition to the current radar screen rather than changing its entire look and feel. For the APP controllers to accept and use the tool, it should avoid cluttering the radar display with too much information, as this wast the most often heard concern that APP controllers had about display tools. From literature, we know that the chances of an operator accepting a new display tool increase when that tool leaves the operator in control; the display tool must show the operator the options in the system rather than giving advisories, as operators are shown to resist automation that takes control away from them [11]. From these goals, the main design requirements were defined:

1) The tool should be designed such that display clutter is avoided;
2) The tool should show all control options to the controller rather than advising the controller on, or limiting the controller to, a specific approach, leaving the controller in command;

Because of user acceptance being of major importance for this display, and because the aim is to leave the APP controller in control rather than taking control away from the ATCo, the method of Ecological Interface Design (EID) was a large source of inspiration for the display tool.

Since this display is a first concept, it was chosen to include many functionalities in the design. Although this will most likely result in some display clutter, it enables researching which functionalities could be the most useful to an APP controller. After this initial research, the design should then be iterated upon and the functionalities that are most useful should be included in the final design.

B. The Ideal Turn-In Point Display

The path an aircraft takes is a significant variable in the time it takes for an aircraft to land. The major variables defining this path are the points where the aircraft turns toward the ILS (Turn To Base or TTB Point), the point where an aircraft turns into the ILS (Turn To ILS or TTI Point) and the target heading given at these points. Of lesser importance for the arrival time of an aircraft are speed and altitude, since only deceleration and descend commands should ideally be given.

A concept display was designed to assist approach controllers in early decision making, taking the future aircraft path as the variable of interest. The display visualises possible TTB and TTI points, along with possible turn headings, such that aircraft will be safely separated on the ILS. This display will be referred to as the Ideal Turn-In Point Display, or ITIP display. The display first provides support to the APP controller for the possible TTB points, after which support is given for the TTI point. The display will only be visible for an aircraft if that aircraft is selected by the user. The TBS marker as used by the NATS is also included.

All functions of the ITIP display will be treated in this section. In Section IV-H, the underlying algorithm will be explained. Please note that in all of the figures in this section, a large wind to the northeast is present, to illustrate its effect on aircraft trajectory.

C. Ideal Turn-In Points and Trajectory Prediction

In Figure 4, the ITIP display is shown for a selected aircraft (aircraft B) that has not yet turned to base. The main display feature is an array of green and red dots. Each dot represents a future location of the aircraft if its heading angle would remain unchanged. The display assumes the controller will follow the ideal speed and altitude guidelines as described in Section II-A. A dot is visualised for every 5 seconds in the future, which is equal to the radar update rate in the simulator and in most real-life ATC applications. Each dot in the display represents a Turn To Base possibility. The colour of the point is green when the algorithm predicts that the trajectory resulting from turning to base at this point is feasible. Here, feasible means that the aircraft catches the

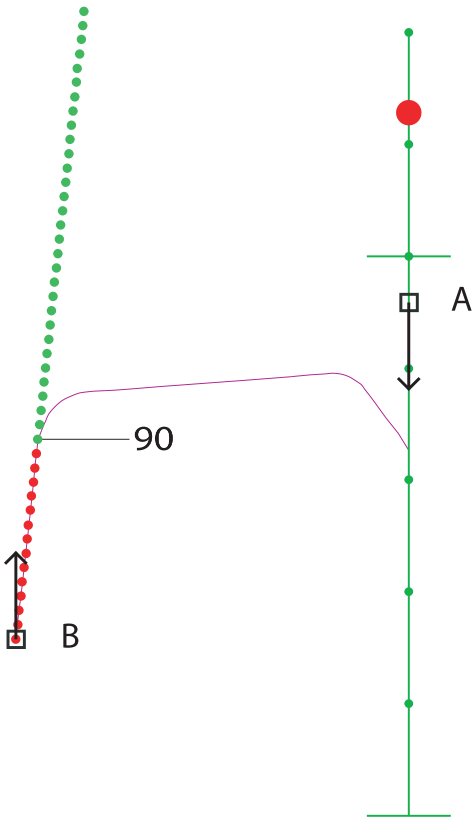


Fig. 4. The ITIP Display for a selected aircraft that has not yet received a TTB command

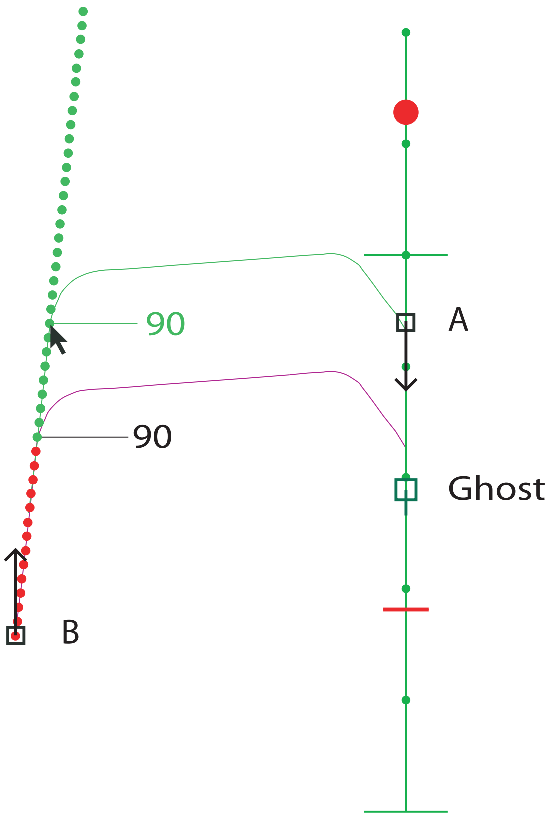


Fig. 5. The ITIP Display for a selected aircraft that has not yet received a TTB command, when hovering over a green TTB point

ILS from below and the aircraft will be separated on final approach from its leading and follower aircraft. A red TTB point indicates a conflict; either the resulting trajectory will result in loss of separation on the ILS or the aircraft has to catch the ILS from above.

The trajectory the aircraft is predicted to take is shown in purple. The algorithm assumes that the user will take the quickest feasible route, i.e., it assumes the user to pick a TTB point such that the total time travelled will be minimised, whilst making sure the trajectory is feasible. The user can update this prediction by clicking on one of the dots, thereby indicating to the display which strategy he/she is going to use. When clicking on a TTB point, the algorithm will update the prediction to a TTB at this point. Simultaneously, the data used for the plotting of the displays for all other aircraft will be updated with this new information. This way, the user can probe their plan to the algorithm so that the user may see if, and how, a certain path change will affect the solution space for other aircraft.

If the aircraft travels beyond the predicted turn-in point, the algorithm will assume the user made a mistake. The algorithm will assume the aircraft will turn at the next turn-in point, effectively ‘pushing’ the turn-in prediction prediction forward.

D. Trajectory Preview and Separation Ghost

In the simulator, the user can use the computer mouse to hover over all the dots of the TTB part of the display. Doing so will plot the predicted trajectory resulting from a TTB at this point, as can be seen in Figure 5. The trajectory will be green if it is possible for the aircraft to take this trajectory, that is, it does not cause a loss of separation on final approach or the aircraft catching the ILS from above. This is the case for the situation in Figure 5. If the trajectory would result in loss of separation or the aircraft catching the ILS from above, the trajectory will be plotted in red.

Next to plotting the predicted trajectory when hovering over a TTB point, the algorithm will also visualise a ghost aircraft on the ILS. The location of this ghost aircraft is equal to the location at which the selected aircraft is predicted to be at the time its leading aircraft lands. A red line is visualised on the ILS, representing the minimum separation the aircraft must have with respect to its leading aircraft. This way, the user can assess the resulting separation at the moment the leading aircraft lands, if the aircraft were to turn to base at the selected TTB Point. To maximise throughput, aircraft must be as close to the TBS marker of a leading aircraft as possible, at the moment the leading aircraft lands. Using the ghosting tool, one can investigate which solution is optimal, when deciding which aircraft to send to the ILS and with which path.

E. Heading Angle Preview and Manipulation

In addition to the predicted route, the heading angle command that should be given at the TTB point is shown. This serves as a reminder to the user. The user can vary this

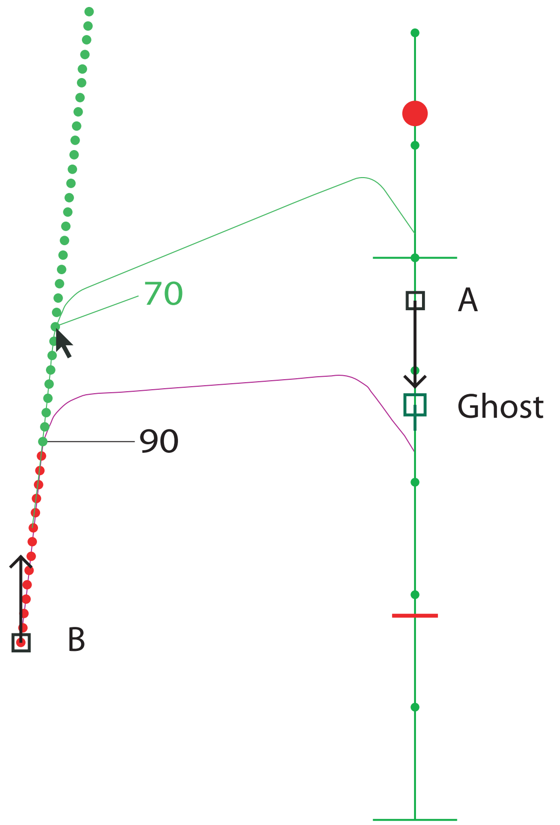


Fig. 6. The ITIP Display for a selected aircraft that has not yet received a TTB command, while hovering over a TTB point with the computer mouse and scrolling to adjust the TTB angle

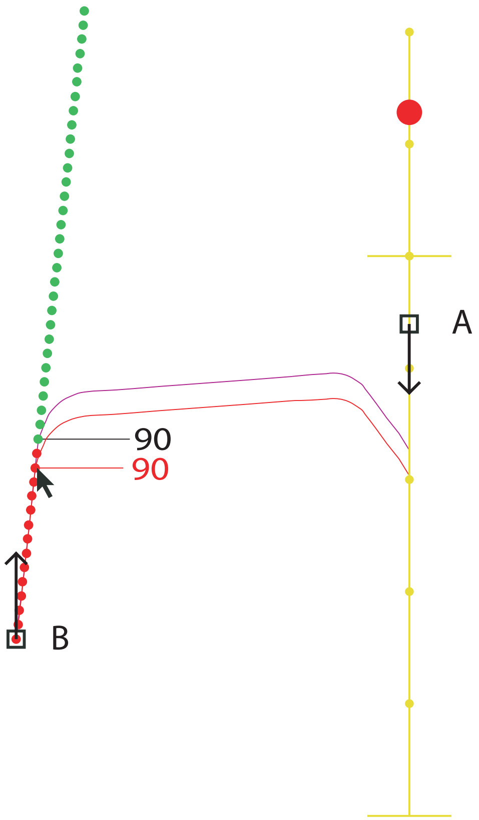


Fig. 7. The ITIP Display for a selected aircraft that has not yet received a TTB command, while hovering over an unfeasible TTB point due to the aircraft having to catch the ILS from above

heading angle by hovering over a turn-in point and using the scroll wheel to adjust the heading angle, as can be seen in Figure 6. Scrolling upwards will increase the heading angle (clockwise) and scrolling down will decrease the heading angle (counter-clockwise), up to some limits. When adjusting the heading angle, the algorithm will again assess separation at landing and path feasibility, changing the colour of the drawn path accordingly. Hence, if a TTB point is red, the user can search for a heading angle which would result in that point being green. The user can update the chosen angle by clicking. This will update the predicted heading angle for the TTB, as well as the colour of the dots in the display; this updated angle is now used as the heading angle for every TTB point. This allows the user to assess how a change in TTB angle would impact the solution space for the aircraft.

F. Highlighting of Cause of Trajectory Unfeasibility

If the plotted trajectory would result in a loss of separation on final approach, or if the trajectory is unfeasible because of the aircraft having to catch the ILS from above, the trajectory will be drawn in red. This situation can be seen in Figure 7. Here, the predicted trajectory resulting from a TTB at the selected point would result in the aircraft having to catch the ILS at less than 6.3 NM from the runway threshold, which would mean a ILS catch from above for an aircraft flying at an altitude of 2,000 ft. In the simulator, if an unfeasible trajectory is plotted when hovering over a point, the algorithm highlights the cause of the trajectory being unfeasible. Here, the ILS is highlighted. As one can see, the ghost aircraft is not drawn for this case, since separation is not the reason for the trajectory being drawn in red.

When separation is the reason for the trajectory being plotted in red, the display will highlight the aircraft with which a loss of separation is predicted to occur. This situation is visualised in Figure 8. In this figure, the TTB point for aircraft A at the computer mouse location would result in a loss of separation with leader aircraft C. This aircraft is highlighted by the disp. On the ILS, the resulting separation can be previewed; the ghost location lies before the TBS marker, indicating a loss of separation will occur at the moment aircraft C lands.

G. TTI Support in the ITIP Display

The TTI support of the display, shown in Figure 9, operates in the same way as the TTB part of the Display, the major difference being is that it is only visualised for aircraft flying in leg 3. The boundaries with which the user can change the turn-in angle are such that the angle with respect to the ILS can only be 30° or less, up to 5°. All other functions work exactly the same as for the TTB support stage of the display; the cause of trajectory infeasibility will be highlighted when plotting a trajectory in red, the separation result will be visualised using ghost

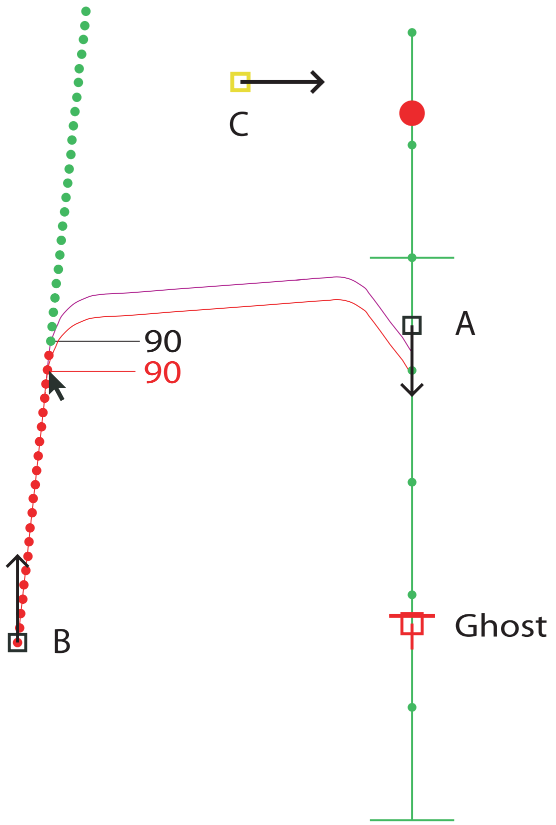


Fig. 8. The ITIP Display for a selected aircraft that has not yet received a TTB command, while hovering over an unfeasible TTB point due to a predicted loss of separation on the ILS with aircraft RA4743

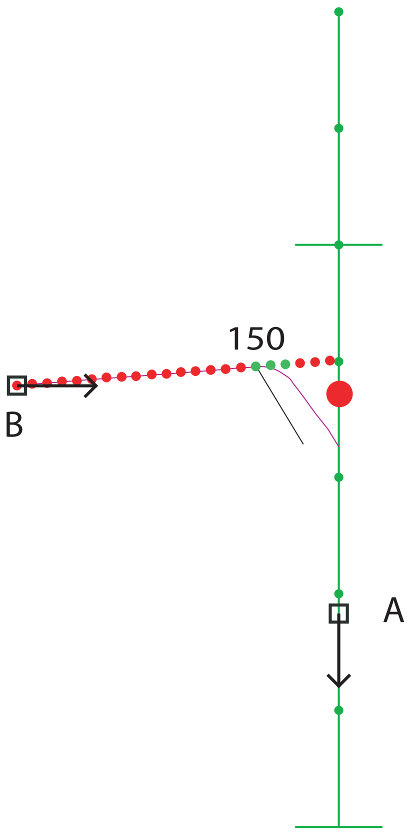


Fig. 9. The ITIP Display for a selected aircraft that is travelling toward the ILS

aircraft and the user can update the trajectory prediction by clicking TTI points and scrolling to adjust the turn-in angle.

Note that the TTI location impacts the final arrival time less than the TTB location. As such, the TTI support is used for minor efficiency improvements and small error corrections whereas the more influential TTB support is used for selecting the main aircraft path.

H. Display Algorithm

To calculate whether a TTB point is feasible, the display algorithm integrates the predicted path and compares the predicted landing times of the selected aircraft with all aircraft that have received a TTB command. Aircraft that have not yet received a TTB command are considered 'undecided'; taking these aircraft into account for the predictions would result in inaccurate visualisations since nothing is known about the controller's intentions with these aircraft. As soon as an aircraft receives a TTB command, the trajectory prediction is seen as more reliable.

When integrating the total path, the algorithm first decides which phase of approach the aircraft is in. Depending on the aircraft heading, location and controller commands, the algorithm predicts if an aircraft still needs to turn to base or if it has already turned to base. Similarly, the algorithm predicts whether the aircraft has already turned into the ILS or if it still needs to receive this command. After the phase of approach is determined, the algorithm integrates the path that the aircraft has yet to take.

The path is integrated by first deriving the groundspeed and flight path angle of the aircraft from its IAS, altitude and wind speed and direction. A constant heading angle is assumed for straight paths of flight except for when the aircraft has intercepted the ILS; here the heading angle is such that the aircraft heading angle counteracts drift caused by wind. For parts of the path where the aircraft is turning, the turn rate is derived using a constant bank angle and the aircraft groundspeed.

For the trajectory prediction, the algorithm assumes the aircraft will follow the APP controller guidelines as described in Section II-A. This means that the algorithm assumes an aircraft will slow down to an IAS of 220 when it has a higher IAS at that time and it has not yet received a TTB command. It will assume the aircraft will slow down to 180 kts as soon as the TTB command is given and a slowdown to an IAS of 160 kts is assumed when the aircraft is on the ILS. The assumption for altitude is that aircraft will descend to FL 20 as soon as possible. Deviation from these guidelines is possible and in some cases encouraged but the user should be aware of the implications which diverging from these reference guidelines may have.

V. Concept Evaluation and Sensitivity Analysis

A. Trajectory Prediction Sensitivity Analysis

A possible weakness of the use of the ITIP display could be its dependence on trajectory predictions. Discrepancies in the landing time prediction can be caused by varying pilot response times or by the ATCo reacting later than anticipated. Whilst the APP guidelines described in Section II-A are taught at the LVNL, every individual ATCo will have his/her own strategies and common practices. Both situations are basically the same; the aircraft executes a control action later than expected. To assess how this affects landing time predictions, a sensitivity analysis has been carried out.

To analyse this effect, the difference in landing time was plotted for certain control actions, if these actions were to be delayed by five seconds (a commonly used traffic radar update frequency). The the impact of delaying these actions was plotted for an array of wind headings, ranging from -45° to +45°, with 0° being a direction to the north, i.e., pure headwind at approach. Five wind velocities were used, ranging from five kts (standard headwind) to 25 kts (strong headwind). The six actions for which the effect of a five second delay on the landing time of an aircraft was assessed are:

1) Turning to base;
2) Decelerating from 250 kts to 220 kts when flying parallel to the ILS;
3) Descending from an altitude of FL50 to FL20 when flying parallel to the ILS;
4) Turning into the ILS;
5) Decelerating from 220 kts to 180 kts when flying toward the ILS;
6) Decelerating from 180 kts to 160 kts when lined up with the ILS.

These actions were selected because they are the standard decision points in an aircraft approach and are thus most likely to be delayed. For the first three actions, a medium type aircraft located at 8 NM to the west and 8 NM to the north of the runway threshold with a heading of 0° was taken. These aircraft had a standard speed and altitude of 220 kts and FL20. For the fourth action, a medium type aircraft located at 8 NM to the west and 10 NM to the north of the runway threshold with a heading of 90° and an altitude of FL20 was used and the fifth action made use of a medium type aircraft located at 2 NM to the west and 10 NM to the north of the runway threshold with a heading of 90°, an altitude of FL20 and an IAS of 180 kts. The sixth action was analysed for a medium type aircraft located on the ILS, 8 NM to the north of the runway threshold, at an altitude of FL20.

Note that the data presented in this section is taken from simulator experiments. Due to discretisation of the algorithm integrator functions, deviations of multiple times the used timestep can be present in the data. A second-order polynomial function has been fitted through the data to visualise the main effect.

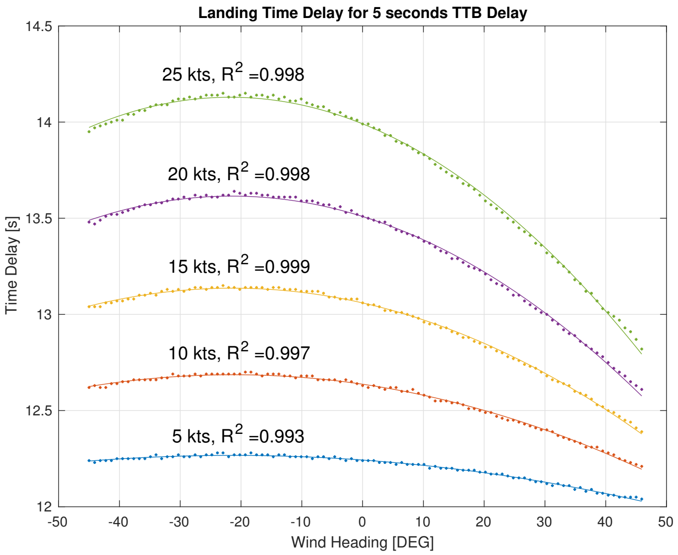


Fig. 10. Landing time delay caused by a five second TTB delay in seconds, for multiple wind settings

In Figure 10, the landing time delay for a five second TTB delay can be found. When looking at this figure, it is clear that the wind velocity is of major impact of this delay. The main parabolic effect of the time delay graph can be explained by the TTB delay lengthening the trajectory; a TTB delay increases the trajectory length by the aircraft travelling away from the runway but also by the aircraft drifting to the side by the wind. These effects cause an optimum, i.e., there exists a certain wind angle (± -20°) where the landing time delay due to five seconds TTB delay is at a maximum. Increasing wind velocities increases the landing time delay; for stronger winds a lengthened trajectory will result in a more lost time. Up to 14.2 seconds of delay can be present, almost tripling that of the TTB delay itself.

In Figure 11, the influence of decelerating from an IAS of 250 kts to 220 kts while the aircraft is flying parallel to the ILS on the aircraft landing time can be found. The delay is negative since a descent delay would result in the aircraft flying at a higher groundspeed for a longer period of time. The landing time deviation decreases for increased wind velocities and lightly varies with wind direction; at approximately -20° the time deviation seems not to be influenced by the wind velocity.

In Figure 12, the landing time delay for a five second descent delay for an aircraft flying parallel to the ILS can be found. It is evident that the impact of a descent delay in this situation is minimal; the landing time prediction can only be shortened by 0.23 to 0.18 seconds. The shape of the time delay is approximately the same as for the deceleration delay, the difference being the magnitude of the effect, amplifying the signal to noise ratio for these data.

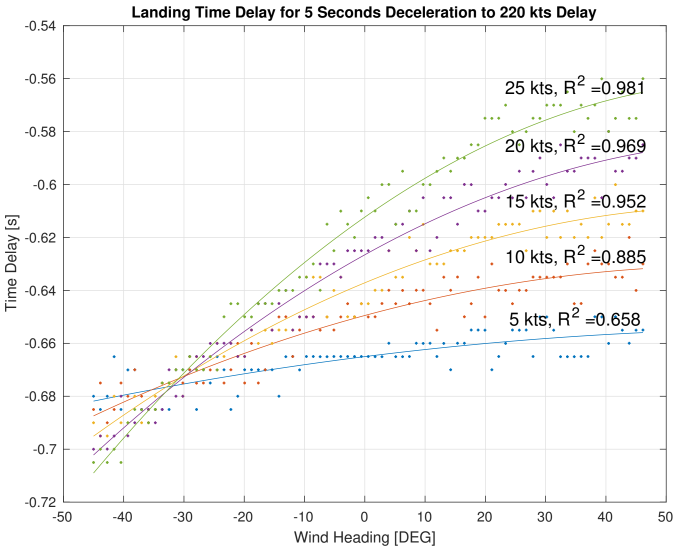


Fig. 11. Landing time delay caused by a five second deceleration from 250 to 220 kts delay,in seconds, for multiple wind settings

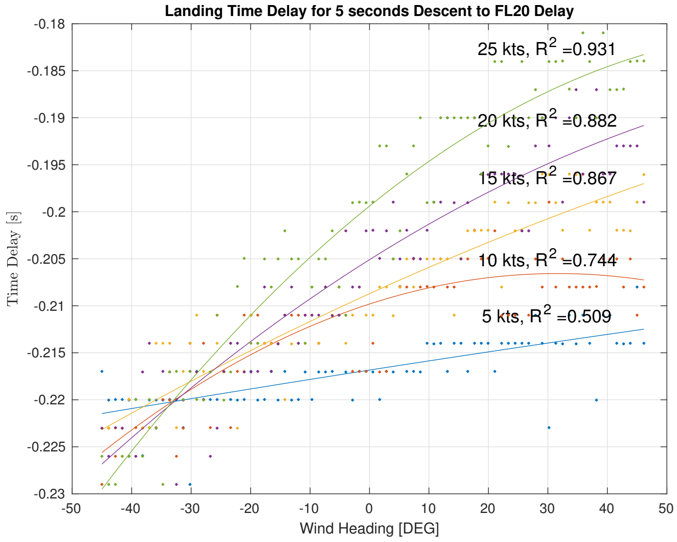


Fig. 12. Landing time delay caused by a five second descent from FL50 to FL20 delay,in seconds, for multiple wind settings

In Figure 13, the time deviations for a 5 second TTI delay can be seen. The time deviations follow largely the same pattern as the TTB delay but with a smaller magnitude. The main parabolic effect can be explained by the difference in path length the TTI delay causes; since the majority of the path after the TTI consists of the aircraft travelling in opposite direction to the wind, a wind with a heading of 0° will result in the largest delay for an increased path length.

In Figure 14, the influence of a five second deceleration from 220 to 180 kts on the landing time is visualised. As could be expected, the effect bears resemblance to the deceleration delay for an aircraft flying parallel to the ILS. The time delay increases for increased wind heading the effect of wind heading angle is more pronounced for higher wind velocities.

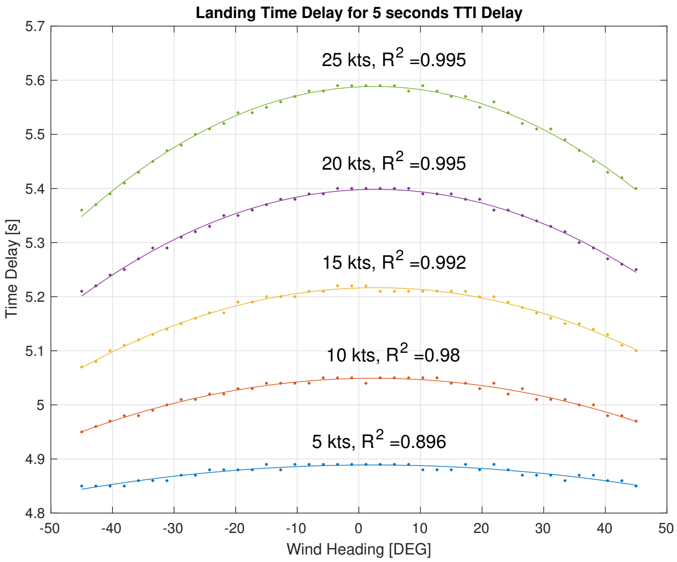


Fig. 13. Landing time delay caused by a five second deceleration to 180 kts delay when on travelling toward the ILS, in seconds, for multiple wind settings

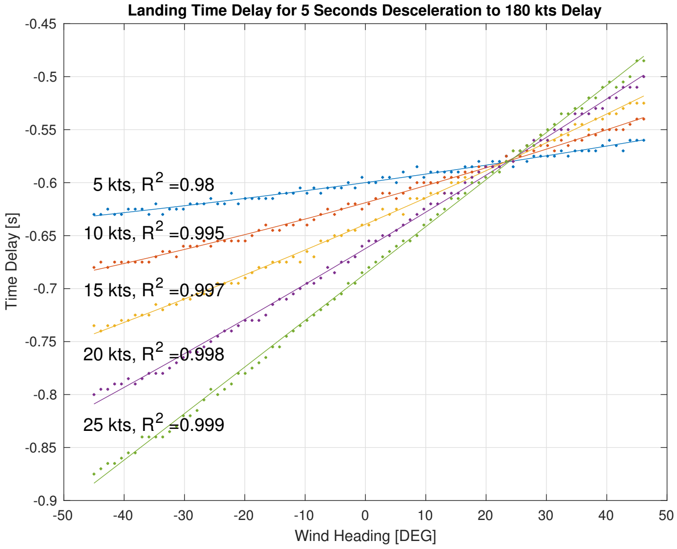


Fig. 14. Landing time delay caused by a five second deceleration to 180 kts delay when on travelling toward the ILS, in seconds, for multiple wind settings

In Figure 15, the landing time delay for when the aircraft decelerates from 180 kts to 160 kts 5 seconds later than expected is given. As was expected, the time delay is symmetrical around 0°. What is interesting to note is that for larger headwinds, aircraft arrive faster when decelerating five seconds later.

What is interesting to note is that in larger headwind conditions deviations from the expected action times will have a higher impact for the actions that are most influential, i.e., the TTB, TTI and deceleration to 160 on the ILS actions. This result is especially interesting since this tool is developed for TBS, a concept where the most efficiency

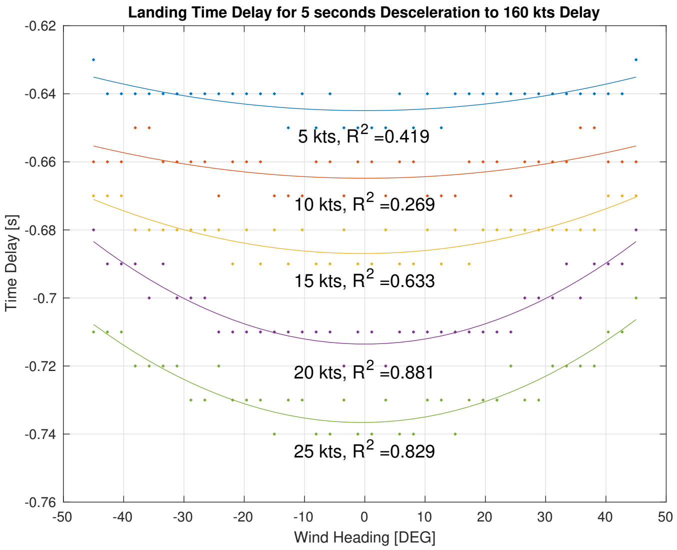


Fig. 15. Landing time delay caused by a five second deceleration to 160 kts delay when on the ILS, in seconds, for multiple wind settings

gains are accomplished in strong headwind conditions. The display is thus more susceptible to pilot or ATCo response time deviations when strong headwinds are present. Next to this being a possible weakness of the display algorithm, this finding underscores the importance of precise action timing and support thereof, especially in strong headwind conditions.

B. Professional Opinions

The concept has been proposed to two APP controllers at the LVNL. In these test sessions, the controllers used the ITIP display and the NATS display in a simulated environment and provided feedback on their experience. The opinions of the controllers were diverse; one of the controllers was very positive, noting that he found the ITIP display to be a *'Great experimental tool which could be a great benefit in real life to support or help controllers optimise the final approach'*. The controller noted that he strongly believed that with further research the ITIP display should be (partly) implemented in future APP control. He found that the ITIP display was even more useful than the NATS display and that he experienced no increased display clutter when using it. This ATCo used the display mainly for confirmation of his chosen flight path strategy. As an extra feature, he noted that speed advisories might be added for a future display iteration.

The other APP controller stated his interest in the TTI support of the display. However, this more conservative ATCo noted that *'At the moment [he does not] need additional support, just the TBS indicator'*. He found that *' More support leads to a more passive controller and might delay corrective actions if needed'*.

This is in line with the expectations. The more progressive ATCos are positive towards new display tools and are open to more, and newer forms of, support. More conservative controllers tend to rely on their current tools and are more suspicious of large adaptations to their trusted work environment. To fully assess the ITIP display, performance of controllers using the ITIP display must be compared top the performance of those controllers using only the NATS display, which will be done in Sections VI and VII.

VI. EXPERIMENT

The goal of the display was to assist the APP controller in attaining optimum time-based separation on final approach, whilst maintaining safety. To investigate the performance of the designed display tool, an initial proof-of-concept experiment was conducted. In this experiment, participants played out APP scenarios in real time, both with the NATS display and with the ITIP display.

A. Participants, Instructions and Procedure

For the experiment, a within-subjects design was chosen i.e. all participants perform all experiment scenarios. An easy and a hard scenario were designed. All participants played through the two easy and the two hard scenarios, both with the NATS and with the ITIP display, totalling 4 measurement runs per participant. Before the measurement, each participant received 75 minutes of training; 15 minutes without a display, 30 minutes with the NATS display and 30 minutes with the ITIP display. The measurement runs were balanced such that each possible order of measurement runs was given to one participant, and took 15 minutes each.

All participants were semi-professionals; people with knowledge about APP control and/or simulator training in general ATC. Eight participants in total were tested. The participants were told to optimise efficiency in the scenarios; they had to maximise throughput by landing as much aircraft as possible, whilst keeping separation.

B. Independent Variables

Two independent variables (IVs) were defined for this experiment:

1) The display type, two levels: NATS and ITIP;
2) Scenario difficulty, two levels: Easy and Hard.

Here, both IVs had two levels. The display types used were the NATS and the ITIP display. The NATS display was recreated in the simulation environment. The scenario difficulty featured an easy and a hard scenario.

The reason for testing two difficulties was the possibility that participants would play the scenarios perfectly, not showing any performance deviation between the NATS and ITIP displays. For a difficult scenario, the deviations were expected to be more pronounced

In the easy and hard scenarios the number of aircraft arriving in the TMA was the same, but the direction in which the aircraft arrived differed. The easy scenario featured traffic

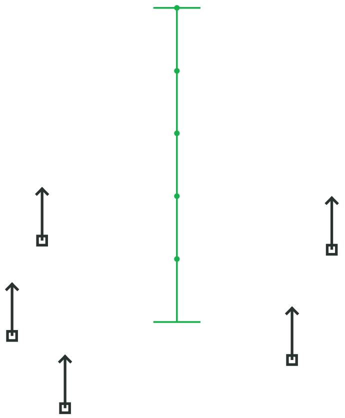


Fig. 16. A possible realisation of an easy scenario

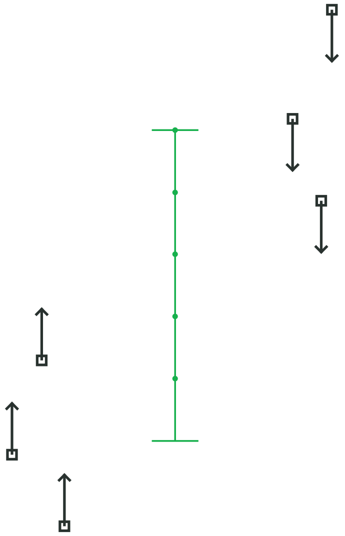


Fig. 17. A possible realisation of a hard scenario

arriving from the south, at both the east and west side of the ILS, as can be seen in Figure 16. For the hard scenario, traffic arrived from the south at one side of the ILS, and from the north on the other side of the ILS, as visualised in Figure 17. All aircraft arrived with speeds ranging from 220 to 250 kts, at an altitude of FL20, for simplicity. For both the easy and the hard scenario, two versions were made (V1 and V2) in which small deviations were present, as to prevent scenario recognition by the participants. The order in which participants received these different versions and the combination with displays with which they received them was balanced. The balanced within-subjects design matrix can be found in Table III. The reason for changing the direction of arriving traffic to vary scenario difficulty, as opposed to increasing the number of aircraft, was to avoid measuring the obvious increase in efficiency when more aircraft were present in a scenario.

In the scenarios, a wind of 25 kts was present. For the easy scenarios, this wind had a heading angle of 10 degrees, for the hard scenarios a heading angle of 15 degrees was used. The wind was varied to further distinguish between the scenarios, not to increase or decrease scenario difficulty.

TABLE III

Within-subjects design experiment matrix, E = Easy, H = Hard

P	Test 1	Test 2	Test 3	Test 4
1	E , NATS, V1	H, NATS, V1	E, ITIP, V2	H, ITIP, V2
2	E, ITIP, V2	H, ITIP, V2	E, NATS, V1	H, NATS, V1
3	H, NATS, V1	E, NATS, V1	H, ITIP, V2	E, ITIP, V2
4	H, ITIP, V2	E, ITIP, V2	H, NATS, V1	E, NATS, V1
5	E, NATS, V2	H, NATS, V2	E, ITIP, V1	H, ITIP, V1
6	E, ITIP, V1	H, ITIP, V1	E, NATS, V2	H, NATS, V2
7	H, NATS, V2	E, NATS, V2	H, ITIP, V1	E, ITIP, V1
8	H, ITIP, V1	E, ITIP, V1	H, NATS, V2	E, NATS, V2

C. Control Variables

For the experiment, several control variables (CVs) were identified:

1) The sector geometry with the runway and ILS location;
2) The number of aircraft in each scenario;
3) The update rate of the traffic (each 5 seconds);
4) Aircraft performance;
5) Aircraft altitude at TMA entry;
6) Type of aircraft mix.

The sector used was a square of 72 by 72 NM, corresponding to the radar screen used at the LVNL, which is a circle with a radius of 36 NM. The runway was located in the middle and the ILS was directed due north. In each scenario, 11 aircraft would enter the sector. Traffic was updated every 5 seconds, as is the standard radar update frequency at the LVNL. In the scenarios, only the types heavy and medium were used, to increase scenario simplicity. All aircraft types had the same performance and the traffic mix corresponded to that at Amsterdam Schiphol Airport; 10% of the aircraft was of the type heavy and 90% of type medium.

D. Dependent Measures

From the experiment, the variables of interest were efficiency and safety. For determining safety, the following Dependent Measures (DMs) were measured during the runs:

1) Number of TBS conflicts;
2) Maximum conflict intrusion distance;
3) Average conflict duration;
4) Average distance from the runway at conflict initiation.

Here, one TBS conflict is counted when a follower aircraft intrudes on the minimum separation behind a leading aircraft on the ILS, irrespective of the conflict duration. For every conflict, the maximum distance that the follower aircraft intrudes on its minimum separation is also stored and divided by the number of total TBS conflicts to get the average maximum intrusion. The duration of the conflict in seconds is also logged, as well as the distance from the runway at the time the conflict is initiated. An increase in the amount of conflicts, average maximum conflict intrusion, an increase in average conflict duration and a increase in average distance from the runway at conflict initiation would signify decreased safety in the system.

For determining efficiency, the following DVs were measured:

1) Average time separation buffer per landed aircraft;
2) Landing rate;
3) Number of non-treated aircraft at the end of the scenario;
4) Distance from runway to ILS catch point;
5) Controller workload.

The separation buffer is the spare amount of separation in seconds between an aircraft and its leader, at the moment the leading aircraft lands. A lower separation buffer would indicate a more efficient solution. The landing rate is equal to the number of aircraft landed per minute, starting from the time at which the first aircraft lands. A higher landing rate would indicate higher efficiency. The number of non-treated aircraft is the number of aircraft that have not yet received a TTB command at the end of the scenario and are thus untreated by the user. More non-treated aircraft would indicate lower controller capacity and thus a lower efficiency. The point at which the aircraft catches the ILS is also logged. The theory is that if aircraft catch the ILS at a larger distance from the runway, capacity is decreased since there is less room on the ILS. The controller workload is the workload as perceived by the controller, where a lower workload would indicate more efficiency. Here, all data were taken from the recorded data in the test runs, except for the controller workload. For the workload, the controllers were given a subjective rating scale on paper after each run in which they could mark the perceived workload on a RSME (Rating Scale Mental Effort) for that specific scenario. Distinction was made between three kinds of workload: overall workload, physical workload and mental workload.

Next to these numerical data, the participants will be asked to complete a survey with questions about both displays, allowing them to submit their opinion and input.

E. Hypotheses

The first hypothesis is that the use of the ITIP display will maintain safety with respect to the NATS display. For the DVs, this means that the use of the ITIP display will result in (1) the same amount of TBS conflicts, (2) the same average maximum conflict intrusion, (3) the same conflict duration and (4) the same average distance from the runway at conflict initiation.

The second hypothesis is that the use of the ITIP display will increase efficiency with respect to the NATS display. More specifically, the use of the ITIP display will result in (1) a smaller separation buffer, (2) a higher landing rate, (3) less non-treated aircraft at the end of the scenario, (4) an average ILS catch point closer to the runway and (5) a lower workload, with respect to the use of the NATS display.

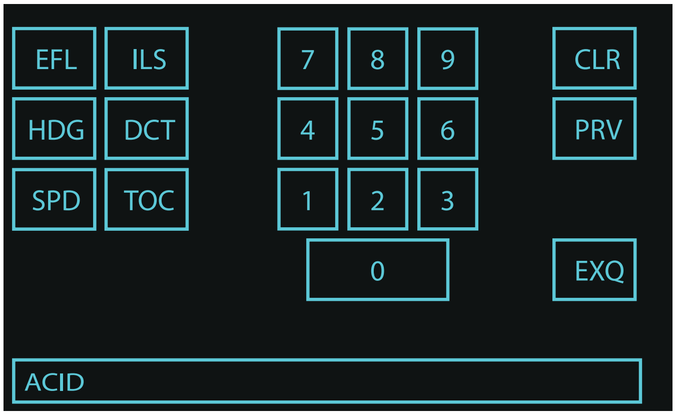


Fig. 18. The command interface as visualised in the simulator

F. Apparatus

To test the displays, a medium-fidelity Java-based ATC simulator created at Delft University of Technology called Sector X was adapted for APP ATC. In the simulator, the user can control aircraft much like in reality. Instead of voice commands, a command interface was used to communicate clearances to aircraft, see Figure 18. In this figure, one can see that the user can give all the basic commands to an aircraft that an APP controller would give in a real-life situation: the user can give heading, altitude and speed commands, as well as being able to clear the aircraft to catch the ILS using the 'ILS' button.

In the simulator, certain assumptions and modelling choices have been made. All aircraft respond immediately to the commands they are given i.e. reaction times of the virtual pilots were instantaneous. Aircraft were modelled to turn with their maximum bank angle of 30 degrees. Aircraft accelerated and decelerated with 2 kts/s^2. The rate of climb varied from 470 ft/min for a light aircraft to 3,160 ft/min for a heavy aircraft and the rate of descent varied from 1,345 ft/min for a heavy aircraft to 2,580 ft/min for a light aircraft. The International Standard Atmosphere model was used to calculate groundspeed from IAS and altitude.

In the simulator, only aircraft of type light, medium and heavy were present. For future research the RECAT-EU categories can be implemented with relative ease, however this was outside the scope of this research as the influence of the ITIP display could be evaluated sufficiently for this initial proof-of-concept experiment.

VII. RESULTS

In this section, the results obtained from the experiment are presented. For dependent variables, the distribution of the results did not differ significantly from a normal distribution. However, because of the small sample size of the experiment, and because the assumption of a normal distribution is inherently flawed for some of the IVs as they have theoretical minima/maxima, the statistical significance of the results was assessed using the non-parametric, relatively conservative

Wilcoxon Signed-Ranks test instead of the more widely used repeated measures Analysis of Variance (ANOVA). For all tests, a significance level of 0.05 was used. In the testing, statistical significance was assessed for the different display types per scenario difficulty, i.e., no statistical significance between scenario difficulty was assessed. This is because the effect of difficulty on the DMs was not the focus of this research.

A. *Safety*

In Figures 19, 20, 21 and 22, bar charts show the data for the number of TBS conflicts, average maximum conflict intrusion, conflict duration and average distance from the runway at conflict initiation, respectively. In the bar charts, the individual contribution per participant is given. For bar charts listing averages, the fraction of the total average contributed by that participant is given.

A trend can be seen where the number of conflicts increases when using the ITIP display in the easy scenarios but decreases when using the ITIP display in the hard scenarios. The average maximum intrusion decreases in the ITIP scenarios and the average conflict duration is also decreased. For the average distance to the runway at conflict initiation, a trend shows a decrease in distance for the hard scenarios while an increase in distance is observable for the easy scenarios.

Looking at these data, two observations can be made. Firstly, when looking at the amount of TBS conflicts in Figure 19, one can see that the use of the ITIP display increases the number of conflicts for the easy scenarios whilst decreasing the number of conflicts for the hard scenarios. This is in line with observations made during the experiment; using the ITIP display participants were more likely to pick more high-risk strategies, since they were confident these strategies were safe. Since the participants had more free time during the easy scenarios, they were more likely to investigate high-risk scenarios, possibly explaining the difference in trend for the two scenario difficulties. Oftentimes, these high-risk strategies resulted in a small conflict close to the runway as participants gave the 160 kts command too late on the ILS, resulting in a small intrusion caused by the compression effect.

Controllers were not able to solve these conflicts since they were often initiated close to the runway, as can be seen in Figure 22. Oftentimes the leading aircraft had already passed the 4DME point, reducing the control options to mitigate an impending conflict. Reducing the follower aircraft speed would either not impact the separation enough or cause conflicts behind that aircraft. That these conflicts were small and less severe than the conflicts for the NATS scenarios is reflected in both the average maximum intrusion, visualised in Figure 20 and the average conflict duration, visualised in Figure 21. In real-life ATC, small infringements of less than a few tenths of NM can occur, underscoring the moderate severity of the conflicts in the ITIP scenarios.

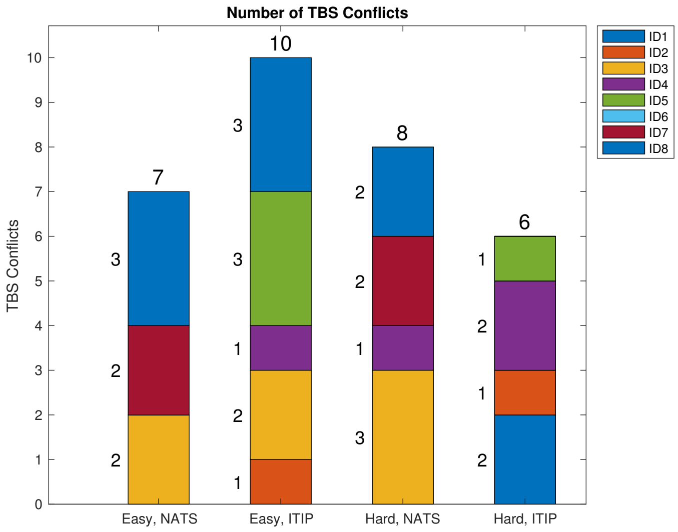


Fig. 19. The number of TBS conflicts, per scenario

Secondly, it seems as if some participants that chose a safe strategy in the NATS display scenarios were tempted to pick more high-risk strategies when using the ITIP display, whereas participants that are more risk-taking in the NATS display scenarios tended to select safer strategies when using the ITIP display. This can be seen by looking at Participants 3, 7 and 8, who each have a large number of lengthy TBS conflicts with significant maximum intrusions for the NATS scenarios which are reduced in amount and severity when using the ITIP display, whereas Participants 2, 4 and 5 seem to cause more, prolonged conflicts with larger intrusions when using the ITIP display. This corresponds to the responses of the participants, some of whom noted that the ITIP display encouraged them to take more risky decisions whilst others noted the display showed them their preferred strategy was unsafe.

Whilst any conflict is one too many in real-life ATC, the number of conflicts and their maximum intrusions must be seen relative to the other scenarios, as non-professional participants were used. These participants are likely to cause a conflicts in this relatively difficult ATC task. It is expected that professional ATCos will not cause any separation conflicts, as they are known to always implement some safety buffer; they will most likely not choose risky boundary solutions [12]. The conflicts that were encountered for the ITIP display scenarios could also be mitigated by design, i.e., using a small buffer in the algorithm. To conclude that the use of the ITIP display increases safety at this point would be premature, however it can be concluded that, in the current setup, use of the ITIP display does not decrease safety.

B. *Efficiency*

In Figure 23, the average separation buffer for each scenario can be observed. A trend where the ITIP display results in a reduction of the average separation buffer is visible. A Wilcoxon-Signed-Ranks test was carried out to

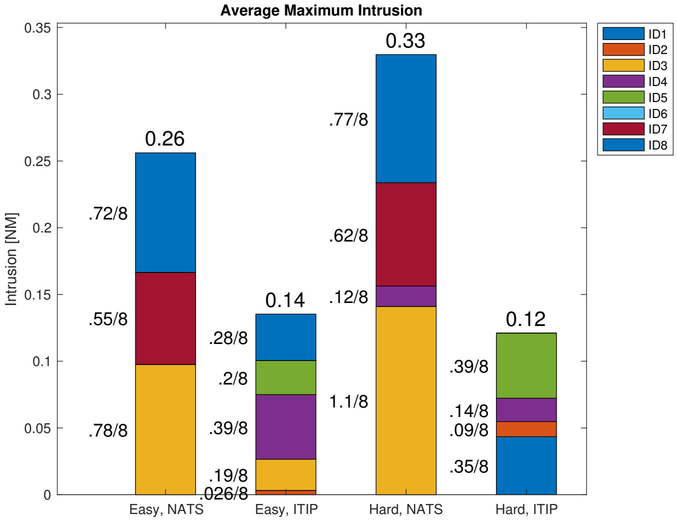


Fig. 20. The average intrusion per TBS conflict in nautical miles, per scenario

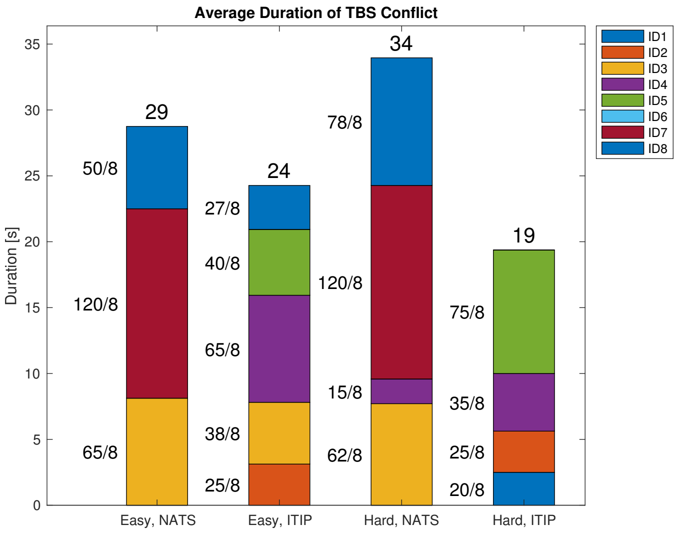


Fig. 21. The average duration of a TBS conflict in seconds, per scenario

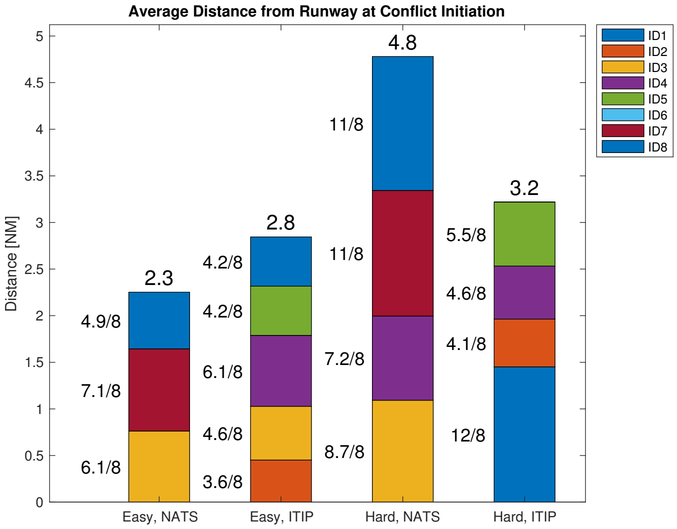


Fig. 22. The average distance from the runway at a TBS conflict initiation in nautical miles, per scenario

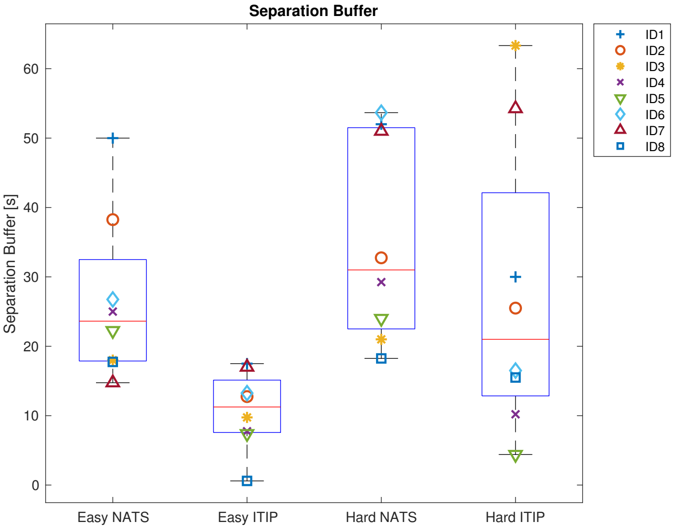


Fig. 23. The average separation buffer in seconds, per scenario

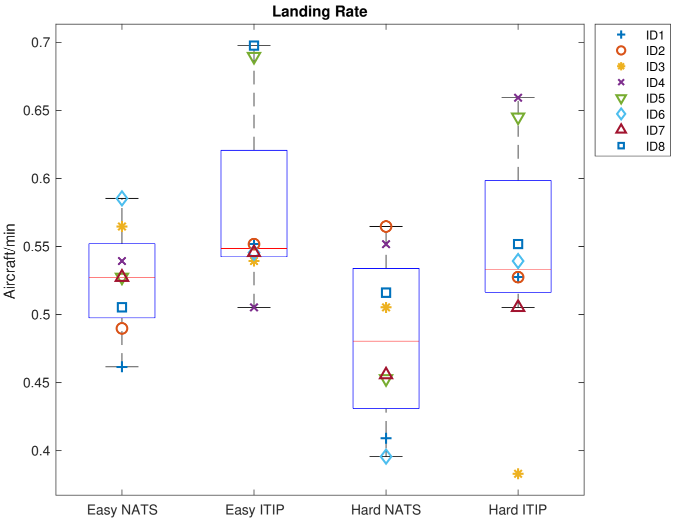


Fig. 24. The landing rate in aircraft per minute, per scenario

compare the average separation buffer for the four scenarios. For the easy scenarios, there was a significant impact of the used display on the separation buffer, Z(1) = 1.0, p = .017. The display type had a non-significant effect on the separation buffer for the hard scenarios, Z(1) = 26.0, p = 0.263.

In Figure 24, the average landing rate for each scenario can be found. A trend where the ITIP display results in a higher landing rate is visible, however, A Wilcoxon-Signed-Ranks test found a non-significant impact of used display on the landing rate for both the easy and hard scenarios; Z(1) = 27.0, p = .208 and Z(1) = 28.0, 0.161, respectively.

In Figure 25, the average point at which aircraft catch the ILS as measured from the runway threshold can be found. The red line depicts the minimum distance at which an aircraft flying at a standard approach altitude of 2,000 ft can intercept the ILS; 6.3 NM. A trend where the ITIP display results in ILS Catch closer to the runway can be observed.

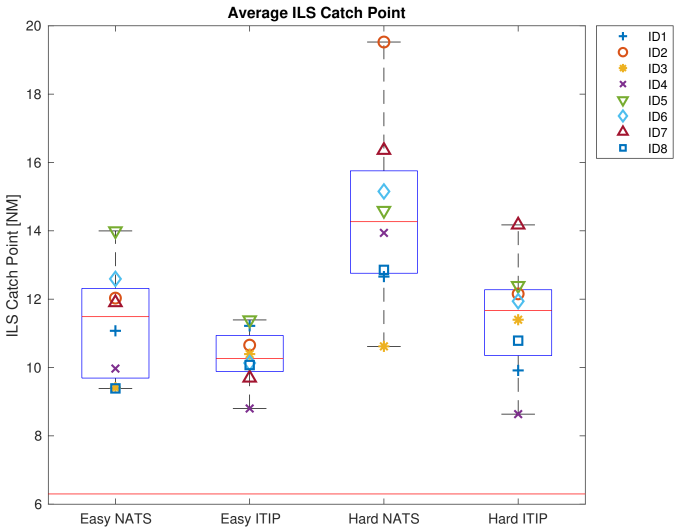


Fig. 25. The ILS catch point distance in nautical miles, per scenario

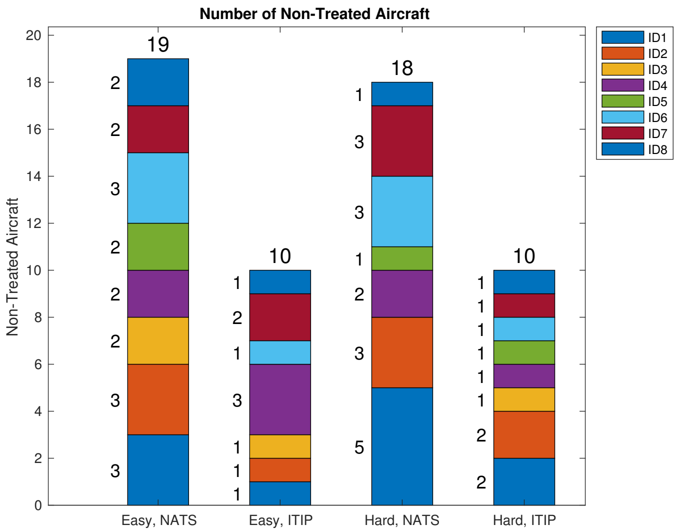


Fig. 26. The number of non-treated aircraft at the end of the scenario, per scenario

A Wilcoxon-Signed-Ranks test was carried out to compare the average ILS catch point for the four scenarios. For the easy scenarios, there was a non-significant impact of the used display on the ILS catch point, Z(1) = 6.0, p = .093. The display type did have a significant effect on the ILS catch point for the hard scenarios, Z(1) = 1.0, p = 0.017.

In Figure 26, a bar chart of the number of non-treated aircraft at the end of the scenario is given. A trend where the ITIP display results in less non-treated aircraft is visible.

In Figures 27, 28 and 29, the perceived overall, physical and mental workload for each scenario can be found, respectively. A trend where the ITIP display results in a lower physical and mental workload is visible. A Wilcoxon-Signed-Ranks test was carried out to compare the perceived physical workload for the four scenarios. For both the easy and hard scenarios, there was a non-significant impact of the used display on the

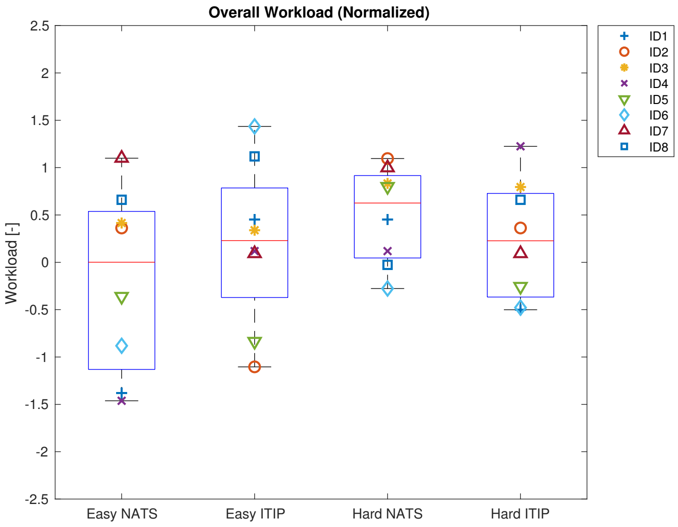


Fig. 27. The normalised perceived overall workload, per scenario

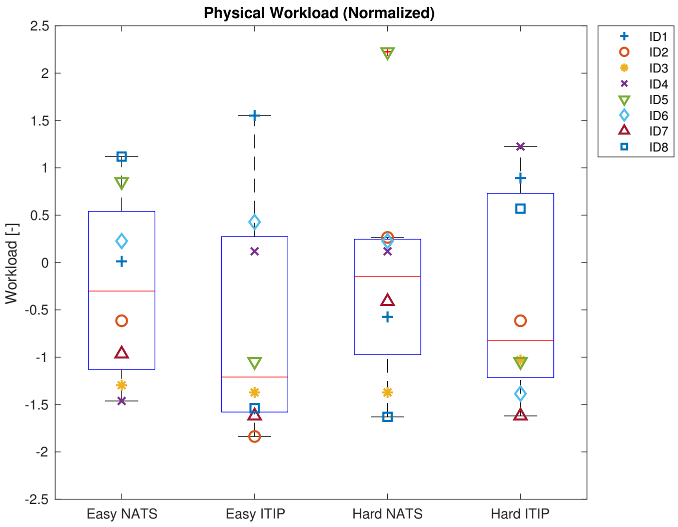


Fig. 28. The normalised perceived physical workload, per scenario

physical workload; Z(1) = 13.0, p = .484 and Z(1) = 16.0, p = .779, respectively. For the perceived mental workload, a Wilcoxon-Signed-Ranks test also yielded a non-significant impact of the used display on the mental workload for both the easy and hard scenarios; Z(1) = 23.0, p = .484 and Z(1) = 9.0, p = .208, respectively.

The decreased separation buffer, increased landing rate, decreased ILS catch point distance, reduced number of non-treated aircraft and reduced physical and mental workload lead to believe that the use of the ITIP display increases efficiency in this specific setup. For both the separation buffer and the ILS catch point, it looks as if the spread of the results is smaller for the ITIP display, in the easy scenarios. This could indicate more consistent performance of ATCos when using the ITIP in these scenarios. It would be interesting to investigate this effect for professional ATCos; increasing performance consistency would increase system predictability.

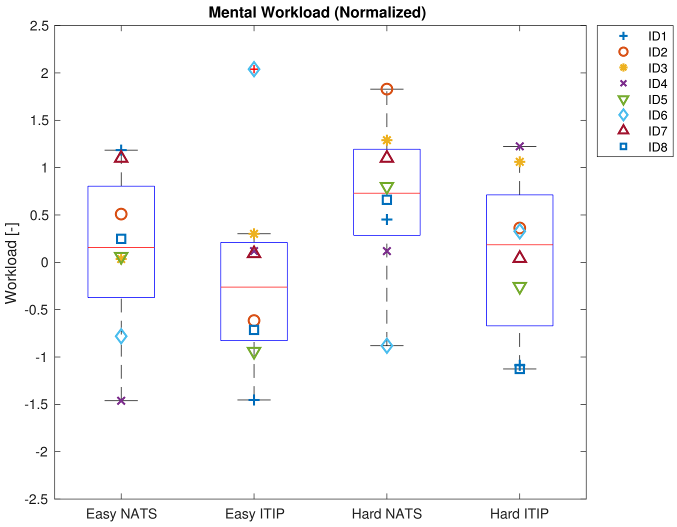


Fig. 29. The normalised perceived mental workload, per scenario

VIII. Discussion

When looking at the results, it can be concluded that the trends in the data are all in line with the hypotheses; use of the ITIP display maintains safety, increases efficiency and maintains general workload. For the workload, trends show a reduction of physical and mental workload when using the ITIP display. Some of the results were statistically significant, however further testing should be done with more and with better trained test participants to further investigate the hypotheses. It should be noted that this test was purely meant as a proof-of-concept; its results can not directly be extrapolated to real life APP ATC. In this section, the implications and real-life relevance of the results, as well as the experiment design itself are discussed and recommendations for future research are given.

A. Learning Effect

It was hypothesised that the observed effect where the use of the ITIP display decreased the number of TBS conflicts for some participants while increasing it for others might have been due to the order in which the participants tested the respective displays, in combination with the participants still improving their skill during the four test runs. To assess this learning effect, the number of TBS conflicts was evaluated per run in the order the test participants received them. The result can be found in Figure 30. As one can see, there was no significant general learning effect present in the number of conflicts. When investigating the individual participants, it can again be concluded that no learning effect is present.

To further investigate a possible learning effect, the separation buffer was evaluated per run, for which the results can be found in Figure 31. Again, no discernible learning effect is found over the four runs. It can thus be concluded that the difference in the number of TBS conflicts was not due to a learning effect. To conclude that no learning effect whatsoever was present would be premature; participants noted in the survey that they felt more training would have improved their performance.

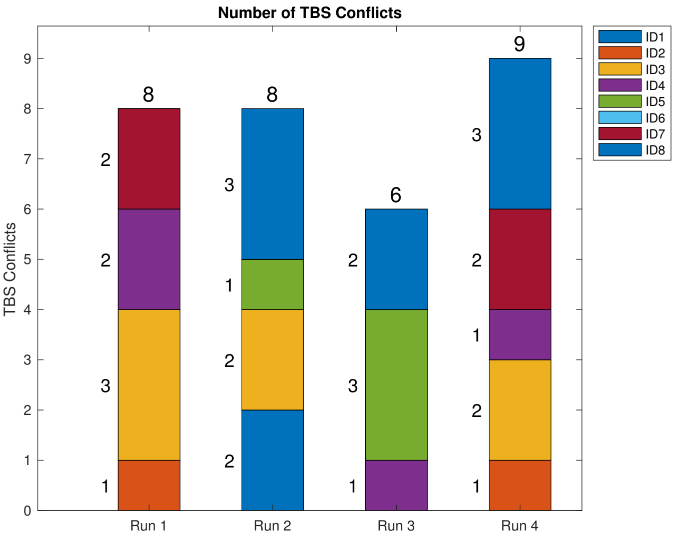


Fig. 30. The number of TBS conflicts, per run

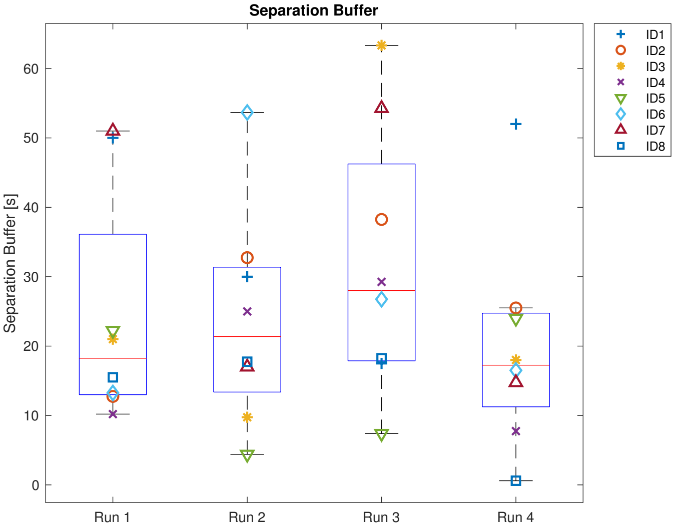


Fig. 31. The average separation buffer in seconds, per run

B. Display Use by Participants

During the experiments, it was noted that some participants used the trajectory prediction line as an advisory path, limiting their incentive to explore; participants followed the trajectory prediction as if it were the goal of the simulation. This is in line with what the participants thought about the display, one of them noting that he *‘used the display in a rule-based fashion [...] very rarely looking at the ghost or trying an optimal trajectory’*. Other participants were more active in looking for an optimal solution using the ghost, noting that *‘the ITIP helped me validate (and optimise) the control strategies that [he] had planned’*. Almost none of the participants made frequent use of the ability to change turn-in heading angles.

It can thus be concluded that some of the participants experienced restrictions from using the ITIP display, whilst others were able to use it to its full advantage, probing for the most optimal solution in the scenario. The participants who experienced restrictions did note that, if they would have had more training, they would most likely have more incentive to explore optimal solutions.

The restrictive nature of the display comes forth from the requirement of minimising display clutter; if the display is to be designed to be as minimalistic as possible, some assumptions have to be made and some restrictions applied. For instance, the ITIP display only plots the TTB points for the aircraft without a heading change. In the display, one ideal trajectory was emphasised and while users were able to explore multiple options, exploration was not actively encouraged by the visualisation. Plotting the TTB points in an area around the selected aircraft would show more possibilities in the system but it would also result in a lot of extra distracting information being shown to the controller. For future research, it should be investigated whether professional ATCos also experience restrictions while using the ITIP display, as it could very well be that professionals will ‘ignore’ the solution proposed by the display and using it in a more investigative manner. This is underscored by the fact that a professional APP controller at the LVNL noted he did make frequent use of the ability to change the turn-in heading angle, signalling that professional APP controllers might experience more incentive to explore based on their experience.

C. *Concept Maturity*

One of the design requirement for the ITIP display was that is could be implemented in the APP controller radar display at the LVNL without drastic alterations. A few considerations on implementation should be taken into account.

First of all, in this study, pilot response times were instantaneous, i.e., there was no time delay between the execution of the command by the controller and the execution by the aircraft. In reality, the delay between the command being received by the aircraft and the aircraft executing that command depends on multiple factors; the pilots, the airline company, the aircraft etc. The impact of these stochastic aircraft response times on the functioning on the ITIP display should be investigated.

Furthermore, an extra separation buffer should be considered to reduce TBS conflicts to zero. Although it an be expected that professional ATCos will not cause any TBS conflicts no matter what display they use, any alteration to their trusted work environment should be made with extreme caution. First, a safety buffer should be implemented, expanding the minimum separation by a few seconds. Next, the separation performance should be assessed with the ITIP display operational, in a real or in a simulated environment. Finally, the safety buffer should be tuned such that the probability of a TBS conflict occurring is at an acceptable level.

In the model, wind was modelled to be uniform. With minimal adaptions, detailed wind prediction models from, e.g., the Royal Dutch Meteorological Institute (KNMI) can be used for the trajectory prediction. After implementation, the display should be tested for robustness. It is expected that implementation of these non-uniform wind models will not drastically impact display functionality since it only impacts the integration done by the display algorithm. Uncertainties in the wind models can be of greater impact; inaccurate landing time predictions will most likely result in a need for larger buffers which will yield less efficient solutions.

In the simulations, aircraft were modelled to slow down to landing speed from 4DME to 130 kts automatically with a constant deceleration. In reality, the LVNL has data available on the speed profiles of aircraft in their final approach phase. These data must be used by the algorithm to calculate a predicted average approach speed for specific aircraft types and airlines. The uncertainty in landing time prediction this introduces should be investigated.

Finally, some participants noted reduced situation awareness, trusting the tool to do all the thinking for them. Participants noted that they were not paying attention to a.o. aircraft types and aircraft that were considered to be ‘dealt with’, e.g., aircraft on the ILS with a speed of 160 kts. If the display tool were to malfunction for some reason, the controller must be able to take over without losing control of the situation. The impact of the display on situation awareness must thus be assessed thoroughly before implementation.

D. *Experiment Design*

The experiment executed in this study was primarily meant as a proof-of-concept experiment. As such, it was conducted with limited resources. The sample size was relatively small and the training time of the participants was limited. This was noted by the participants as well, many of whom felt they needed more training. To further test the hypotheses, more and better trained participants need to be tested.

Due to the APP controller task being relatively difficult for the participants in this experiment, some simplifications were made in the scenarios. Altitude was not used as a variable (all aircraft arrived at FL20) and aircraft only arrived parallel to the ILS. More real-life scenarios, with aircraft arriving from all should be tested, with professional ATCos.

The participants that were chosen were non-professional ATCos. It should be noted that it is expected that these ATCos would profit more from the use of the ITIP display than professionals. The effect of the ITIP display on the performance of professional ATCos will be less pronounced, as professionals will most likely perform better in the baseline NATS scenario. The testing of professionals should thus focus more on how the display affects controller strategy and workload, as the impact on performance will be less evident.

IX. CONCLUSION

A display tool to aid APP controllers in attaining efficient Time-Based Separation on final approach was designed and evaluated in an initial proof-of-concept experiment. In contrast to the current industry standard which provides separation support for aircraft close to the ILS, this display aimed to aid controllers in the early stage of approach; support was provided from the moment the aircraft entered the TMA. The display was tested against the current industry standard designed by the National Air Traffic Services in the United Kingdom, currently operational at Heathrow Airport. The results are promising, showing trends of maintained safety, increased efficiency and reduced physical and mental workload. This suggests that early-stage strategy support for separation could potentially mitigate the feared performance gap between theory and practice when implementing TBS. Before implementation, more extensive testing should be done with professional ATCos. The impact of real-life factors such as pilot response times and imperfect approach speed models on the display robustness should be investigated, as well as the impact of the display on situation awareness. Safety buffers should be implemented and tuned to the measured performance in order to optimise separation, whilst maintaining safety at all times.

REFERENCES

[1] International Civil Aviation Organization. ICAO doc 4444, fourteenth edition. 2001.

[2] EUROCONTROL. RECAT-EU; European Wake Turbulence Categorisation and Separation Minima on Approach and Departure, Edition 1.2, Available: https://www.eurocontrol.int/sites/default/files/content/documents/sesar/recat-eu-released-september-2018.pdf, 2018.

[3] The European Commission. Commission Implementing Regulation (EU) No 716/2014, Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014R0716from=EN, 2014.

[4] U.K National Air Traffic Services. Enhanced Time Based Separation (eTBS), Available: http://www.nats.aero/wp-content/uploads/2017/03/Full_eTBS_PresentationV1.pdf, 2017.

[5] EUROCONTROL. Time-Based Separation (TBS) Solution and Controller Tool for Final Approach, Available: https://www.eurocontrol.int/sites/default/files/publication/files/time-based-separation-factsheet-2016.pdf, 2016.

[6] U.K. National Air Traffic Services (NATS). Time based separation, available: https://www.nats.aero/tbs/, 2018.

[7] U.K. National Air Traffic Services (NATS). Time-Based Separation at Heathrow, Available: https://nats.aero/blog/2014/02/explaining-time-based-separation-heathrow/, 2018.

[8] Onderzoeksraad voor Veiligheid. Neergestort tijdens nadering, boeing 737-800, nabij amsterdam schiphol airport, projectnummer m2009lv0225_01, 2009.

[9] Charles Morris, John Peters, and Peter Choroba. Validation of the Time Based Separation concept at London Heathrow Airport . *Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013), Chicago*, 2013.

[10] International Civil Aviation Organization. Wake turbulence aspects of Airbus A380-800 aircraft, Reference TEC/OPS/SEP 08-0294.SLG. 2008.

[11] Mahtab Ghazizadeh, John D. Lee, and Linda Ng Boyle. Extending the Technology Acceptance Model to assess automation. *Cogn Tech Work*, 14:39–49, 2012.

[12] Jean-Franois DArcy and Pamela S. Della Rocco. Air Traffic Control Specialist Decision Making and Strategic Planning A Field Survey, 2001.

II

Literature Review and Preliminary Research *(Graded under AE4020)*

1000
019A
6353
019A
1000
013A
1000
042
3237
047
1000
026A
3.8 095
3577
000A
3534
011A
3220
050
1000
029A
6251
000A
000A
0163
002A
2131
055
7000
006A
7000
005A
E

Design of an Ecological Human-Machine Interface for Final Approach using Time-Based Separation

Preliminary Thesis

M. Dirkzwager

Delft University of Technology

Design of an Ecological Human-Machine Interface for Final Approach using Time-Based Separation

Preliminary Thesis

by

M. Dirkzwager

to partially obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4206223	
Project duration:	May 23, 2018 – March 29, 2019	
Thesis committee:	Dr. Ir. C. (Clark) Borst,	TU Delft, First supervisor
	Prof. Dr. Ir. M. (Max) Mulder,	TU Delft, supervisor
	Dr. Ir. M. M. (René) van Paassen,	TU Delft, Supervisor

An electronic version of this thesis will be made available at `http://repository.tudelft.nl/`.

TU Delft

Preface

This report states the preliminary steps taken for a master thesis at the Delft Technical University, Faculty of Aerospace Engineering in the Control and Operations division. The thesis project was started on 23-05-2018 and will be finished on 29-04-2019.

I would like to take this moment to thank a few people who have contributed to this project. Firstly I would like to thank my supervisors from the TU Delft, Dr. Ir. Clark Borst, Prof. Dr. Ir. Max Mulder and Dr. Ir. René van Paassen, guiding me through this thesis project with all their useful insights and motivating enthusiasm. Next I would like to thank Ferdinand Dijkstra for providing the link between the TU Delft and the LVNL, and for providing me with all the practical knowledge I need concerning air traffic management. Finally, I would like to thank the LVNL for giving me the opportunity to perform a master graduation internship at the LVNL, enabling me to make the link between academics and practice.

Mats Dirkzwager
28-09-2018

List of Figures

2.1 Flight Information Regions in Europe, adapted from [5] 6
2.2 Sectors in the EHAA FIR, for lower (left) and upper (right) airspace [6] 7
2.3 Approximate locations of Initial Approach Fixes RIVER, SUGOL and ARTIP, together with location of Schiphol (SPL) 7
2.4 Size comparison of the Airbus A340-600 and the Boeing B767-300 [8], adapted from [10] and [11] 10
2.5 Comparison of minimum Distance-Based radar separation of a Boeing B767-300 and an Airbus A340-600 for current ICAO and RECAT-EU system [8], adapted from [10] and [11] 10
2.6 Schematic of aircraft division into RECAT-EU categories [8] 11
2.7 Examples of aircraft in each RECAT-EU category [8] 11
2.8 Runways at Schiphol Airport Adapted from [12] 12
2.9 Illustration of increasing aircraft separation using a dog-leg maneuver 14
2.10 Illustration of reducing aircraft separation using a tromboning maneuver 14
2.11 Screenshot of APP controller radar screen with IAFs and vector area for 18R 15
2.12 Schematic depiction of an ideal trajectory for an aircraft arriving from RIVER at Schiphol runway 18R, including ideal speed profile in red 16
2.13 Ground tracks of all aircraft landing on Schiphol runway 18R from 16:00 - 17:00 on 29-01-2017 . 17

3.1 Aircraft Distance-Based Separation in light headwind conditions. Values added for illustration purposes only [14] 20
3.2 Aircraft Distance-Based Separation in strong headwind conditions. Values added for illustration purposes only [14] 20
3.3 Aircraft Time-Based Separation in strong headwind conditions. Values added for illustration purposes only [14] 21
3.4 Depiction of the Optimised Runway Delivery market, used at Heathrow Airport, for 3 consecutive phases of flight: 1. both aircraft before 4DME, 2. Aircraft 1 after DME, 3. Aircraft 1 on runway 22
3.5 Depiction of the use of the ROT marker, as used by Heathrow Airport 23
3.6 Screenshot of informational movie by NATS on TBS at Heathrow, depicting the HMI used for TBS 24

4.1 The skills-rules-knowledge taxonomy, a tool for analysing worker competencies [18] 26
4.2 Short-, Medium-, and Long-term control for a pilot [19] 26
4.3 The abstraction hierarchy with the questions relative to each abstraction level, adapted from [23] 28
4.4 An Abstraction Hierarchy for the arrival controller work domain 29
4.5 Decision ladder for the arrival controller task 31
4.6 Information flow map for the strategies an arrival controller can take, including actor task division: green for human, yellow for computer and blue for shared 32
4.7 Information flow map for the path revision strategy, including actor task division: green for human, yellow for computer and blue for shared 33
4.8 Information flow map for the sped revision strategy, including actor task division: green for human, yellow for computer and blue for shared 33
4.9 A SRK taxonomy for the ARR controller 34

5.1 Illustration of an aircraft projected on a route of another aircraft, taken from [25] 37
5.2 Projection of an aircraft on a non co-linear route, taken from [25] 38
5.3 Division of air route legs between waypoints using parabolas, including four aircraft tracks and their segment transitioning points, adapted from [27] 39
5.4 Relative speed of ghost w.r.t. parent aircraft for parabola segment decision, taken from [27] . . . 39
5.5 Division of air route legs between waypoints, dividing in straight and corner regions with bisector, taken from [27] 40

5.6 Projection of aircraft position in a turn around the turn centre, taken from [27] 40
5.7 A point merge manoeuvre of an aircraft travelling on a sequencing leg [29] 41
5.8 Schematic of TBS margin visualisation concept with compressibility margin and minimum TBS, beginning and final aircraft locations, no wind 42
5.9 Schematic of TBS margin visualisation concept with compressibility margin and minimum TBS, beginning and projected final aircraft locations, with wind 43
5.10 Schematic of TBS margin visualisation concept with reduced clutter, using just a coloured band and separation markers, beginning and projected final aircraft locations, with wind 44
5.11 Schematic of second TBS margin visualisation concept with visualisation of separation reduction possibilities, beginning and projected final aircraft locations, with wind 45
5.12 Early illustration of TTB possibility visualisation concept, beginning and projected final aircraft locations 46
5.13 TTB display concept showing locomotion space where TTB is possible (in green), no headwind 47
5.14 TTB display concept showing locomotion space where TTB is possible (in green), 50 kts headwind 48
5.15 TTB display concept showing locomotion space where TTB is possible (in green), 50 kts headwind and two aircraft in traffic 48
5.16 TTB display concept showing locomotion space where TTB is possible (in green), 50 kts headwind yellow band for 30 second margin 49
5.17 TTB display concept showing ghosting of aircraft future location while user hovers around the TTB area 49
5.18 The decision ladder for the arrival controller task with implementation of the TTB display‘ . . . 51
5.19 The SRK taxonomy for the ARR controller task, as affected by the TTB display 52

6.1 A screenshot of two aircraft in a sector with (a) a waypoint 18R and (b) an ILS for runway 18R, as shown in Sector X 54
6.2 A screenshot of an aircraft in four different situations: (a) an unselected aircraft travelling in a straight line, (b) a selected aircraft travelling in a straight line, (c) an unselected descending aircraft and (d) a selected aircraft in a turn 55
6.3 The Control Display Unit in Sector X, adapted to include ILS catch function 55
6.4 Picture of the ILS catch algorithm, illustrating compensation for overshoot due to the large timestep used in the simulation 56

List of Tables

2.1 Distance-Based radar separation minima for different wake pairs, low headwind conditions (<5 kts), in nautical miles [9], [7]. Where no minimum is given, the MRS applies 9
2.2 Wake vortex categories based on aircraft MTOW [9], [7] 9
2.3 Distance-Based radar separation minima for different wake pairs based on RECAT-EU categories, low headwind conditions (<5 kts), in nautical miles. Where no minimum is given, the MRS applies [8] 12

C.1 Explanation of booleans used for identifying phase of arrival flight 65
C.2 Booleans that are used to identify a flight leg, per leg 66

List of Symbols & Abbreviations

Symbols

γ	Heat capacity ratio, 1.4 at 20	$[-]$
ρ	Density	kg/m^2
g	Gravitational Constant, 9.81	m/s^2
p	Pressure	Pa
r	Radius	m
V	Speed	m/s

Abbreviations

ACC	Area Control Center, in charge of CTA
AH	Abstraction Hierarchy
APP	Approach/Departure Controller, in charge of TMA
ARR	Arrival Controller
ATC	Air Traffic Control
ATCo	Air Traffic Controller
CAS	Calibrated Airspeed
CRDA	Converging Route Display Aid
CTA	Control Area, managed by ACC
CTR	Control Zone, managed by TWR
CWA	Cognitive Work Analysis
DBS	Distance-Based Separation
DBS	Distance-Based Separation
DCO	Departure Controller
DL	Decision Ladder
DME	Distance Measuring Equipment
DMI	Direct Manipulation Interface
EAS	Equivalent Airspeed
EID	Ecological Interface Design
ETA	Estimated Time of Arrival
FDR	Feeder
FIR	Flight Information Region

FL	Flight Level, FL 1 = 100 ft altitude
GS	Groundspeed
HMI	Human Machine Interface
IAF	Initial Approach Fix
IAS	Indicated Airspeed
ICAO	International Civil Aviation Organisation
ILS	Instrument Landing System
KNMI	’Koninklijk Nederlands Meteorologisch Instituut’ (Dutch Meteorological Institute)
LVNL	’Luchtverkeersleiding Nederland’ (Dutch ATC)
MRS	Minimum Radar Separation
MTOW	Maximum Take-Off Weight
NATS	National Air Traffic Services
ORD	Optimised Runway Delivery
R/T	Radio Telephony
RMS	Radar Minimum Separation
ROT	Runway Occupancy Time
RPI	Relative Position Indicator
SID	Standard Instrument Departure
SRK	Skill, Rule, Knowledge
TAS	True Airspeed
TBS	Time-Based Separation
TBS	Time-Based Separation
TMA	Terminal Control Area, managed by APP Controller
TTB	Turn To Base
TWR	Tower, in charge of CTR
UTA	Upper Control Area

Contents

List of Figures v

List of Tables vii

List of Symbols & Abbreviations ix

1 Introduction 1
- 1.1 Background 1
- 1.2 Problem Statement 2
  - 1.2.1 Research Question 2
  - 1.2.2 Research Approach and Scope 2
- 1.3 Report Structure 2

2 The Approach Controller's Work Domain 5
- 2.1 Airspace Division 5
- 2.2 The Approach/Departure Controllers 6
  - 2.2.1 The Approach/Departure Supervisor 6
  - 2.2.2 The Feeder/Departure Controller 6
  - 2.2.3 The Arrival Controller 8
- 2.3 Distance-Based Separation Minima 8
  - 2.3.1 ICAO Distance-Based Separation Minima 8
  - 2.3.2 RECAT-EU Distance-Based Separation minima 9
- 2.4 Schiphol Runway 18R 9
- 2.5 Arrival Controller Best Practices 12
  - 2.5.1 Arrival Controller Strategies 12
  - 2.5.2 Nominal Approach Trajectory 13
  - 2.5.3 Arrival Controller Limitations 16

3 Time-Based Separation 19
- 3.1 The Time-Based Separation Concept 19
  - 3.1.1 Explanation of the Time-Based Separation Concept 19
  - 3.1.2 Advantages and Disadvantages of Time-Based Separation 21
- 3.2 Time-Based Separation at Heathrow Airport 22

4 Cognitive Work Analysis 25
- 4.1 On Ecological Interface Design Concept 25
  - 4.1.1 The Ecological Interface Design Concept 25
  - 4.1.2 Advantages and Disadvantages of Ecological Interface Design 26
- 4.2 Cognitive Work Domain Analysis 27
- 4.3 Control Task Analysis 29
- 4.4 Strategies Analysis 30
- 4.5 Social Organisation and Cooperation Analysis 32
- 4.6 Worker Competencies Analysis 34

5 Concept Design 37
- 5.1 Previous Work 37
  - 5.1.1 Relative Position Indication 37
  - 5.1.2 The Point Merge Concept 40
- 5.2 Preliminary Concept Displays 41
  - 5.2.1 Display Design Considerations 41
  - 5.2.2 Separation Margin Visualisation 42
  - 5.2.3 Separation Reduction Possibilities Visualisation 43
  - 5.2.4 Turn To Base Possibilities Visualisation 45

5.3 The Turn To Base Display 46
    5.3.1 Offline Simulations 46
    5.3.2 Variations on the TTB Display 48
    5.3.3 The TTB Display in the Cognitive Work Analysis 49
    5.3.4 Critical Analysis and Hypothesis 51

6 Preliminary Work 53
6.1 ATC Simulator Sector X 53
    6.1.1 Aircraft Visualisation in Sector X 53
    6.1.2 Adapting Sector X 54
6.2 Trajectory Prediction 56

7 Future Work 57
7.1 Finalising Sector X Adaption 57
7.2 Update Trajectory Prediction 57
7.3 Display Design and Testing 57
7.4 Project Planning 58

8 Concluding Remarks 59
8.1 Concluding Remarks 59
8.2 Preliminary Answers to Research Question 59

A Aircraft in RECAT-EU Categories 61

B Airspeed Definitions 63
B.1 Indicated Airspeed 63
B.2 Calibrated Airspeed 63
B.3 Equivalent Airspeed. 63
B.4 True Airspeed 63
B.5 Groundspeed 64

C Trajectory Prediction in Sector X: Leg Estimation 65

D Planning Gantt Chart 67

Bibliography 69

1

Introduction

1.1. Background

In the Netherlands, airspace is divided in several sectors. For each sector, an air traffic controller (ATCo) is responsible for guiding the aircraft towards their waypoint whilst keeping all aircraft in that sector safely separated. The ATCo's task is to provide efficient and expeditious flow of traffic whilst ensuring that a certain, predefined distance is kept between aircraft at all times. This last practice is called Distance-Based Separation (DBS). Maintaining separation is in the first place important to prevent mid-air collisions and in the second place because of the air turbulence that aircraft create behind them due to wing-tip vortices. An aircraft flying too close behind another aircraft may experience loss of control due to this phenomenon. This is true for every aircraft flying directly behind another aircraft or for aircraft crossing paths. As such, it is of special importance when aircraft are flying close behind each other, such as in the final phase of flight, the approach phase.

In the final phase of flight, an Arrival (ARR) controller is responsible for guiding aircraft to the runway threshold. In this process, the ARR controller must keep a predefined distance between all aircraft. These minimum distances are defined per aircraft pair; a heavy aircraft can fly close to a light aircraft whilst a light aircraft has to keep a relatively large distance from a heavy aircraft. This is because heavy aircraft create larger wake vortices and light aircraft are less resilient to turbulent air.

A problem occurs when fast headwinds (>25 kts) are present at the runway area. In this situation, the groundspeed of the aircraft is significantly reduced. This results in a decreased landing rate, since aircraft still need to maintain the same distances between each other.

Research from the National Aviation Traffic Service (NATS) pointed out that this reduced landing rate due to strong headwinds is one of the largest factors in delays at large airports [1] and it can be assumed that this is also the case for Schiphol Airport. An internal, unpublished research performed by To70 Aviation Consultants for the Dutch ATC (LVNL) pointed out that Schiphol loses as much as 55 to 85 movements per day on stormy days [2].

This problem is partly solved when aircraft keep a predefined time between them instead of a predefined distance; this is called Time-Based Separation (TBS). Using TBS, aircraft may fly closer to each other in large headwind conditions, theoretically maintaining the same landing rate as for low headwind conditions.

In multiple analyses, TBS has been proven to increase landing rate in large headwind conditions [2] [3] [4]. However, experts fear that the practical gain in runway throughput will be less than the theoretical gain, since ARR controllers are not used to separating based on time. Separating on time is less intuitive than separating based on distance. After all, on the radar screen which the ARR controller uses to monitor all aircraft states, distance can be seen whilst time is not directly a visible variable.

1.2. Problem Statement

Due to the non-intuitive nature of TBS, and due to the workload of ATCos being very high already, the question arose for a tool assisting ARR controllers in obtaining optimum time-based separation.

First, a research question for this research must be devised, along with some clarifying subquestions.

1.2.1. Research Question

The main research question for this research is defined as follows:

’How can the Human Machine Interface for an Air Traffic Arrival Controller be improved to best assist the arrival controller in obtaining optimum time-based separation between aircraft during the final approach phase?’

Subquestions for this research are:

1. What are current methods and best practices to maintain safe separation during the final approach phase?
2. How is optimum time-based separation defined?
3. What problems arise when using time-based separation instead of distance-based separation, and what information does the controller need to solve these problems?
4. How is this information best visualised?

The research objective is given as:

’To design, build an test a Human Machine Interface , using the principles of Ecological Interface Design, which assists the approach controller in obtaining optimum time-based separation.’

1.2.2. Research Approach and Scope

The approach of this research is as follows. The principles of Ecological Interface Design will be applied to the design of a display which will aim to aid the ARR controller in obtaining optimum separation. The display will be designed in such a way that the controller can, in one single observation, observe what strategies he/she can take to enforce the most critical goal of the ARR controller at that time, be it separating aircraft, merging traffic streams or steering aircraft towards the runway threshold.

Since the success of TBS relies heavily on timing in the final phase of flight, this research will focus on developing a tool for the Arrival controller. Other controllers ave different needs and therefore need different display designs. Next to focusing on the ARR controller, this research will focus on Schiphol runway 18R, more commonly known as The ’Polderbaan’. In the initial phase, only one stream of aircraft will be considered. This research will thus only aim to develop the concept and perform initial tests to prove its usefulness. Expansion of the concept, such as applying the concept to more runways, and extensive testing are beyond the scope of this work.

1.3. Report Structure

This report is structured as follows. In Chapter 2, the work domain of air traffic controllers in The Netherlands is analysed. Here, special focus lies on the domain of the ARR controller.

In Chapter 3, the phenomenon of TBS is explained in more detail.The situation of Heathrow Airport is analysed, since this airport already uses TBS.

In Chapter 4, a cognitive work analysis will be performed. This analysis provides insight in the work domain and the task of an ARR controller, in a way that is relevant for display design.

In Chapter 5, the preliminary concept will be developed. This will be done by evaluating previous relevant work, after which some initial concepts will be devised. One concept has been chosen and has been treated in more detail.

In Chapters 6 and 7, the preliminary work that has been done and the work that shall be done in the future is discussed, respectively. Finally, in Chapter 8, some concluding remarks are given.

2

The Approach Controller's Work Domain

In this chapter, the work domain of the approach controller will be analysed. In Section 2.1, the division of Dutch airspace will be explained. In Section 2.2, the task distribution between different types of approach controllers will be handled. In Section 2.3, the Distance-Based separation concept will be treated an in Section 2.4 some information will be given on Schiphol runway 18R. Finally, in Section 2.5, the best practices for arrival controllers will be treated.

Note that on the subject of ATC in The Netherlands, not much literature is published. Therefore, a lot of information for this research is obtained via interviews with experts at the LVNL and internal, unpublished material of the LVNL [2].

2.1. Airspace Division

Airspace around the world is divided in Flight Information Regions (FIRs). The Netherlands is entirely enclosed in one FIR: the Amsterdam 'EHAA' FIR. The EHAA FIR, together with the other FIRs in Europe, can be found in figure 2.1.

Airspace in a FIR can be controlled or uncontrolled. The main difference between controlled and uncontrolled airspace is that for uncontrolled airspace, Air Traffic Control (ATC) service is not necessary or possible. Aircraft have to maintain separation from other aircraft themselves. In controlled airspace, ATC provides separation services to aircraft that fly on instruments (Instrument Flight Rules, IFR). This enables IFR aircraft to fly through areas with limited visibility, such as clouds. Aircraft that maintain separation by visual observation (Visual Flight Rules, VFR) are required to maintain separation themselves. To enter controlled airspace, a pilot needs clearance from the Air Traffic Controller (ATCo) in that airspace.

In vertical direction, airspace is organised as follows. Above FL 195 (Flight Level 195 = 19,500 ft altitude) the airspace is called the Upper Control Area (UTA). In this area, general Air Traffic Control (ATC) is responsible for the flow of traffic. This area stretches across FIRs; Eurocontrol is an example of such a general ATC.

The airspace below FL 195 is called a Control Area (CTA). In this area, the Area Control Center (ACC) is responsible for the flow of traffic. This area falls within a FIR. In The Netherlands, the 'Luchtverkeersleiding Nederland' (Dutch Air Traffic Control), or LVNL for short, is the ACC (except for in military areas).

Inside a CTA, below FL 105 and above FL 15, a Terminal Control Area (TMA) is present around an airport. In this airspace, Approach/Departure (APP) controllers are responsible for the flow of traffic. The APP controllers provide the connection between the CTA and the Control Zone (CTR), where the tower (TWR) is responsible for safe traffic on the ground. The APP controllers are thus responsible for the last and first phases of flight.

On the ground, an aircraft is managed by the local ATC, or tower (TWR). This happens in the Control Zone (CTR), which is often a circular area around an airport. The TWR is responsible for all taxiing aircraft and for

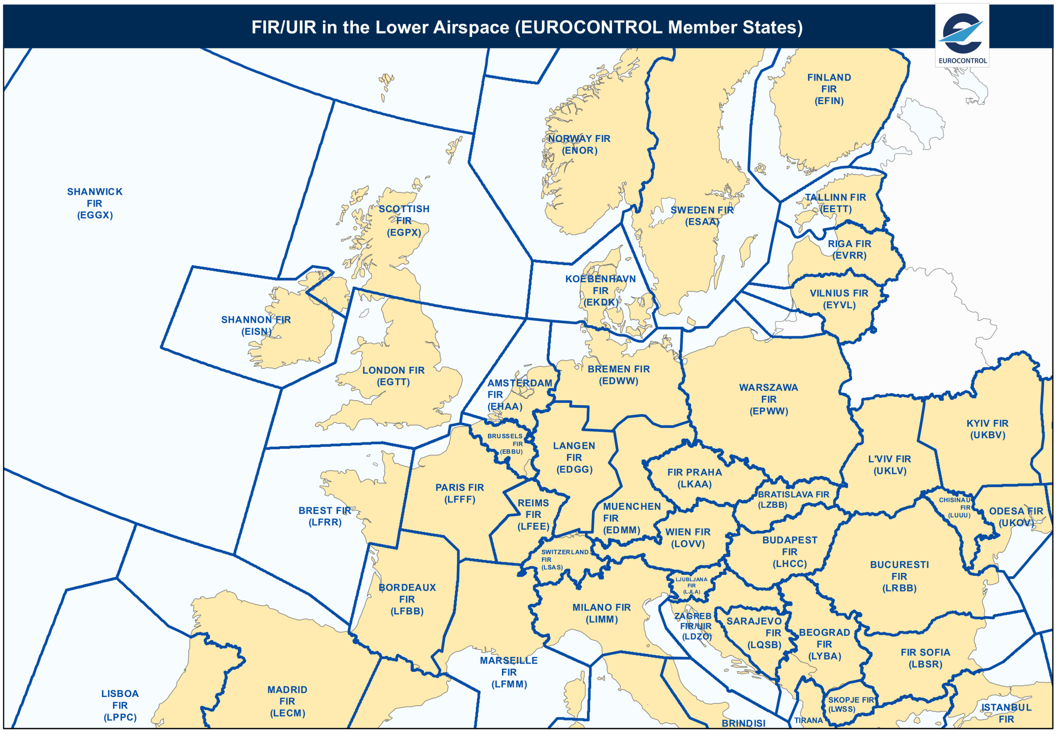

FIR/UIR in the Lower Airspace (EUROCONTROL Member States)
EUROCONTROL
SHANWICK FIR (EGGX)
SCOTTISH FIR (EGPX)
SHANNON FIR (EISN)
LONDON FIR (EGTT)
NORWAY FIR (ENOR)
SWEDEN FIR (ESAA)
FINLAND FIR (EFIN)
TALLINN FIR (EETT)
RIGA FIR (EVRR)
VILNIUS FIR (EYVL)
KOEBENHAVN FIR (EKDK)
BREMEN FIR (EDWW)
AMSTERDAM FIR (EHAA)
BRUSSELS FIR (EBBU)
LANGEN FIR (EDGG)
WARSZAWA FIR (EPWW)
KYIV FIR (UKBV)
L'VIV FIR (UKLV)
FIR PRAHA (LKAA)
BRATISLAVA FIR (LZBB)
CHISINAU FIR (LUUU)
ODESA FIR (UKOV)
PARIS FIR (LFFF)
REIMS FIR (LFEE)
MUENCHEN FIR (EDMM)
WIEN FIR (LOVV)
BUDAPEST FIR (LHCC)
BUCURESTI FIR (LRBB)
BREST FIR (LFRR)
SWITZERLAND FIR (LSAS)
LJUBLJANA FIR (LJLA)
BORDEAUX FIR (LFBB)
MILANO FIR (LIMM)
ZAGREB FIR/UIR (LDZO)
SARAJEVO FIR (LQSB)
BEOGRAD FIR (LYBA)
FIR SOFIA (LBSR)
MARSEILLE FIR (LFMM)
LISBOA FIR (LPPC)
MADRID FIR (LECM)
SKOPJE FIR (LWSS)
BRINDISI
TIRANA
ISTANBUL FIR

Figure 2.1: Flight Information Regions in Europe, adapted from [5]

guiding them to the right terminal.

The EHAA FIR is divided into smaller sectors. Distinction is made between the Upper airspace (above FL 195), and lower airspace (below FL 195). The division of the EHAA FIR into sectors for the upper and lower airspace can be found in figure 2.2.

2.2. The Approach/Departure Controllers

For the last phase of its flight, an aircraft is handled by approach/departure controllers. At the LVNL, the APP task executed by three controllers, each having their own task. The three functions for the APP controllers are:

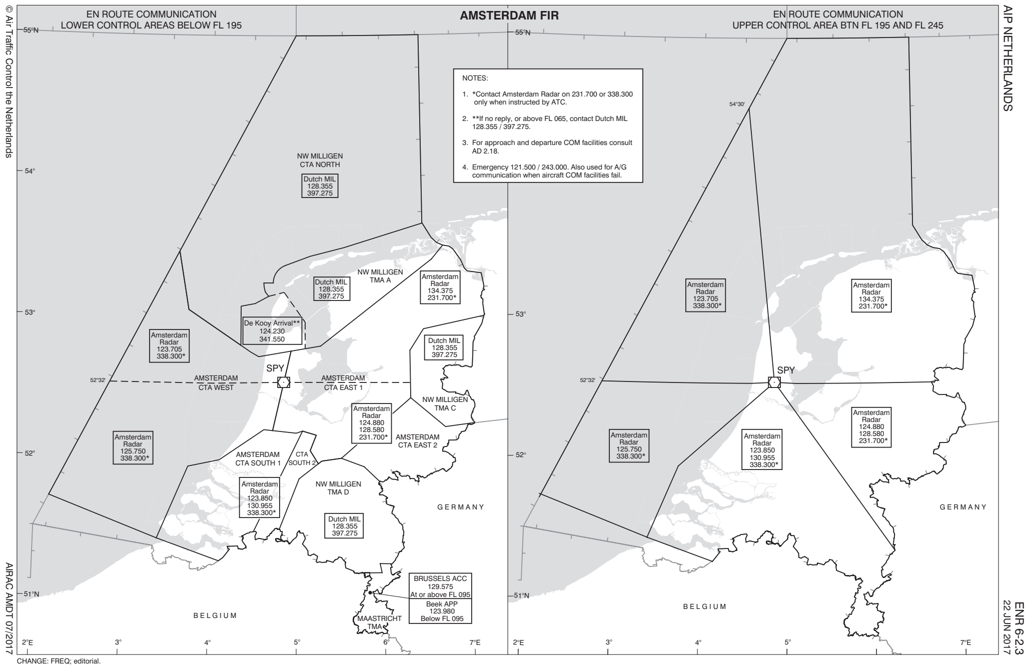
- The APP Supervisor (APP-SUP)
- The Feeder (FDR) and Departure Controller (DCO)
- The Arrival Controller (ARR)

2.2.1. The Approach/Departure Supervisor

The Approach/Departure Supervisor is responsible for planning of aircraft streams. The supervisor decides when aircraft can enter the TMA or when aircraft need to hold before entering. The supervisor (aided by computer tools) assigns every aircraft to a runway and decides the order in which aircraft will land (sequencing). The order of aircraft is mostly based on first-come-first-serve basis. Sequencing is done for i.a. load balancing reasons. Load balancing here means ensuring that the aircraft are (approximately) equally divided over the runways. The supervisor does not control any aircraft; he/she just plans where the aircraft should go and when they should go there.

2.2.2. The Feeder/Departure Controller

The Feeder (FDR) and Departure Controller (DCO) is responsible for guiding the aircraft streams from the Initial Approach Fix (IAF) to the vector area of the runway to which the aircraft is assigned. Note that the FDR and DCO tasks are different, but at the LVNL they are grouped together and assigned to one controller. The


© Air Traffic Control the Netherlands
AMSTERDAM FIR
EN ROUTE COMMUNICATION
LOWER CONTROL AREAS BELOW FL 195
EN ROUTE COMMUNICATION
UPPER CONTROL AREA BTN FL 195 AND FL 245
AIP NETHERLANDS
55°N
54°
53°
52°
51°N
52°32'
54°30'
2°E
3°
4°
5°
6°
7°E
NW MILLIGEN
CTA NORTH
Dutch MIL
128.355
397.275
NW MILLIGEN
TMA A
Amsterdam
Radar
134.375
231.700*
De Kooy Arrival**
124.230
341.550
Amsterdam
Radar
123.705
338.300*
SPY
AMSTERDAM
CTA WEST
AMSTERDAM
CTA EAST 1
NW MILLIGEN
TMA C
Amsterdam
Radar
124.880
128.580
231.700*
AMSTERDAM
CTA EAST 2
Amsterdam
Radar
125.750
338.300*
AMSTERDAM
CTA SOUTH 1
CTA
SOUTH 2
Amsterdam
Radar
123.850
130.955
338.300*
NW MILLIGEN
TMA D
GERMANY
BELGIUM
BRUSSELS ACC
129.575
At or above FL 095
Beek APP
123.980
Below FL 095
MAASTRICHT
TMA
AIRAC AMDT 07/2017
ENR 6-2.3
22 JUN 2017

NOTES:

1. *Contact Amsterdam Radar on 231.700 or 338.300 only when instructed by ATC.
2. **If no reply, or above FL 065, contact Dutch MIL 128.355 / 397.275.
3. For approach and departure COM facilities consult AD 2.18.
4. Emergency 121.500 / 243.000. Also used for A/G communication when aircraft COM facilities fail.

CHANGE: FREQ; editorial.

Figure 2.2: Sectors in the EHAA FIR, for lower (left) and upper (right) airspace [6]

reason for this is that both the FDR and DCO tasks manage streams of aircraft inside the TMA, outside the runway vector areas (the workspaces of the Arrival Controllers). As such, it does not matter if the flights are in- or outbound; the tasks both entail keeping streams of aircraft safely separated.

When an aircraft wants to initiate its approach, the ACC delivers the aircraft to an Initial Approach Fix (IAF). This is a waypoint where the Approach and Departure Controllers take over from the ACC. At the IAF, the aircraft officially starts its approach. In the EHAA FIR, the three possible IAFs are RIVER, ARTIP and SUGOL. Their (approximate) location can be seen in figure 2.3.

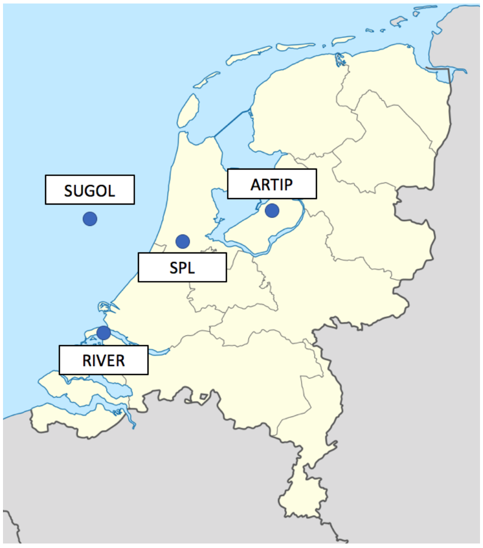

SUGOL
ARTIP
SPL
RIVER

Figure 2.3: Approximate locations of Initial Approach Fixes RIVER, SUGOL and ARTIP, together with location of Schiphol (SPL)

The FDR task is to safely guide the aircraft from the IAF to the vectoring area of the runway to which the aircraft is assigned. The vector area is the workspace of the Arrival controller. The FDR does not divert from the sequencing unless traffic demands it.

The DCO task is approximately the same as the FDR task, but reversed. Sequencing and take-off control for departing flights is done by the TWR. The DCO is responsible for safely guiding the departing flights out of the TMA into the Standard Instrument Departure (SID) routes.

2.2.3. The Arrival Controller

The Arrival Controller (ARR) works in so called vector areas. This is an area defined around a runway in which aircraft are vectored to final approach. This area is defined per runway; no vector area is the same. Handing over an aircraft from the FDR to the ARR happens on the border of a vector area. An ARR is always responsible for only one vector area and thus only for one runway.

The ARR controller's task is to guide the aircraft to the runway threshold whilst maintaining aircraft separation. What this separation should be is prescribed by several rules, described in Section 2.3. The controller guides the aircraft to the runway by vectoring the aircraft towards the localizer and glideslope, which together form the Instrument Landing System (ILS). These are predefined paths in the airspace preceding a runway which the aircraft can sense using radio receivers. The localizer places an aircraft in front of the runway, the glideslope defines the perfect altitude path for an aircraft to follow (3 degrees from the touchdown point). The localizer and glideslope (ILS) are visualised on the ARR controller radar screen by a green line with dots at every 2 NM, see figure 2.11.

2.3. Distance-Based Separation Minima

Currently, most aircraft separation is done based on distance. This means that aircraft must always keep a certain distance between itself and other aircraft. This is called Distance-Based Separation, or DBS. Aircraft can also be separated based on time, this will be treated in chapter 3 and is called Time-Based Separation (TBS). Currently, only Heathrow Airport is using TBS.

There are two ways to determine the minimum separation distance: based on minima prescribed by the International Civil Aviation Organization (ICAO) and based on minima prescribed by the newer RECAT-EU, which is a relatively new expansion of the ICAO method. Currently, most airports work with the ICAO separation minima; the use of the RECAT-EU separation minima must start in 2024, as per a regulation of the European Union.

2.3.1. ICAO Distance-Based Separation Minima

For the ICAO separation minima, the minimum separation between a leading and a following aircraft depends on the Maximum Take-Off Weight (MTOW) of both aircraft. Since wake vortices created by aircraft tend to get bigger with increasing MTOW, and since a lower MTOW infers a higher susceptibility to turbulent airflows, lighter aircraft must keep more distance from heavier aircraft than heavier aircraft must keep from light aircraft. When wake vortices do not affect the follower aircraft, the Minimum Radar Separation (MRS) must be adhered to at all times. The MRS is the distance at which aircraft need to be separated in order for the operational radar system to be able to tell the two aircraft apart. The MRS at Schiphol is 3 NM. With newer radar systems, a MRS of 2.5 NM can be attained. Note the difference between the MRS, which is the minimum separation imposed by technical limitations of the the radar system, and the radar minimum separation (RMS), which is the minimum separation imposed by wake vortices when an aircraft is under radar surveillance, in this case the ICAO DBS separation minima.

In table 2.1, the separation minima for different combinations of following and leading aircraft (so called wake pairs), are given. In table 2.2, the classification into categories of aircraft based on their MTOW is given. Note that the 'super' category is not implemented in the international regulations as of yet. However, since the introduction of the Airbus A380-800, in 2008, the International Civil Aviation Organization (ICAO) issued a State guidance strongly advising implementation of the 'Super' category [7]. The 'Super' category is reserved

Table 2.1: Distance-Based radar separation minima for different wake pairs, low headwind conditions (<5 kts), in nautical miles [9], [7]. Where no minimum is given, the MRS applies

[NM]	**Follower**			
Leader	**Super**	**Heavy**	**Medium**	**Light**
Super	-	6	7	8
Heavy	-	4	5	6
Medium	-	-	-	5
Light	-	-	-	-

Table 2.2: Wake vortex categories based on aircraft MTOW [9], [7]

Aircraft	**MTOW [kg]**
Super	Airbus A380 & Antonov AN-124
Heavy	> 136,000
Medium	7,000 - 136,000
Light	< 7,000

specifically for the Airbus A380 and the Antonov AN-124 [8].

2.3.2. RECAT-EU Distance-Based Separation minima

To optimise aircraft throughput, Eurocontrol has developed a re-categorisation of the current wake vortex categories. This re-categorisation is called RECAT-EU, and has to be implemented at all European airports in 2024.

The principle of RECAT-EU is to divide aircraft in more categories than the current four. With the current separation minima, a relatively large aircraft in the 'Heavy' category will have an overly conservative separation from a relatively small aircraft in the 'Heavy' category.

To illustrate, in figure 2.4, the size of a Boeing B767-300 and an Airbus A340-600 is compared. While these aircraft both fall into the 'Heavy' category for the standard ICAO separation minima, the difference between the wingspan of these aircraft is about 15 meters. This will result in an overly conservative separation, when the A340 (bigger) to fly behind the B767 (smaller). By introducing more categories, the separation can be more specific for aircraft pairs, resulting in a smaller minimum separation in some cases, see figure 2.5. The reduced separation minima result in higher landing and take-off rates and thus in a higher throughput per runway.
The RECAT-EU category division is not solely based on MTOW. Instead, after dividing the aircraft in 'Light', 'Medium' or 'Heavy' classes based on the MTOW, the aircraft are then further divided into CAT 'A', 'B', 'C', 'D', 'E' or 'F', based on wingspan. In this division, CAT-A aircraft create the most severest wake vortices, whilst CAT-F aircraft create small wake vortices. Aircraft can be moved from class 'E' to 'D', as well as from class 'C' to 'B' and from class 'B' to 'A', based on individual inspection. This division process is illustrated in figure 2.6. This results in a division which is more precise. Examples of aircraft in their respective category can be found in figure 2.7. In Appendix A, a more extensive list of aircraft in their RECAT-EU categories can be found.

The aircraft in their RECAT-EU categories form wake pairs or which the minimum separation can be seen in table 2.3.

2.4. Schiphol Runway 18R

Schiphol Airport has 6 runways. For each runway, both ends are coded based on their heading angle and relative position. For example, the southern end of the most western runway is called '36L' because if an aircraft wants to land on this runway, its heading must be approximately 360 degrees and the runway is on the left relative to the other runways that have this heading. The runways and their respective codes can be seen in figure 2.8.

This research will focus on runway 18R, also called the 'Polderbaan'. From an internal study conducted by

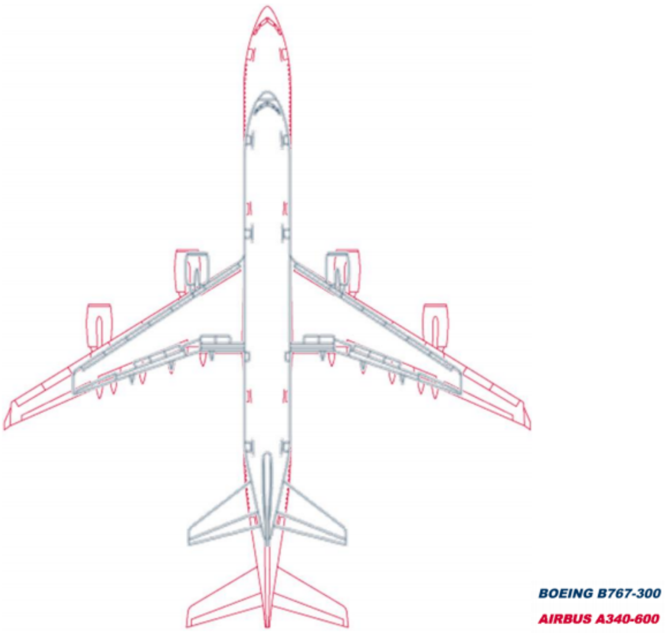

BOEING B767-300
AIRBUS A340-600

Figure 2.4: Size comparison of the Airbus A340-600 and the Boeing B767-300 [8], adapted from [10] and [11]

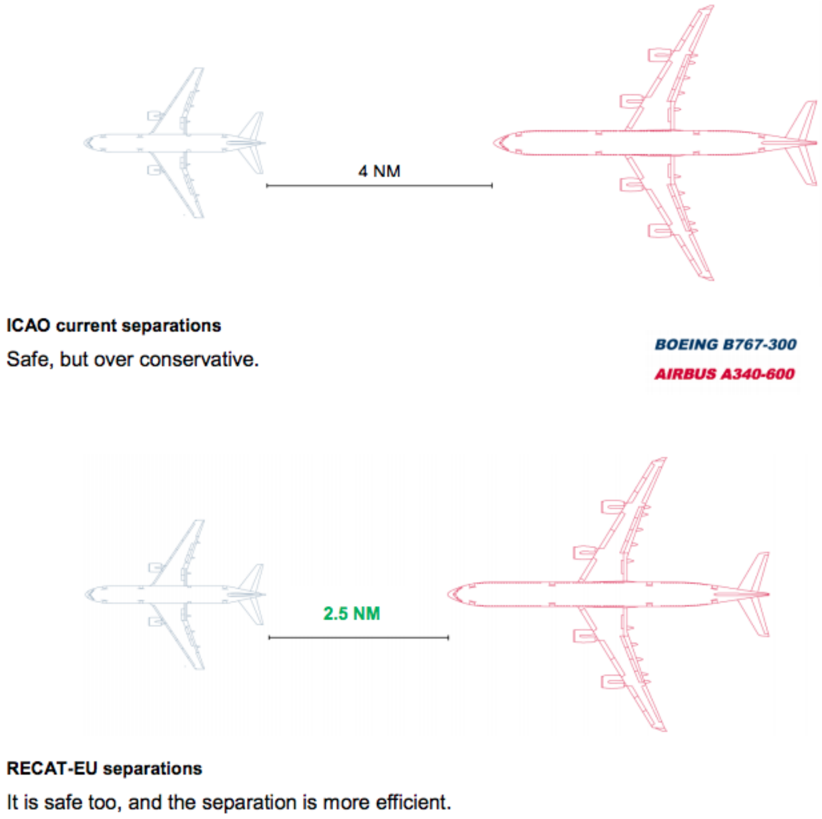

4 NM
ICAO current separations
Safe, but over conservative.
BOEING B767-300
AIRBUS A340-600
2.5 NM
RECAT-EU separations
It is safe too, and the separation is more efficient.

Figure 2.5: Comparison of minimum Distance-Based radar separation of a Boeing B767-300 and an Airbus A340-600 for current ICAO and RECAT-EU system [8], adapted from [10] and [11]

To70 Aviation Consultants it became clear that this runway is in use about 53% (highest usage of all runways) of the time and experiences an average headwind of 7 kts. On average, this runway experiences strong head-

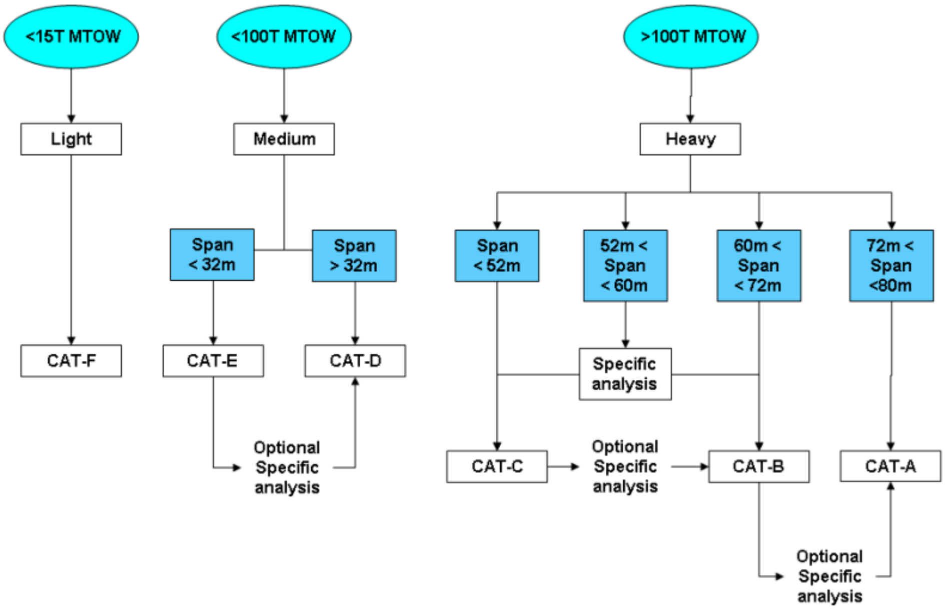

<15T MTOW
<100T MTOW
>100T MTOW
Light
Medium
Heavy
Span < 32m
Span > 32m
Span < 52m
52m < Span < 60m
60m < Span < 72m
72m < Span <80m
CAT-F
CAT-E
CAT-D
Specific analysis
Optional Specific analysis
CAT-C
Optional Specific analysis
CAT-B
CAT-A
Optional Specific analysis

Figure 2.6: Schematic of aircraft division into RECAT-EU categories [8]

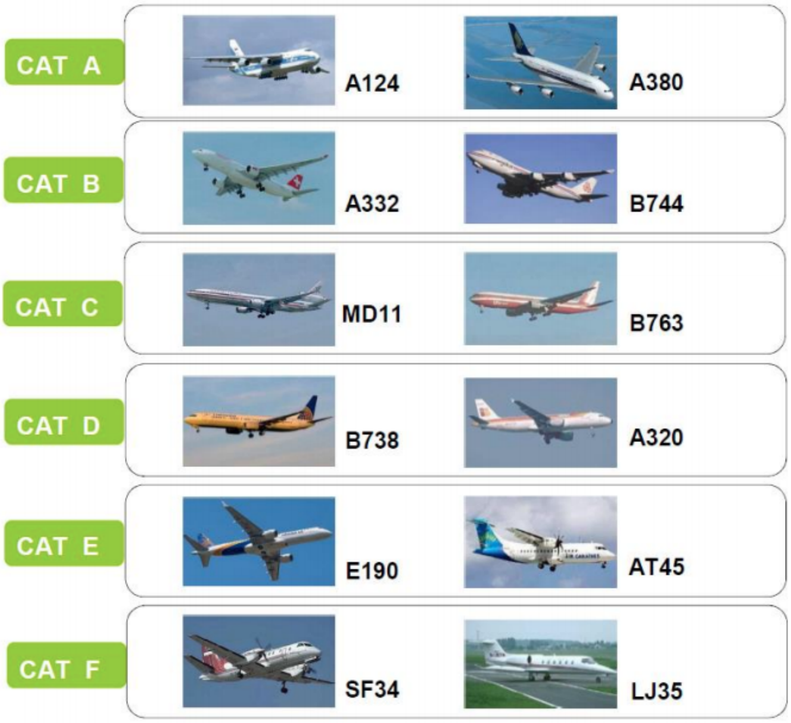

CAT A
A124
A380
CAT B
A332
B744
CAT C
MD11
B763
CAT D
B738
A320
CAT E
E190
AT45
CAT F
SF34
LJ35

Figure 2.7: Examples of aircraft in each RECAT-EU category [8]

winds (+20 kts) about 3% of the time [2]. This research will focus on this runway, with the assumption that the 18R runway is one of the most critical runways. The results of this research can hopefully be extrapolated to other runways as well.

Table 2.3: Distance-Based radar separation minima for different wake pairs based on RECAT-EU categories, low headwind conditions (<5 kts), in nautical miles. Where no minimum is given, the MRS applies [8]

[NM]	**Follower**					
Leader	**CAT-A**	**CAT-B**	**CAT-C**	**CAT-D**	**CAT-E**	**CAT-F**
CAT-A	3	4	5	5	6	8
CAT-B	-	3	4	4	5	7
CAT-C	-	-	3	3	4	6
CAT-D	-	-	-	-	-	5
CAT-E	-	-	-	-	-	4
CAT-F	-	-	-	-	-	3

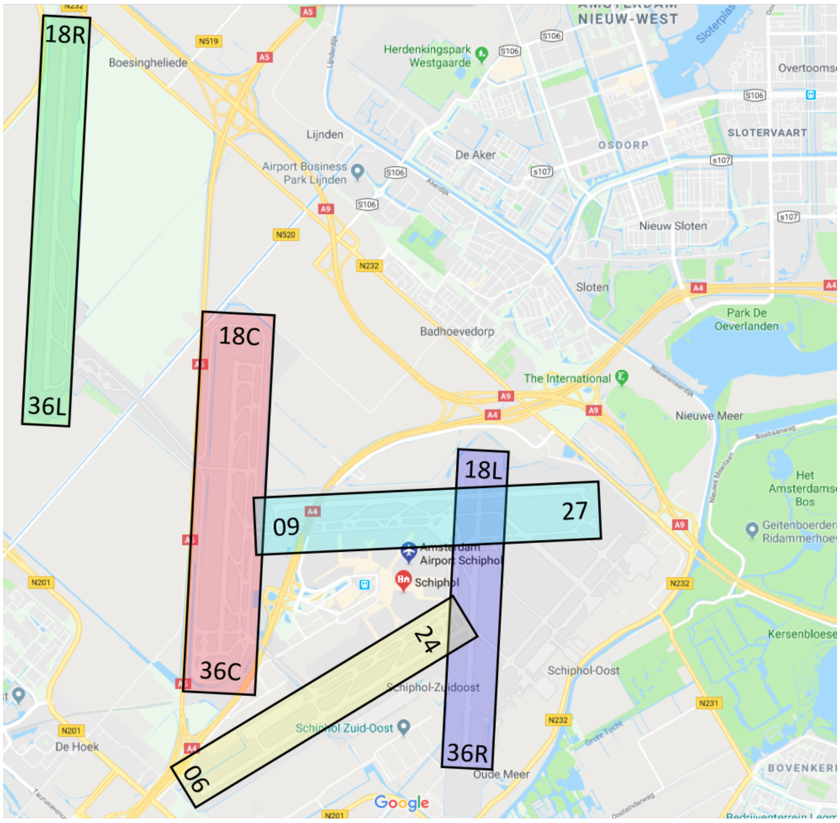

18R
36L
18C
36C
18L
36R
09
27
24
06
Boesingheliede
Herdenkingspark Westgaarde
NIEUW-WEST
Overtoomse
SLOTERVAART
Lijnden
Airport Business Park Lijnden
De Aker
OSDORP
Nieuw Sloten
Sloten
Park De Oeverlanden
Badhoevedorp
The International
Nieuwe Meer
Het Amsterdamse Bos
Geitenboerderij Ridammerhoeve
Amsterdam Airport Schiphol
Schiphol
Kersenbloese
Schiphol-Oost
Schiphol-Zuidoost
Schiphol Zuid-Oost
De Hoek
Oude Meer
BOVENKERK
Google

Figure 2.8: Runways at Schiphol Airport Adapted from [12]

2.5. Arrival Controller Best Practices

An arrival (ARR) controller's task is to guide approaching aircraft to the runway, whilst maintaining separation, as described in section 2.3. In order to do this, the ARR controller has a few control tools at his/her disposal. To achieve their goal of safe and robust control, a few best practices and requirements are applicable. To derive what ARR controller 'best practice' actually is, expert interviews were conducted with ARR controllers at the LVNL.

2.5.1. Arrival Controller Strategies

An ARR controller works in a vector area, around a runway. In figure 2.11, the ARR controller radar screen can be seen, with the vector area highlighted in white. The FDR feeds aircraft from the initial approach fixes, also highlighted, to the vector area, using predefined approach routes as guidelines (highlighted in green). Officially, the FDR transfers an aircraft to the ARR controller at the border of the vector area; in reality this may differ based on when the FDR deems transfer to be safe and/or necessary.

To control aircraft, the ARR controller has a radio communication channel (Radio Telephony, R/T) with all aircraft on his/her frequency. The controller can select aircraft in his vector area, and communicate voice commands to the aircraft. Possible voice commands are speed, altitude and heading commands. Here, speed is given to be the aircraft Indicated Airspeed (IAS), in kts, altitude is given to be the altitude measured from the airport to the aircraft (in Flight Levels, 100 ft per FL) and heading is measured in degrees, due north being a heading of 360 degrees, measured clockwise. Only minor speed reductions not exceeding ± 20 kts IAS should be used for aircraft on final approach.

To obtain optimum separation, the ARR can use a few strategies. An obvious way to reduce or increase aircraft separation is reducing or increasing speed. However, since the ARR controller operates in the last phases of flight, where the aircraft needs to lose its kinetic energy, speed increase instructions should be avoided, unless an emergency situation occurs.

Next to speed instructions, the timing of aircraft descent can play a role in aircraft separation. This is due to the differences in Indicated Airspeed (IAS), which is the airspeed as displayed in the aircraft, True Airspeed (TAS), which is the airspeed relative to the air at the height of the aircraft, and Groundspeed (GS), which is the speed of the aircraft's projection on the ground. For more details on the difference between IAS and GS, see appendix B. Since the density of air is lower at higher altitudes (at least in the region that commercial aircraft fly), an aircraft at an altitude of 4,000 ft with an IAS of 220 kts will have a higher IAS than an aircraft flying with that same IAS at an altitude of 2,000 ft. By adding wind at the altitude of the aircraft, the GS is obtained. Thus, an APP controller can decide to delay the descent of a following aircraft is he/she wants to reduce separation. The difference between the TAS and the GS is caused by wind. The GS vector is simply the TAS vector with the wind speed vector added to it. Thus, a strong wind can significantly influence the ground tracks of the aircraft. This is the reason why TBS is beneficial because this is where the difference in landing rate comes from when landing aircraft in strong headwind conditions.

Another way to change aircraft separation is to alter the course of one of the aircraft. This can be done in multiple ways. Two frequently used maneuvers are dog-legging and tromboning.

'Dog-legging' is a maneuver where the following of two aircraft, heading towards the same waypoint, is sent via another point such that the distance the aircraft needs to travel is increased. This will result in the rerouted aircraft arriving at the waypoint at a delay, increasing separation. This is illustrated in figure 2.9. Here, aircraft 1 and 2 are heading for point A. If no action is taken, the aircraft will collide. Aircraft 2 is sent via waypoint B to achieve sufficient separation when the aircraft arrives at point A.

Another maneuver that can be used is 'tromboning'. This maneuver is specific to the approach situation because it can only be executed if the future path of an aircraft involves a heading change of 180 degrees. This is often the case for approaching aircraft in the vector area, as will be explained in section 2.5.2. The tromboning maneuver is based on timing the 180 degree turn of an aircraft. If two aircraft need to make a 180 degree turn, and the following aircraft turns before the point where the leading aircraft turns, separation is reduced. Similarly, if the following aircraft's turn is initiated at a point after the place where the leading aircraft started its turn, separation is increased. This is illustrated in figure 2.10. In this figure, two aircraft which will land on a runway in south direction, are flying towards the north. To reduce separation to optimise throughput, the following aircraft's turn is initiated at a point before the point where the leading aircraft initiates its turn. The circles around the aircraft represent the minimum separation, the green line on the right represents the localizer and glideslope location for the runway. In this line, every 2 NM is depicted with a small crossing line.

2.5.2. Nominal Approach Trajectory

Within the vector area for runway 18R, aircraft can arrive from all three IAFs; RIVER, ARTIP and SUGOL. The radar screen for an ARR controller can be found in figure 2.11. As one can see, from the threshold of runway 18R, depicted by a yellow dot, a green dotted line goes upwards. This is the visualisation of the localizer, part of the Instrument Landing System (ILS). The space between each dot represents 2 NM. This is currently the only tool ARR controllers have to estimate distance between aircraft. In the vector area, the aircraft arriving from either three IAFs is guided onto this ILS by the ARR controller. The FDR feeds the aircraft via the routes depicted in green. These are the ideal routes; most of the time the FDR will deviate from this nominal path because of traffic or weather conditions.

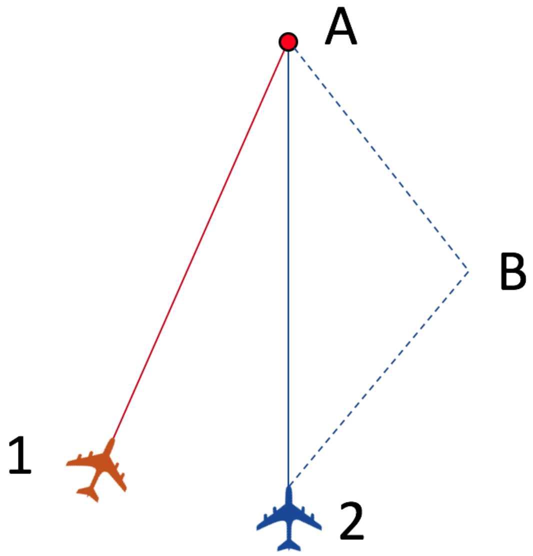


Figure 2.9: Illustration of increasing aircraft separation using a dog-leg maneuver

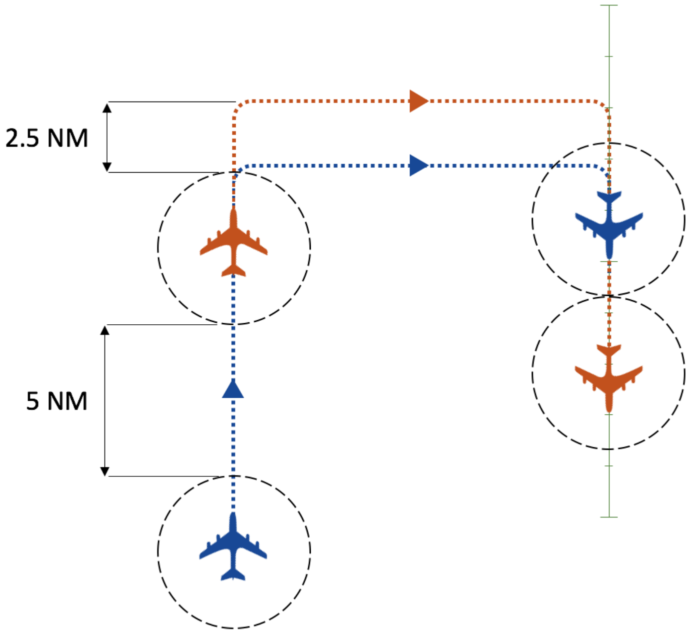


Figure 2.10: Illustration of reducing aircraft separation using a tromboning maneuver

In the process of guiding aircraft to this ILS, and eventually to the runway threshold, ARR controllers follow a certain nominal path. This path is generally accepted as being the 'best practice' by the LVNL; this is what ATC students learn to do. Please note that this path is the 'ideal' path; in reality the track of an aircraft is heavily reliant on the ARR controller in command, the pilots of the planes, the weather and traffic demands. In reality, the ideal path will never be flown exactly. However, if one must take an average of the paths taken by aircraft, this would be it. It is interesting to study this path for trajectory prediction purposes. The nominal path contains more than just waypoints; it also dictates variables like speed and altitude at certain parts of the trajectory. Below a description of an ideal trajectory is given. Please note that this description is for an aircraft arriving from the RIVER IAF. However, this ideal route can also be applied to aircraft arriving from SUGOL or ARTIP.

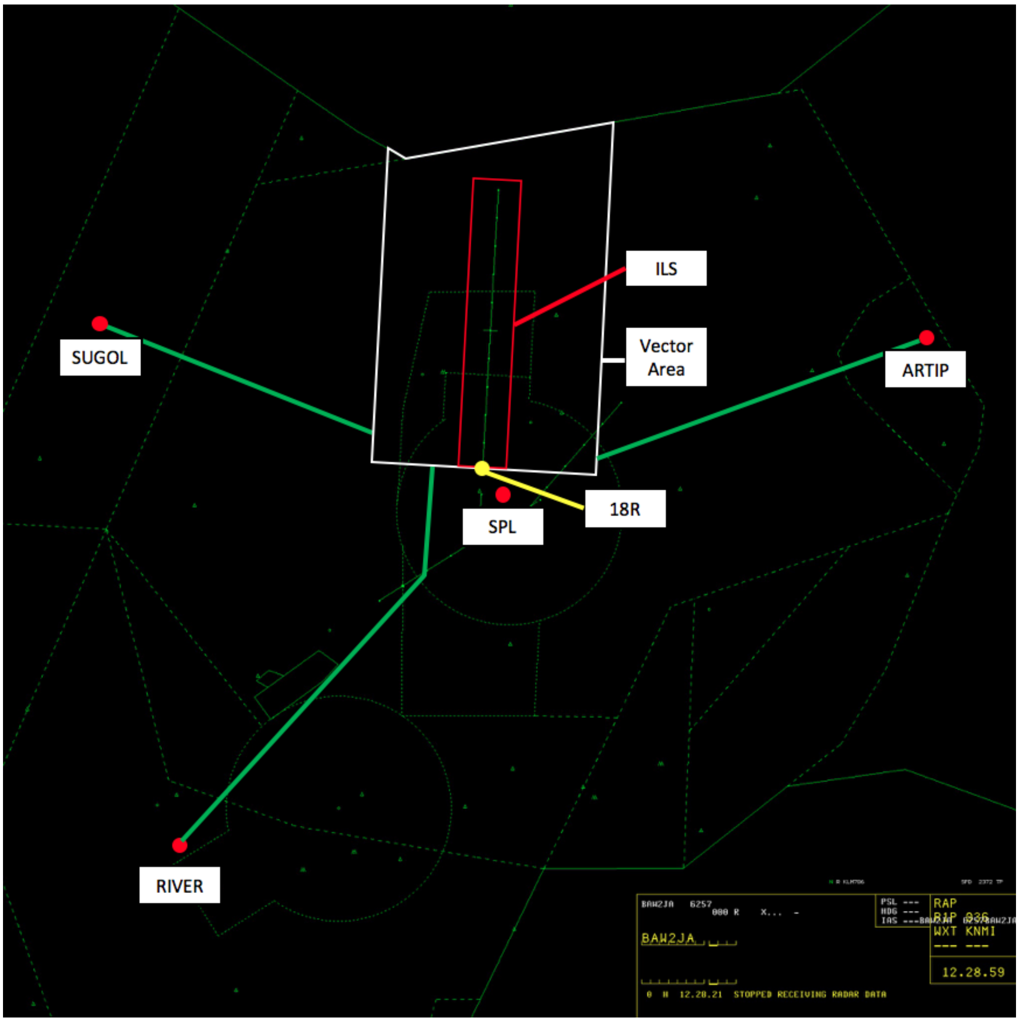


Figure 2.11: Screenshot of APP controller radar screen with IAFs and vector area for 18R

An schematic illustration of a nominal path for an aircraft arriving from the RIVER IAF on Schiphol runway 18R can be found in figure 2.12. In the schematic, one can observe 7 ’legs’ of the arrival trajectory.

In general, the nominal path begins with the downwind leg, leg 1 in figure 2.12. The downwind leg is ideally parallel to the ILS at ± 5 NM from the ILS. The downwind leg is the phase of flight where the aircraft travels to the point where it will Turn To Base (TTB); the aircraft will turn towards the ILS with a heading angle of 90 degrees w.r.t. the ILS. This happens at about 10 NM of the ILS. The TTB leg is leg 2 in the picture. While for an ideal flight the TTB point lies at 10 NM of the ILS, the ARR controllers can alter the point where an aircraft turns to base to increase or decrease separation as described in Section 2.5.1. After the aircraft has turned to base, it will travel towards the ILS with a heading angle of 90 degrees w.r.t. the ILS. At some point, the controller will steer the aircraft into the ILS with an angle of 30 degrees. This is done such that the aircraft will intercept the localizer on 8 NM, measured from the runway threshold. The controller will also clear the aircraft to intercept the localizer at this moment. This turn is depicted as leg 4 in the schematic, leg 5 represents the travel path towards the ILS. At some point, the aircraft will intercept the ILS (automated) and turn into the localizer, leg 6. The aircraft has then intercepted the ILS and travels towards the runway, leg 7.

The red numbers in figure 2.12 depict the ideal speed profile of an aircraft arriving from the RIVER IAF. Three requirements are given. At the beginning of the TTB, the aircraft must not fly faster than 220 kts. At the beginning of leg 4 (turn into ILS), the speed must not be higher than 180 kts. At 4 NM before the runway threshold, the speed must be 160 kts. From this point onwards the pilot may decide the aircraft speed for safe landing.

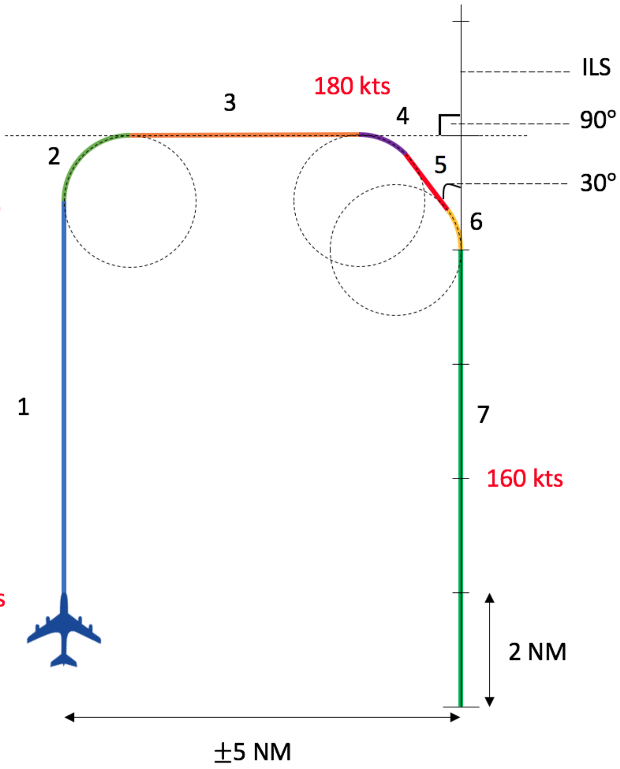


Figure 2.12: Schematic depiction of an ideal trajectory for an aircraft arriving from RIVER at Schiphol runway 18R, including ideal speed profile in red

This point is called 4DME in literature, where DME stands for Distance Measuring Equipment. At the beginning of the downwind leg, the aircraft ideally arrives with a speed of about 250 kts.

The altitude of the aircraft has as single limitation that it needs to be 2,000 ft at the interception of the localizer. Depending on traffic on other runways, the aircraft altitude will lie somewhere between 4,000 and 7,000 ft when it enters the sector. When the aircraft catches the glideslope, a 3 degree glideslope angle is automatically followed.

To compare the ideal nominal path with actual flight tracks, the ground tracks of all aircraft landing on Schiphol runway 18R from 16:00 - 17:00 on 29-01-2017 are plotted in figure 2.13. In this figure, the distances are measured in nautical miles from the SPL beacon, depicted in figure 2.11. The runway threshold of runway 18R is at the red dot. The IAFs are visible because ground tracks of aircraft cross frequently at this point and are marked in red. As one can see, the actual path of aircraft arriving on 18R differs from the nominal path. During this hour of landing aircraft all aircraft are neatly on the localizer before the 8 NM mark. The ARR controller has used the tromboning maneuver frequently; aircraft caught the localizer at a range of 8 - 15 NM. This illustrates that the ideal path is just an imaginary track in the mind of the controller, and that tromboning is an important tool for ARR controllers.

2.5.3. Arrival Controller Limitations

Next to tools the ARR controller may use and the ideal path, altitude and speed profiles he or she must try to follow, there are also some limitations that the ARR controller must consider whilst steering the aircraft.

Firstly, the ARR controller must consider that the aircraft are in their final phase of flight. As such, the aircraft must lose its potential and kinetic energy (descend and decelerate respectively). Therefore it is not good practice to give speed or altitude commands that increase the potential/kinetic energy of the aircraft. This translates to the ARR controller only being able to give speed decrease and altitude decrease instructions to aircraft. Next to only being able to giving these instructions, high rates of descent and speed reduction should be avoided.

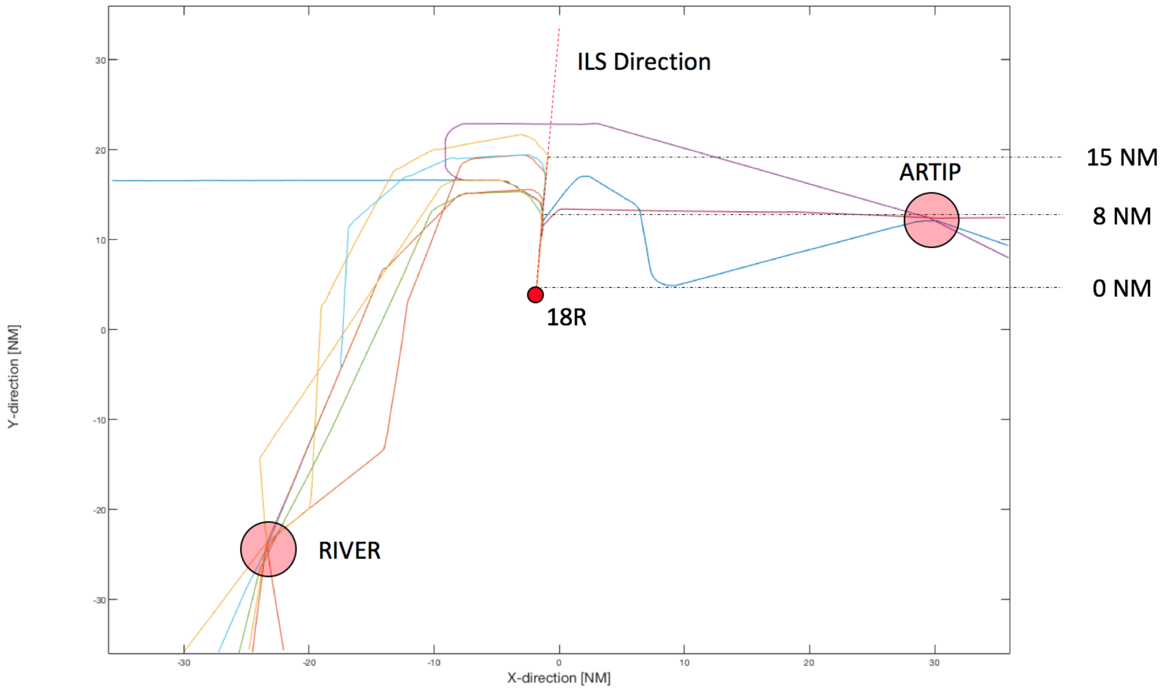


Figure 2.13: Ground tracks of all aircraft landing on Schiphol runway 18R from 16:00 - 17:00 on 29-01-2017

In addition to this, aircraft can not lose speed whilst descending without applying speed brakes. Applying speed brakes is inefficient for fuel consumption and pilots often resist speed reduction commands whilst descending. Therefore, the giving of speed reduction commands whilst in descent must be avoided.

Thirdly, ARR controllers must strive to let aircraft approach the glideslope from under the glideslope. This is because a glideslope catch from above the glideslope is more complex and results in more workload for the pilot. Since the crash of the Turkish Airlines flight on 25-02-2009 near Schiphol airport, for which a catch-glideslope-from-above maneuver was responsible in part [13], it is prohibited to let aircraft catch the glideslope from above.

Lastly, traffic at other runways dictates certain limitations. For instance, if runways 18R and 18C are simultaneously in operation, aircraft that land on 18C must keep an altitude of 4,000 ft before the aircraft catches the localizer. Similarly, aircraft landing on 18R may not approach with an altitude above 2,000 in order to keep both aircraft streams safely vertically separated.

3

Time-Based Separation

In this chapter, the concept of TBS will be further evaluated. In Section 3.1, the working principle of the concept will be explained and some advantages and disadvantages given. In Section 3.2, this situation on Heathrow Airport, where TBS is already implemented, will be analysed.

3.1. The Time-Based Separation Concept

The concept of TBS will first be explained thoroughly. After this explanation, the advantages and disadvantages will be discussed.

3.1.1. Explanation of the Time-Based Separation Concept

The current standard for ATC works with Distance-Based Separation (DBS). With With DBS, aircraft on final approach are vectored such that the distance between a lead and a following aircraft is never less than a constant, predefined value. Since aircraft are slowing down during final approach, this causes the distance between lead and following aircraft to get smaller during final approach, reaching a minimum when the lead aircraft touches down. This is known as the compression effect [1].

The minimum distance between aircraft depends on the combination of the lead and following aircraft. In general, the bigger the lead and the smaller the following aircraft, the larger the minimum separation distance between these aircraft. The DBS situation for light headwinds can be seen in figure 3.1.

A flaw of the DBS system is revealed when strong headwinds occur during final approach. When an aircraft flies in strong headwinds, its groundspeed is significantly reduced, see Appendix B. Since the aircraft still needs to maintain a specified distance from the lead aircraft, a reduced groundspeed will result in longer intervals between the landing of aircraft, and thus in a reduced landing throughput. For airports operating near full capacity, this influence of headwind is one of the major causes of delays and cancellations [1]. The strong headwind DBS problem is illustrated in figure 3.2.

From an internal investigation performed by To70 aviation consultants, it became clear that this reduced landing rate due to headwinds is one of the major causes of delays and cancellations at Schiphol Airport as well. On a stormy day, landing capacity can be reduced by as much as 85 aircraft per day [2].

This flaw of DBS can be mitigated by using Time-Based Separation (TBS). When using TBS, aircraft on final approach are no longer separated by a minimum distance, but a minimum time. Distance minima for low headwind conditions (± 5-7 kts) are converted to time intervals equivalent to the time it would take the aircraft to traverse this distance [1]. Since this is a function of groundspeed, and since groundspeed is a function of the headwind, these time interval minima will become smaller when headwind gets bigger. This way, the landing rate is theoretically the same with strong headwind as with low headwind conditions. This situation is illustrated in figure 3.3.

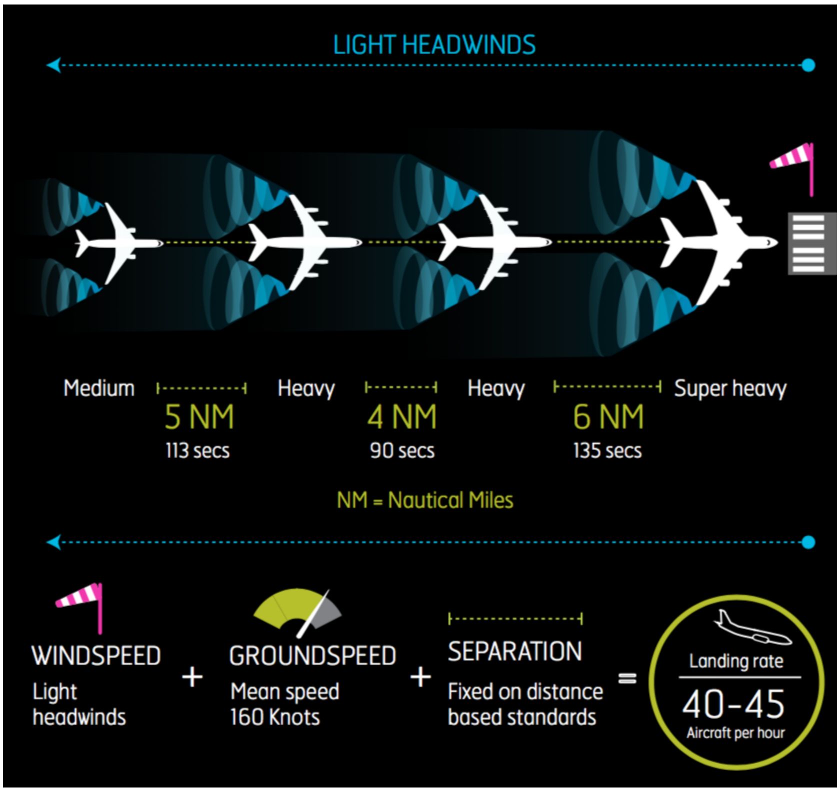

LIGHT HEADWINDS
Medium
5 NM
113 secs
Heavy
4 NM
90 secs
Heavy
6 NM
135 secs
Super heavy
NM = Nautical Miles
WINDSPEED
Light headwinds
+
GROUNDSPEED
Mean speed 160 Knots
+
SEPARATION
Fixed on distance based standards
=
Landing rate
40-45
Aircraft per hour

Figure 3.1: Aircraft Distance-Based Separation in light headwind conditions. Values added for illustration purposes only [14]

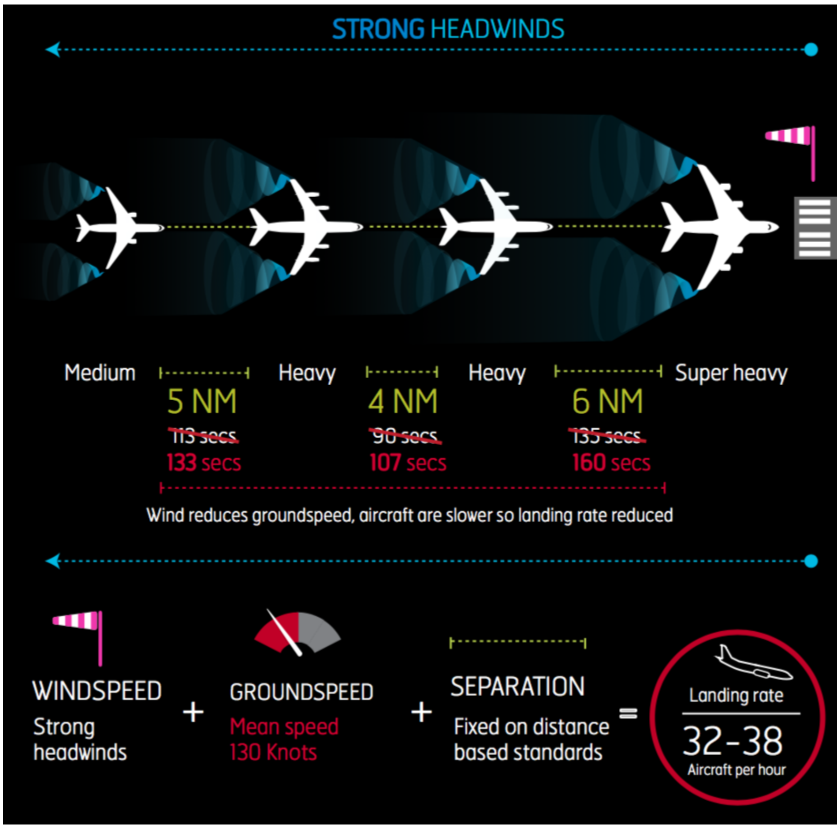

STRONG HEADWINDS
Medium
5 NM
~~113 secs~~
133 secs
Heavy
4 NM
~~90 secs~~
107 secs
Heavy
6 NM
~~135 secs~~
160 secs
Super heavy
Wind reduces groundspeed, aircraft are slower so landing rate reduced
WINDSPEED
Strong headwinds
+
GROUNDSPEED
Mean speed 130 Knots
+
SEPARATION
Fixed on distance based standards
=
Landing rate
32-38
Aircraft per hour

Figure 3.2: Aircraft Distance-Based Separation in strong headwind conditions. Values added for illustration purposes only [14]

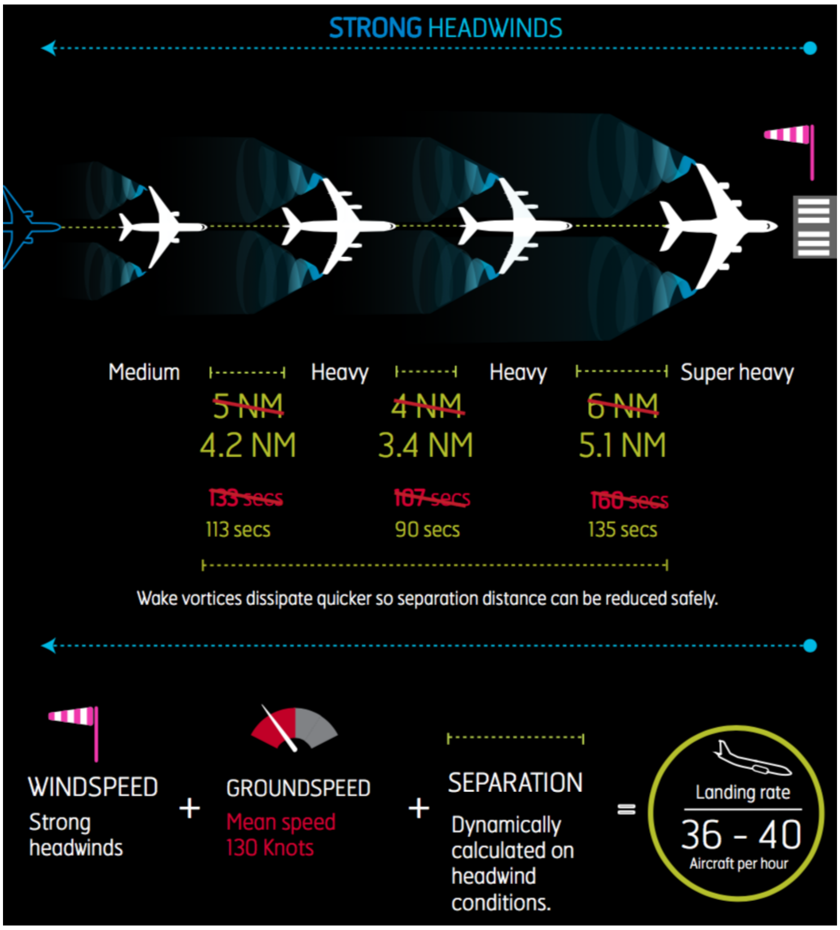

STRONG HEADWINDS
Medium
Heavy
Heavy
Super heavy
~~5 NM~~
4.2 NM
~~4 NM~~
3.4 NM
~~6 NM~~
5.1 NM
~~133 secs~~
113 secs
~~107 secs~~
90 secs
~~160 secs~~
135 secs
Wake vortices dissipate quicker so separation distance can be reduced safely.
WINDSPEED
Strong headwinds
+
GROUNDSPEED
Mean speed 130 Knots
+
SEPARATION
Dynamically calculated on headwind conditions.
=
Landing rate
36 - 40
Aircraft per hour

Figure 3.3: Aircraft Time-Based Separation in strong headwind conditions. Values added for illustration purposes only [14]

Research into the TBS concept has been done by Eurocontrol, a company which aims to support European aviation, and the National Air Traffic Service (NATS), a company providing air traffic services in the U.K. The NATS in cooperation with Heathrow Airport, has realised the first operational TBS based system at Heathrow airport. For this system, a HMI for approach controllers was developed. The situation at Heathrow airport and the HMI will be described in section 3.2.

3.1.2. Advantages and Disadvantages of Time-Based Separation

The advantages of TBS are fairly obvious. Using TBS, delays can be reduced in strong headwind conditions. At Heathrow airport, in 2017, the NATS reported a 62% reduction in wind-related delays with up to 44 movements a day recovered [3]. This is not all due to the implementation of TBS; the implementation of RECAT-EU distance separation minima can itself boost runway throughput by 5-10% [4]. However, the gains of implementing TBS are still significant.

However, there might also be some negative effects of the implementation of TBS. For one, using TB is only beneficial when the separation between aircraft is monitored and corrected precisely. This might result in more corrections per aircraft, which means more fuel used due to frequent small changes in speed/altitude/direction.

Next to this, the increased runway throughput might encourage airport operators to accept more flights in the sector. This will result in busier sectors, in the TMA but also in the CTA and UTA. Busier sectors will result in more workload for the already very demanding ATCo jobs.

Finally, The need to control aircraft more precisely could result in more workload for the ARR controller. With DBS, the controller would 'eyeball' the separation; once separated the aircraft would stay separated enough since the controller would know how much separation to give aircraft approximately such that the separation at the runway threshold would be optimal. When implementing TBS, the control needs to be more actively regulated at the final approach phase; this might result in an increase in controller interventions and thus workload.

3.2. Time-Based Separation at Heathrow Airport

Since the spring of 2015, TBS is operational at Heathrow Airport. For Heathrow Airport, a final approach HMI was developed to facilitate TBS operations. Not much is known about the HMI, but from a few short informational movies published by the NATS it is possible to get a feel of the Heathrow way of TBS implementation.

In figure 3.4, one can observe the situation as used at Heathrow Airport. In the movie, it is explained that Heathrow uses an Optimised Runway Delivery (ORD) indicator behind a TBS separation indicator. The distance for the ORD indicator is equal to the distance of the compression effect, i.e., the distance the follower aircraft will gain on the lead aircraft when the lead aircraft slows down after the 4DME point (the point where pilots may reduce speed themselves for landing, 4 NM before runway threshold). This ORD distance will thus decrease to zero when the lead aircraft passes the 4DME point, as is illustrated in figure 3.4.

In this figure, 3 consecutive phases of flight are depicted. In the first phase, both aircraft are before the 4DME point. The TBS distance is equal to the distance as dictated by the RECAT-EU TBS separation protocol. When the first aircraft passes the 4DME point, the ORD distance begins to decrease. In the last phase, when Aircraft 1 touches down, the ORD distance is reduced to zero.

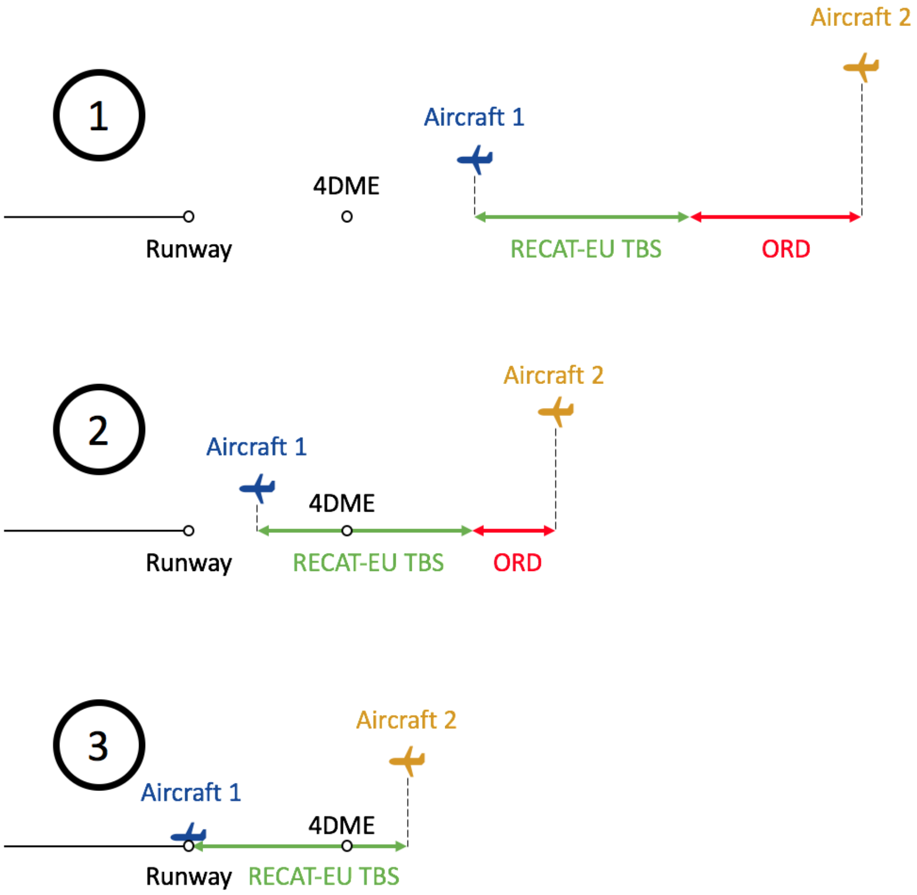


Figure 3.4: Depiction of the Optimised Runway Delivery market, used at Heathrow Airport, for 3 consecutive phases of flight: 1. both aircraft before 4DME, 2. Aircraft 1 after DME, 3. Aircraft 1 on runway

Next to the ORD indicator, a Runway Occupancy Time (ROT) indicator is used as depicted in figure 3.5. This indicator is used when the time separation between aircraft is such that the lead aircraft needs more time to safely vacate the runway before the follower aircraft lands. This is the case when a larger aircraft lands behind a light aircraft; using RECAT-EU TBS margins it can be that the time separation between the two is smaller than the time the light aircraft needs to vacate the runway. It is therefore an indicator which extends the TBS separation distance on the HMI.

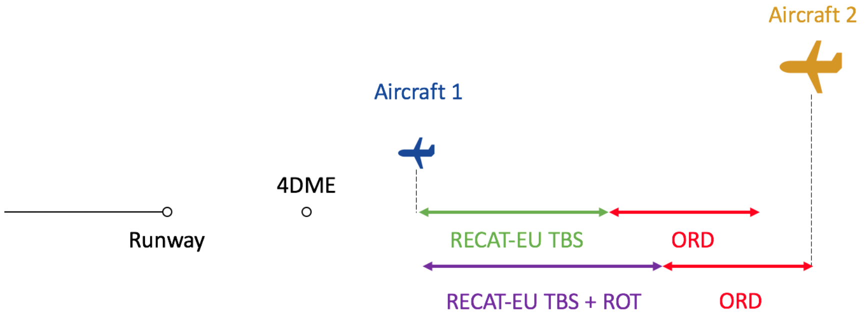
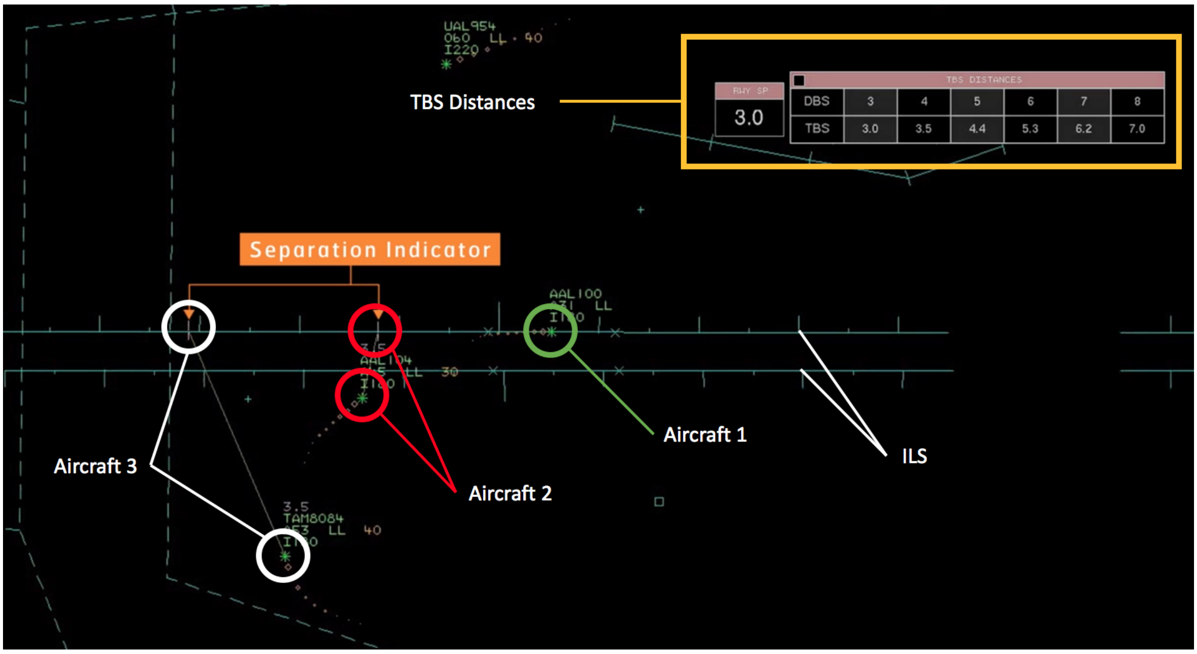


Figure 3.5: Depiction of the use of the ROT marker, as used by Heathrow Airport

In figure 3.6 a screenshot of a movie depicting the HMI for Heathrow TBS operations is depicted. In this screenshot, we see two ILS indicators above each other. Aircraft 1 has already caught the ILS, its separation marker is the marker which Aircraft 2 should not cross. This marker, along with Aircraft 2, is marked in the red circle. In white are Aircraft 3 and its separation marker, created by Aircraft 2. From this screenshot it becomes apparent that Heathrow uses separation indicators projected on a straight line from the runway. This is interesting since separation markers are already plotted for aircraft that are not yet on the ILS (Aircraft 2 is not on ILS and already created a separation market). From this information, it becomes apparent that Heathrow uses some kind of ’ghosting’ algorithm to estimate location on the ILS of aircraft, even though they are not yet physically on the ILS. The concept of ghosting will be treated in Section 5.1.1.

In this figure, there is no clear difference between the ORD and ROT indicator. In the top right corner, a table is visible. This is a table relating certain DBS margins to TBS margins. This is what is called ’procedural’ TBS; for the current headwind conditions the distance separations are calculated and displayed. This way, the controller can work as he/she is used to; separating based on distance. The difference is that the distances the controller uses differ based on wind conditions.


UAL954
060 LL 40
I220
TBS Distances
Separation Indicator
AAL100
LL
3.5
AAL104
LL 30
3.5
TAM8084
LL 40
Aircraft 1
Aircraft 2
Aircraft 3
ILS

RWY SP

3.0

TBS DISTANCES

DBS	3	4	5	6	7	8
TBS	3.0	3.5	4.4	5.3	6.2	7.0

Figure 3.6: Screenshot of informational movie by NATS on TBS at Heathrow, depicting the HMI used for TBS

4

Cognitive Work Analysis

In this chapter, a Cognitive Work Analysis (CWA) will be performed, following the principles of Ecological Interface Design (EID). In section 4.1, the EID principle will be explained.

As opposed to regular user- and technology-centred approaches, the EID framework starts by analysing the work domain for which the display is developed [15]. Next to analysing the work domain, the EID method also considers a control task analysis (Section 4.3), strategies analysis (treated in Section 4.4), social organisation & cooperation analysis (treated in Section 4.5) and a worker competencies analysis (treated in Section 4.6). Each of these analyses will be executed for this research in this chapter.

4.1. On Ecological Interface Design Concept

4.1.1. The Ecological Interface Design Concept

Due to recent technical developments, such as sensors and automation, the role of the human controller in complex control systems has changed. However, it has become evident that humans can not be discarded as part of complex control systems just yet; the ability of humans to deal with unanticipated irregular situations still can not be recreated by fully automated systems, especially in complex sociotechnical environments such as air traffic control. Therefore, human-machine systems must strive towards supporting the human controller, instead of replacing him [15]. In 1989, Rasmussen and Vincente coined the concept of Ecological Interface Design (EID) [16]. EID is designed based on a model of human behaviour; the Skill-Rules-Knowledge Taxonomy.

The Skill-Rules-Knowledge Taxonomy, coined in 1983 by Rasmussen is a framework for analysing human behaviour [17]. In this taxonomy, three types of behaviour were identified; Skill-, Rule-, and Knowledge-based behaviour (SRK). Skill based behaviour is behaviour where the actor acts based on intuition or training. It is almost like a reflex, not a lot of cognitive effort is being used to perform this behaviour. As such, responses based on skill based behaviour are quick and perceived as 'easy'. Rule based behaviour is when an actor acts on rules he/she may have learned. This includes recognising a situation, associating the task relevant for the situation and then, using a stored rule, executing that task. This behaviour already requires more cognitive effort. Skill based behaviour is when the actor actively has to think of an approach, based on the goals he/she may have. As such, it requires the highest cognitive effort. Skill based behaviour is performed when situations occur that are out of the ordinary. A flowchart of human behaviour in the SRK taxonomy is given in figure 4.1.

The EID design framework encourages a constraint-based approach and aims to visualise the space of possibilities (governed by laws of physics) in a way that supports this skill-, rule- and knowledge-based behaviour. In the framework the work domain and control task of the controller is first analysed thoroughly. Then, by analysing the behaviour of the controller and strategies that can be used, a display can be designed which shows the realm of possibilities in the work domain, in a way that naturally represents the work domain to the controller. For instance, the radar screen used by an ATCo represents the controlled area as seen from above (bird's-eye view), which is a natural way to present the work domain to the controller. Over this work

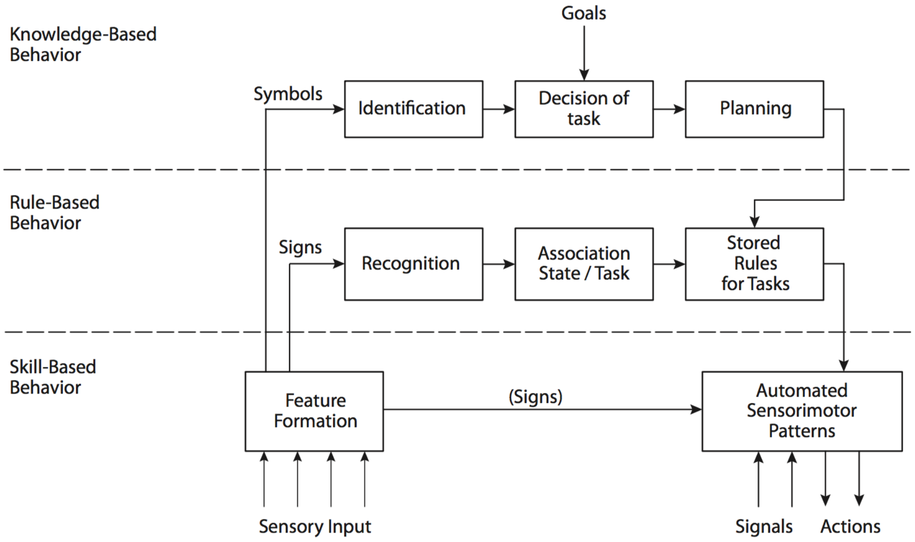

Knowledge-Based Behavior
Goals
Symbols
Identification
Decision of task
Planning
Rule-Based Behavior
Signs
Recognition
Association State / Task
Stored Rules for Tasks
Skill-Based Behavior
Feature Formation
(Signs)
Automated Sensorimotor Patterns
Sensory Input
Signals
Actions

Figure 4.1: The skills-rules-knowledge taxonomy, a tool for analysing worker competencies [18]

domain, the locomotion space of an aircraft can be visualised, showing where this aircraft can go and where it can not go, based such things as the laws of physics, aircraft performance and sector regulations. Such a display moves Knowledge-Based Behaviour (KBB) and Rule-Based Behaviour (RBB) to RBB and Skill-Based Behaviour (SBB); the ATCo can directly observe the realm of possibilities instead of having to infer the possibilities from data.

In recent research, EID has been applied to vehicle locomotion control. The EID framework is especially useful for ATC, since controllers must focus on the current system and respond quickly, whilst also planning their next move. The problem is a multiloop problem; the ATCo must act in the short-, medium-, and long-term. Such a multiloop problem can be given as a set of nested loops; for a pilot this problem is given in figure 4.2. When applying EID, one or more of these nested loops is essentially closed, thereby freeing cognitive space of the controller for other tasks [19].

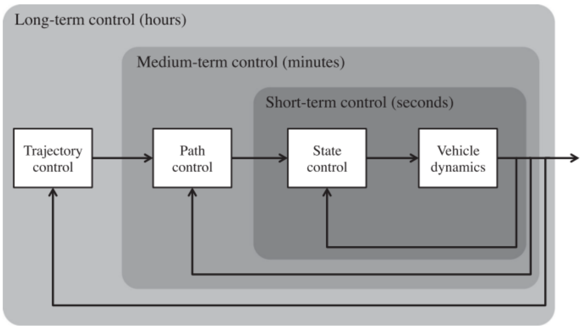

Long-term control (hours)
Medium-term control (minutes)
Short-term control (seconds)
Trajectory control
Path control
State control
Vehicle dynamics

Figure 4.2: Short-, Medium-, and Long-term control for a pilot [19]

4.1.2. Advantages and Disadvantages of Ecological Interface Design

The main advantage of EID is that the resulting displays show the controller what constraints the system has, whilst avoiding taking control away from the human. This way, the human controller is still ’in control’ and is assisted in picking the best possible control strategies. This is because the method of EID promotes the creation of displays that are transparent; the controller knows what constraints the system has and is free to pick any strategy that lies within the realm of possibilities. As stated before, humans are good at providing creative solutions for unanticipated irregular events so this is a huge benefit, especially for complex sociotechnical systems like ATC.

When looking at the HMI used at Heathrow airport in Section 3.2, one can observe that it is fairly basic. The display shows what the separation of each aircraft on the ILS should be, but nothing else. The controller is thus only informed about one constraint; the separation margin. No information is given on what strategies the controller may take to achieve this separation. The controller is in control but only supported in this minimum constraint; the minimum separation margin. An EID could for instance provide insight in which points the aircraft can travel to and at what time they could get there, considering the constraints of the system and those imposed by traffic. This would potentially increase system transparency and reduce controller cognitive workload.

Another advantage of EID is that displays designed using this framework may be easier accepted by industry then displays that, for instance, display advisories. Research has been done in acceptance issues and it was concluded that acceptance of newer technologies is highest if the technology follows the same mental model of the controller [20] [21]. Since EID first thoroughly analyses the work domain and tries to represent the internal model of the actor as close as possible, the acceptance of ecologically designed interfaces will be higher than an interface that shows just an advisory, which is not transparent enough for the controller to understand.

A Disadvantage of EID might be that it may result in cluttered displays, if designed improperly. Since EID attempts to show all constraints in the system a display might become cluttered if not enough assumptions are made. When no assumptions are made and all information is displayed, a display for a complex sociotechnical system will include a lot of information, especially if the display designer wants to show all possibilities at once. A solution to this problem might be to introduce tools like toggle parameters, where the controller can control an extra input of the display. For instance, using the scrolling wheel on a computer mouse, a speed that is used for trajectory prediction can be selected, resulting in various display situations. This way, clutter can be avoided and more constraint situations can be showed to the controller.

4.2. Cognitive Work Domain Analysis

The work domain of an APP controller has already been extensively treated in chapter 2.
To structurally summarize the constraints of the work domain of an ARR controller on Schiphol runway 18R, an Abstraction Hierarchy (AH) was made. This hierarchy has five different levels; each of the levels contains relevant information on the work domain to be modelled. These five levels are [22]:

- Functional purpose (for which system is designed)
- Abstract function (intended process structure in terms of mass, energy and information)
- Generalised function (basic functions system must achieve)
- Physical function (characteristics of components)
- Physical form (appearance and spatial location of components)

The abstraction hierarchy structures all constraints in a psychologically relevant manner. Benefits of the AH are that it provides operators with information for dealing with unanticipated events and the representation is psychologically relevant for problem solving [22].

In each abstraction level, components and restrictions for the actor are given. For each element, the answer to the question 'why?' can be found an abstraction level above the element. The answer to the question 'how?' can be found one abstraction level below the element. As such, the elements get more abstract with each increase in level. This is illustrated in figure 4.3.

In figure 4.4, an AH for the ARR controller task can be observed. Functional purposes of the work domain of an arrival controller are to maintain safety, efficiency and productivity. These are the main goals of the task. Safety is of course the most important aspect of the task, since safety should always be the top priority in aviation. Safety in this work domain is achieved by maintaining separation in the aircraft locomotion space,

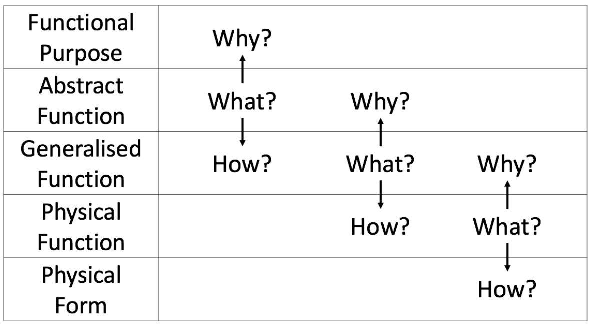


Figure 4.3: The abstraction hierarchy with the questions relative to each abstraction level, adapted from [23]

which are both abstract functions. Another abstract function is throughput, which is needed for efficiency and productivity. Generalised functions that affect the locomotion space for the work domain are:

1. (Future) Aircraft states. The states and possible future states of all aircraft in the domain affect the locomotion space of all aircraft in the vicinity.
2. Vectoring/communication. All aircraft are steered by vectoring, which is done using voice commands.
3. Sequencing. Sequences of aircraft affect where aircraft will go and when, and in what order.
4. Traffic. This affects where aircraft can go without resulting in a collision.
5. Drift. Drift, caused by wind, affects the path of an aircraft and influences affects aircraft locomotion space.
6. Aircraft performance. The performance limit of an aircraft governs what states it can reach and when and thus affects locomotion space.

These generalised functions are also linked with other abstract functions, as can be seen in figure 4.4. All these generalised functions have underlying physical functions:

1. The ILS. The ILS governs where aircraft will land and thus influences future states of aircraft.
2. Vector Area 18R. This area is where vectoring happens and thus influences the vectoring function.
3. Voice R/T. Voice commands are used to communicate with and steer the aircraft and thus influences vectoring and sequencing.
4. Flight plan. The flight plan governs the planning and routes aircraft take and thus influences future aircraft states, sequencing and traffic in general.
5. RECAT TBS separation minima. These minima govern how much distance or, in case of TBS, time aircraft should keep between each other. As such, it affects future aircraft states and traffic.
6. Scheduling demands. Scheduling demands prescribe when aircraft should ideally land. As such, it affects future aircraft states, sequencing and traffic.
7. Wind. Wind causes drift of the aircraft path and thus influences aircraft performance, drift, sequencing and future aircraft states.

Each of the physical functions has a physical form. These are the shapes, locations and statuses of the physical functions. The physical form of the ILS is its direction and location. The shape and size of vector area 18R is what makes up the vector area itself and radio is what voice R/T commands consist of. Observed aircraft state is a physical form of the flight plan. Wake vortices is a physical form of the separation minima, since this is the process governing the minimum distances/separation times. The aircraft planning governs scheduling demands and wind is comprised of its direction and speed.

Note that in figure 4.4, a few parameters are marked in green. These are the parameters that the ARR controller in the current situation can observe/has available. A way to design a display would be to aim at enabling the controller to perceive some of the parameters in the AH that are currently not available, since these are the parameters that the controller has to actively evaluate, increasing his/her workload. In other words, the controller has to use KBB to evaluate at the parameters that are not available and an EID display can help moving this KBB to RBB and even SBB. This would result in a more transparent system and a lower cognitive workload.

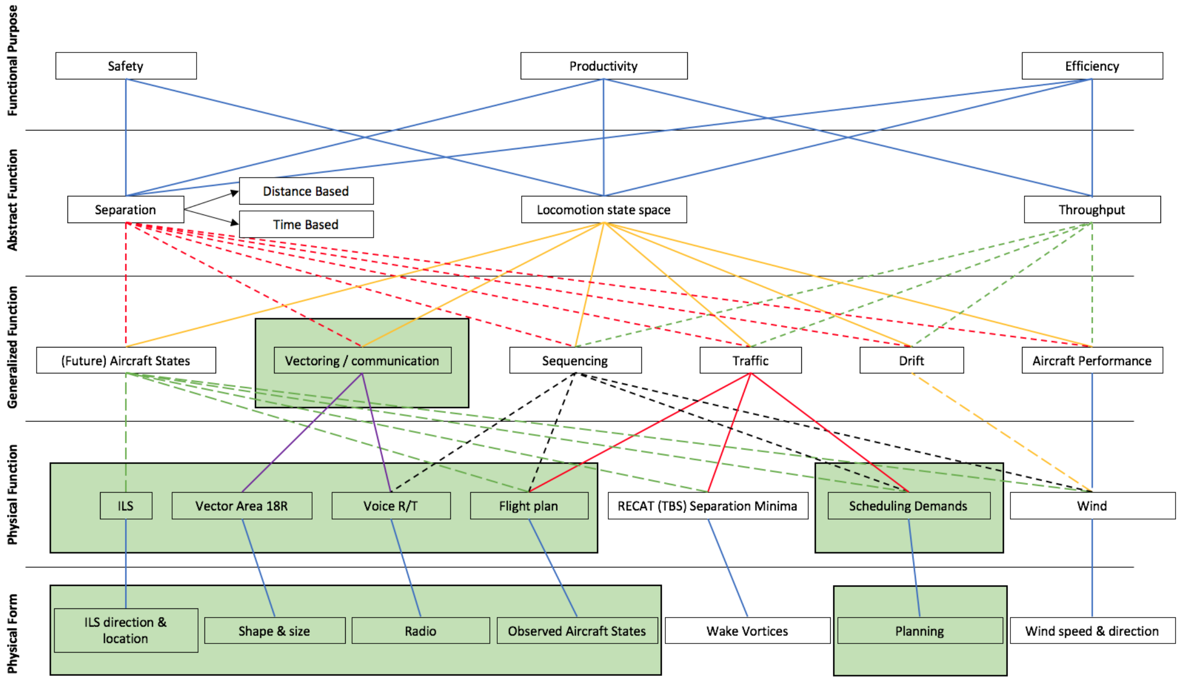

Functional Purpose
Abstract Function
Generalized Function
Physical Function
Physical Form
Safety
Productivity
Efficiency
Separation
Distance Based
Time Based
Locomotion state space
Throughput
(Future) Aircraft States
Vectoring / communication
Sequencing
Traffic
Drift
Aircraft Performance
ILS
Vector Area 18R
Voice R/T
Flight plan
RECAT (TBS) Separation Minima
Scheduling Demands
Wind
ILS direction & location
Shape & size
Radio
Observed Aircraft States
Wake Vortices
Planning
Wind speed & direction

Figure 4.4: An Abstraction Hierarchy for the arrival controller work domain

4.3. Control Task Analysis

While the work domain analysis focuses on the work domain, the control task analysis focuses on the task of the ARR controller. It aims to provide a better understanding of all the actions involved in the decision making process [24]. A tool that is frequently used for this analysis is a decision ladder (DL). In this ladder, all actions that need to be taken are given in boxes. Each action leads to a new state of knowledge, stated in circles. In general, the decision ladder follows the same abstraction levels as the AH; at the top of the ladder are the functional purposes, or general goals of the task, and on the bottom are the actual tasks themselves.

Novice controllers will traverse the DL in full; they will start at the bottom, traverse all the actions and knowledge states, to in the end reach a procedure to follow, based on rational decisions. Expert controllers will most likely show shortcuts in their internal DL. Two shortcuts are possible: a shortcut from an action to a knowledge state (shunt) and a shortcut from a knowledge state to another knowledge state (leap).

In figure 4.5 the DL for the ARR controller task is given, along with two possible expert shortcuts. The DL begins at the ’Detect need for action’ action, where the controller or the computer identifies a possible conflict. This can be a direct conflict (alert by computer, collision impending) or a future conflict detected by the controller, as well as a need for action to steer an aircraft in order for it to traverse its desired path. This action leads to the ’ALERT’ knowledge state; the controller knows that the DL must be initiated.

Next, the ’Observe information & data’ action must be taken. Here, the controller observes the state of every individual aircraft and checks important other conditions of the work domain, such as weather conditions. This results in a knowledge state 'Set of states', where the controller knows the state of every aircraft in its work domain.

Next, the controller will 'Identify the system state', which is to say he/she will combine the states of all aircraft to create an internal model of the state of the whole system. He/she combines all the states of aircraft and weather to a network of data points such that the state of the whole system can be evaluated. This reaults in the 'System state' knowledge state; an extensive model of the whole work domain and all actors in it.

The controller will then interpret this system. He/she will decide if action needs to be taken and which options are available. This results in the 'Options' knowledge state, an internal model representing possible actions that can be taken, given the system of states in the work domain.

Based on the 'Goals' knowledge state, the controller will now decide which option to choose. Goals can be, for instance, maintaining safe separation of aircraft, make this separation more efficient or to steer aircraft towards the landing strip. Note that these goals correspond to the functional purposes identified in section 4.2, figure 4.4. Based on the goal that is the most critical at that moment, the controller performs the 'Performance evaluation' action, in which he/she chooses one of the options identified earlier. This results in a 'Chosen goal' state of knowledge, the controller knows what goal should be given priority.

The 'Interpret system state' action will be performed again, this time to identify the target state, based on the chosen goals. This results in a 'Target state' knowledge state.

The target state will be transformed in a defined tast in the 'Define task' action; from this task a procedure to be taken will be identified in the 'Formulate procedure' action. This procedure will then be executed, ending the DL.

Two shortcuts are defined in this DL. First, a leap is defined from the 'System state' knowledge state to the 'Procedure' knowledge state. An expert controller can sometimes, from an internal system state, directly infer a procedure that needs to be executed. This happens for instance when two aircraft are on a collision course; based on internal rules the controller will take action such that the aircraft will not collide (rule based behaviour). A leap is defined from the 'Interpret system state' action to the 'Task' knowledge state; expert controllers will sometimes identify tasks that need to be taken directly when interpreting the system state.

Note that no leaps or shunts are defined before the 'System state' knowledge state is reached. This is because the current absence of automation in the ARR controller work domain; controllers must always arrive at an internal system state model via observation and deduction, they are not assisted in this process. The concept that will be developed in this research will try to assist the controllers early on in the decision ladder, facilitating better shortcuts in the DL, see chapter 5.

4.4. Strategies Analysis

A strategy analysis aims to investigate the ways in which the tasks can be accomplished. This is done regardless of the actor performing the task [24]. The arrival controller can give the aircraft a certain set of commands, as discussed in section 2.5.2. This limitation on possible commands and the limitations discussed in section 2.5.3 result in two possible strategies for the Arrival controller; the controller can either change the speed or the path of an aircraft. Here, altitude change is seen as a speed change for the strategy analysis since a change in altitude results in a change in groundspeed.

One of the ways to visualise the strategies one can take is using an information flow map, in which the flow of information can be observed. Such an information flow map for the ARR controller strategies can be found in figure 4.6. The 'Path Revision' and 'Speed Revision' blocks are worked out in more detail in figures 4.7 and 4.8, respectively.

Starting again by detecting a need for action in figure 4.6, the controller must then decide upon the most critical goal in the system at that moment. From this goal, the controller must identify what is the best strategy;

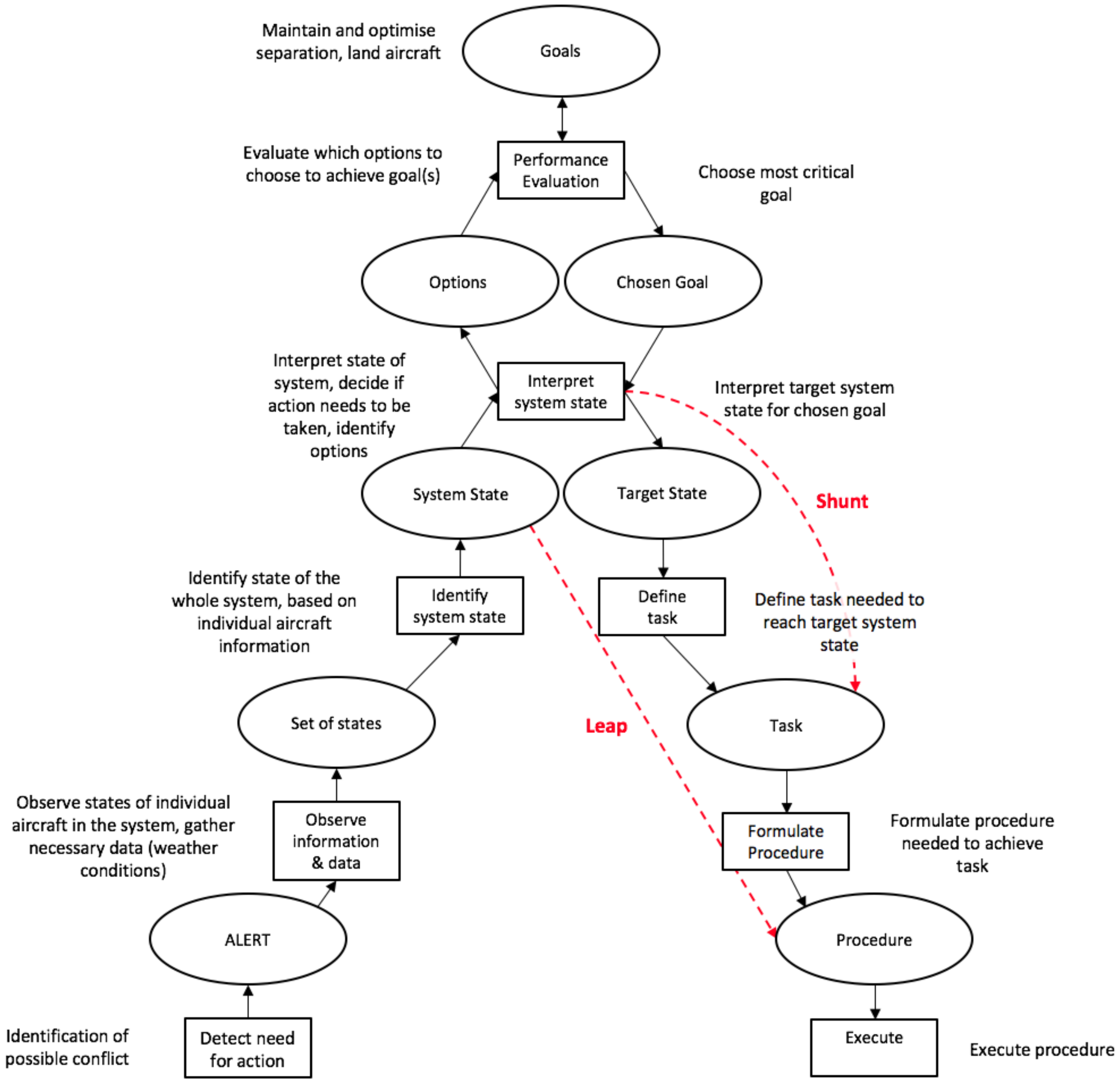

Maintain and optimise separation, land aircraft
Goals
Evaluate which options to choose to achieve goal(s)
Performance Evaluation
Choose most critical goal
Options
Chosen Goal
Interpret state of system, decide if action needs to be taken, identify options
Interpret system state
Interpret target system state for chosen goal
System State
Target State
Shunt
Identify state of the whole system, based on individual aircraft information
Identify system state
Define task
Define task needed to reach target system state
Set of states
Leap
Task
Observe states of individual aircraft in the system, gather necessary data (weather conditions)
Observe information & data
Formulate Procedure
Formulate procedure needed to achieve task
ALERT
Procedure
Identification of possible conflict
Detect need for action
Execute
Execute procedure

Figure 4.5: Decision ladder for the arrival controller task

revising the path or the speed of an aircraft. The controller does this based on his/her experience and training, unassisted by computer aides.

In figure 4.7, the path revision strategy is shown in more detail. When the need for a path revision is identified, the controller will identify which aircraft to reroute, after which he/she will decide to lengthen or shorten the path. In Section 2.5.1, the strategies to obtain a path revision, dog-legging and tromboning, were discussed. In practice, the ARR controller can pick either strategy or a combination of both. If, in this later stage, it becomes evident that another aircraft should be rerouted, an iteration may take place, selecting another aircraft to reroute. The controller does this based on data such as aircraft state, scheduling demands, traffic, wind conditions and separation minima. The outcome of this strategy is a new path for one aircraft.

The speed revision strategy is a little more complex, as can be seen in figure 4.8. First, an aircraft to revise is selected. The controller must then decide if a normal speed reduction is possible with the limitations as described in section 2.5.3. If a normal speed reduction is possible, this speed reduction is the outcome of the strategy. If this is not possible, the controller will look at changing the altitude of the aircraft to change its groundspeed w.r.t. other aircraft. If this is possible, a new descent profile is the outcome of the strategy. If a descent profile revision is not possible, another aircraft must be selected.

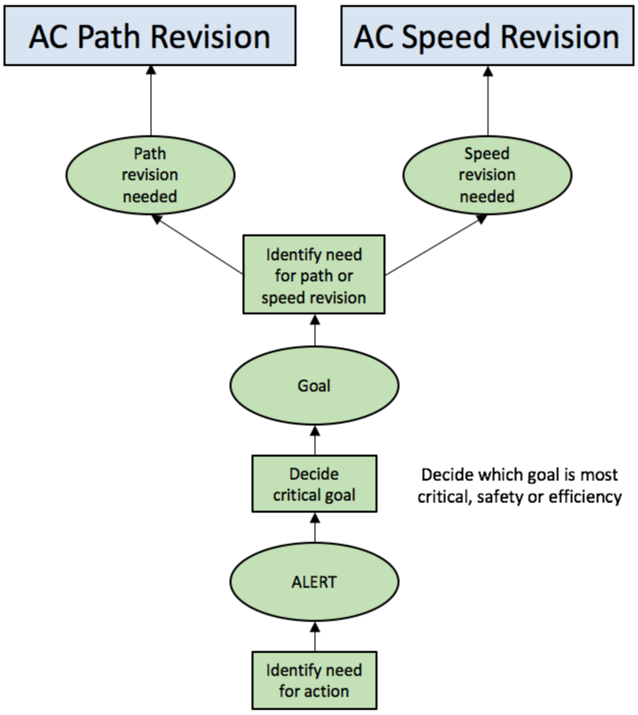

AC Path Revision
AC Speed Revision
Path revision needed
Speed revision needed
Identify need for path or speed revision
Goal
Decide critical goal
Decide which goal is most critical, safety or efficiency
ALERT
Identify need for action

Figure 4.6: Information flow map for the strategies an arrival controller can take, including actor task division: green for human, yellow for computer and blue for shared

4.5. Social Organisation and Cooperation Analysis

In this analysis, the task allocation and distribution amongst actors is investigated [24]. The ARR controller cooperates with the APP-SUP, the FDR and the TWR. However, for the task that the ARR controller has to execute (separating aircraft and steering them to the ILS), only two actors are involved; the ARR controller and the computer that assists the controller. Since the APP-SUP and FDR only influence when the ARR controller task starts, and since handing over an aircraft to the tower is done in the same way at all times, it is left out of the social organisation and cooperation analysis; these external actors do not influence the essence of the ARR task.

In figures 4.6, 4.7 and 4.8 the division between the actors is given in colours of the blocks. Here, green blocks represent items or actions of the human, yellow represents tasks or data from the computer and blue represents a shared task/responsibility.

The strategy decision is entirely in the hands of the human operator, from the identification of a need for action to the controlling action. The controller is only assisted by the computer with data such as aircraft state, traffic, weather conditions and scheduling demands. The separation minima is shared since, when using TBS, the computer must visualise some kind of marker to show where the aircraft must fly to be separated based on time. However, in a normal DBS situation, the human controller is also responsible for evaluating the separation minima.

Picking a suitable strategy is thus mostly the task of the human operator. This gives much of room for improvement; automation tools can assist on many areas in the strategy decision process, which is something this research will aim to do.

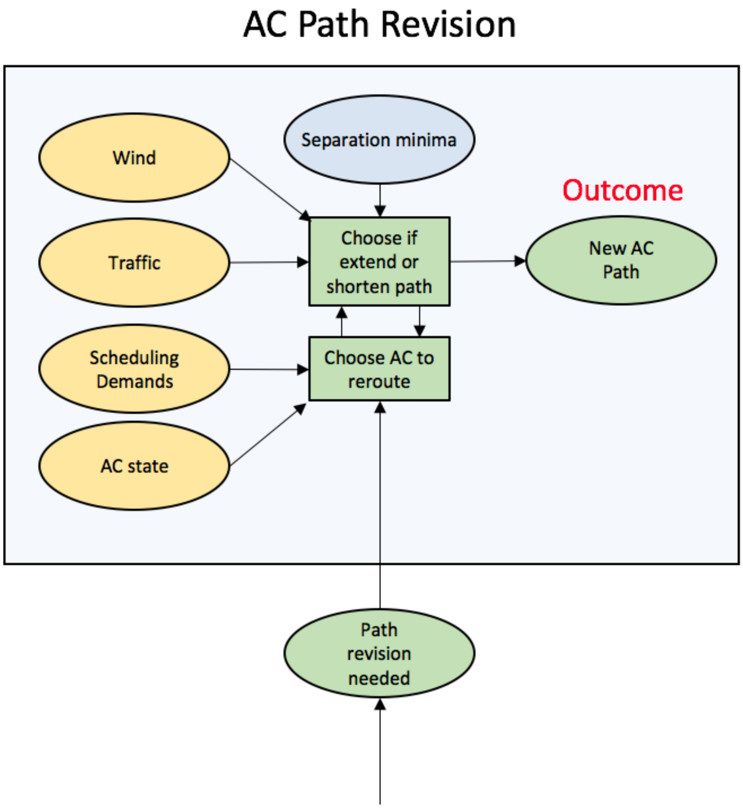

AC Path Revision
Wind
Separation minima
Outcome
Traffic
Choose if extend or shorten path
New AC Path
Scheduling Demands
Choose AC to reroute
AC state
Path revision needed

Figure 4.7: Information flow map for the path revision strategy, including actor task division: green for human, yellow for computer and blue for shared

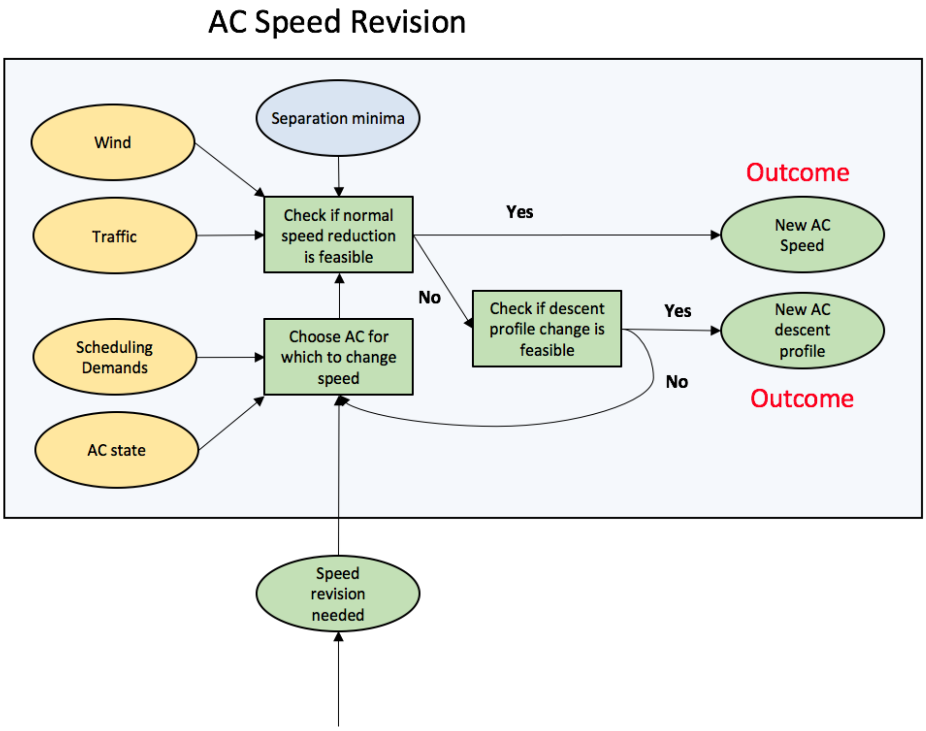
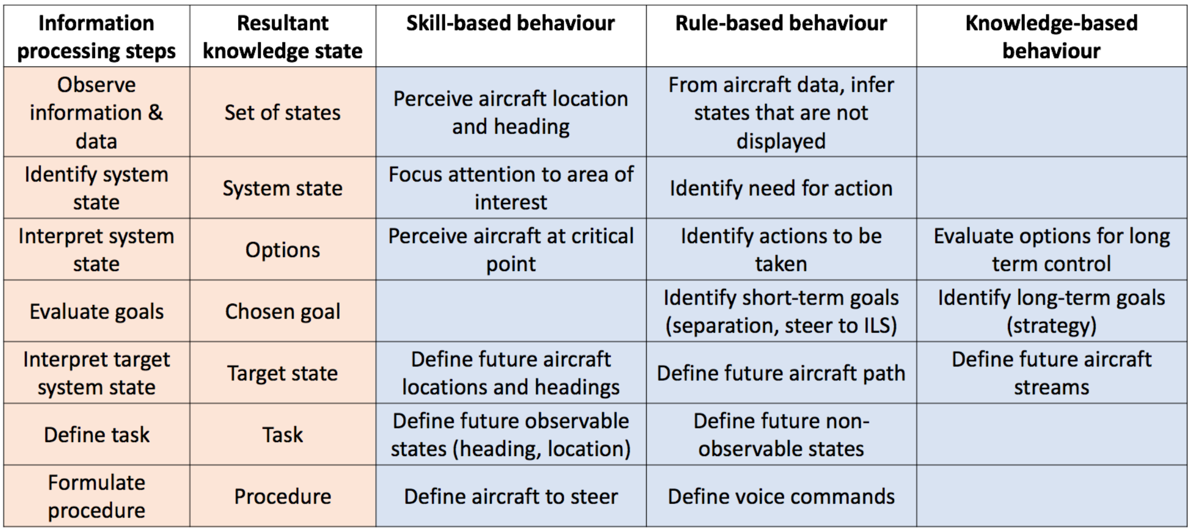

AC Speed Revision
Wind
Separation minima
Outcome
Traffic
Check if normal speed reduction is feasible
Yes
New AC Speed
No
Scheduling Demands
Choose AC for which to change speed
Check if descent profile change is feasible
Yes
New AC descent profile
No
Outcome
AC state
Speed revision needed

Figure 4.8: Information flow map for the sped revision strategy, including actor task division: green for human, yellow for computer and blue for shared

4.6. Worker Competencies Analysis

The purpose of a worker competencies analysis is to evaluate which competencies are needed by by the actors in the work domain to achieve the goals. The goal of the analysis is ’identifying psychological constraints applicable to systems design’ [24]. A framework frequently used for this analysis is the skills-rules-knowledge (SRK) taxonomy, as explained in section 4.1.

To analyse the ARR controller worker competencies, a SRK inventory can be made. This is a table where, for every action in the decision ladder, the rule, skill and knowledge based behaviours are described. A SRK inventory for the ARR controller can be found in figure 4.9.



Information processing steps	Resultant knowledge state	Skill-based behaviour	Rule-based behaviour	Knowledge-based behaviour
Observe information & data	Set of states	Perceive aircraft location and heading	From aircraft data, infer states that are not displayed	
Identify system state	System state	Focus attention to area of interest	Identify need for action	
Interpret system state	Options	Perceive aircraft at critical point	Identify actions to be taken	Evaluate options for long term control
Evaluate goals	Chosen goal		Identify short-term goals (separation, steer to ILS)	Identify long-term goals (strategy)
Interpret target system state	Target state	Define future aircraft locations and headings	Define future aircraft path	Define future aircraft streams
Define task	Task	Define future observable states (heading, location)	Define future non-observable states	
Formulate procedure	Procedure	Define aircraft to steer	Define voice commands	

Figure 4.9: A SRK taxonomy for the ARR controller

For the ’observe information and data’ action, the resulting knowledge state is ’set of states’. For an ARR controller, the location and heading are directly perceivable on the radar screen. This is thus defined as SBB, the controller needs almost no cognitive effort to execute this task, it happens naturally. Already more effort needs to be spend to infer non-observable aircraft data, such as whether the aircraft is ascending or descending, or whether it is accelerating or decelerating. This is defined as RBB. The controller solves rules like ’IF target speed is lower than current speed THEN aircraft is decelerating’. For this action, no KBB would take place since more cognitive heavy behaviour would move up the DL.

Likewise, two different types of behaviour are defined for the ’Identify system state’ action, resulting in the ’System state’ knowledge state. SBB would be where the controller focuses his/her attention towards areas of interest. This can be an area with a group of aircraft, which would possibly indicate the need for separation, or an area where aircraft enter the sector, which would indicate the need for new instructions to these new aircraft. RBB would be when the controller identifies the need for action, based on rules. Rules the controller would solve would for instance be ’IF aircraft are heading for loss of separation THEN action is needed’. For this action, no KBB would take place.

For the ’Interpret system state’ action, the controller might perceive an aircraft to be at a critical location. For instance, when an aircraft approaches the ILS, this signals that action must be taken in the future. Deciding which actions should be taken is RBB; the controller solves rules like ’IF the aircraft is close to the ILS, THEN the aircraft must be steered toward the ILS’. Here, KBB is defined as the evaluation of options for long term control, deciding which options would result in the most robust solution.

For the ’Evaluate goals’ action, the goals are separated. Here, SBB is not defined since for actions this high on the DL, SBB is no longer seen. RBB signifies evaluating short term functional purposes, such as safety (separation). The controller solves the rule ’IF separation is not enough, THEN increase separation’. KBB is needed to evaluate the long term goals (strategy) for the system, the efficiency functional purpose.

The ’Interpret target system state’ action is also divided in SBB, RBB and KBB; the definition of observable states is SBB, the future aircraft path definition is RBB and the most cognitive effort is needed to decide upon the path of aircraft streams (more high level planning).

Task definition is done by, for one aircraft, decide which location and heading it must have (SBB) and defining what its non-observable states should be (RBB).

When the ’Task’ knowledge state is achieved, the controller picks and locates the aircraft to steer (SBB) and defines the voice commands that should be given to the aircraft (RBB).

Note that for experienced controllers, some of the KBB and RBB can move to the RBB and SBB domains because of training and habit. One of the aims of the display that is to be designed is to move KBB and RBB to RBB and SBB, respectively. This can for instance be done by visualising certain states that are currently non-observable. Visualisation of options in the system can help controllers free up cognitive space for extra KBB tasks, since the display will execute some of the cognitive work instead of the controller.

5

Concept Design

In this section, concept displays will be developed. In Section 5.1, some relevant previous work is discussed and conclusions for this research are drawn. In Section 5.2, some initial concepts are treated and in Section 5.3, the Turn To Base (TTB) display concept will be treated in more detail.

5.1. Previous Work

5.1.1. Relative Position Indication

To effectively control aircraft, the arrival controller must have a mental model of future aircraft state. That is to say, a controller must estimate where a certain aircraft will be at a certain time in the future, relative to other aircraft in the area, and the controller must then assess if the aircraft will create a conflict in the future. To increase controller situational awareness, the Relative Position Indicator (RPI) was developed [25] by MITRE corporation.

The RPI is a tool that projects aircraft on the route of another aircraft. As such, ATCos can observe what the position of an aircraft will be in the future, when both streams are merged. This projection aircraft is called a 'ghost'. An illustration of a ghost can be found in figure 5.1, where the black aircraft (actual aircraft position) is merged with the aircraft stream coming from the right. The yellow plane is the ghost; it depicts the position of the actual aircraft relative to the aircraft in the other stream. Using this tool, ATCos can observe in an early stage what the separation of aircraft is going to be at the merging point (in this case, the runway). In EID terms, the ghost brings the KBB and RBB of the controller towards RBB and SBB; the controller can immediately observe the aircraft's relative position in the traffic stream (SBB) and based on this observation can decide that separation must be increased or decreased (RBB).

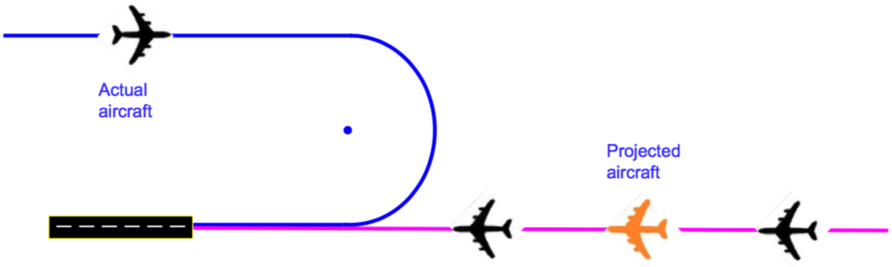


Figure 5.1: Illustration of an aircraft projected on a route of another aircraft, taken from [25]

While the concept of visualising ghosts has been around for some time (e.g. [26]), the RPI differs from most designs in that it is designed to project aircraft on routes that are not co-linear, i.e., the projection can take place on routes that consist of line pieces and turns with a turning radius. This is visualised in figure 5.2.

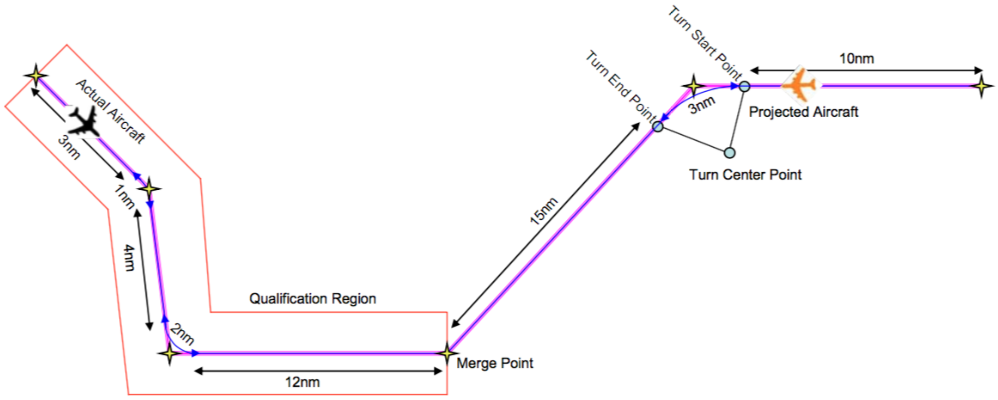


Figure 5.2: Projection of an aircraft on a non co-linear route, taken from [25]

Feedback on the RPI from controllers was very positive and human-in-the-loop simulations resulted in significant benefits of using the RPI, when reworking a set of arrival data from Ronald Reagan National Airport, Washington [25]. It can thus be concluded that the use of ghosts benefits the situation awareness of the ATCo and results in better air traffic solutions.

It is relevant to this research to investigate which problems arose in the development of the RPI, since this topic relates closely to the topic of trajectory prediction, which is something that will most likely be needed to create an useful support tool for the ARR controller. The RPI was developed from the Converging Route Display Aid (CRDA) [27], which had some issues that were fixed in the RPI.

In the past, the CRDA sometimes resulted in ’unstable or undesirable dynamics of the “ghost” targets such as jumping and hesitation’ [27]. This is important because laboratory experiments have pointed out that, in order for an ATCo to be able to make use of the ghost, the ghost aircraft must behave in a similar manner as the aircraft creating the ghost [27]. This means that the ghost needs to have a continuous track and the ghost needs to show the same (increases and decreases in) speed as the parent aircraft. The unstable dynamics of the ghost aircraft were mainly due to the algorithm which estimates the actual aircraft’s progress along the route. [27] suggests two methods to estimate the progress. For both methods, a qualification region is created; a region defined around a certain air route in which aircraft are assumed to be on that air route. Aircraft in this region will create ghosts, preventing ’stray’ aircraft from creating ghosts. An illustration of a qualification route can be seen in figure 5.2.

The first method proposes using parabolas as a division between legs of the air route, such that at every waypoint, there is an area that belongs to the previous and to the next leg of flight. This is illustrated in figure 5.3. In this figure, four aircraft tracks are visible The aircraft in this picture fly from left to right. The waypoints are given as red dots; the divisions between the flight route segments are the red parabolas. For this method, the distance the aircraft still has to cover is taken to be the distance of the aircraft to the next waypoint, plus all the legs the aircraft still has to traverse.

The problem with this method arises when aircraft steer earlier than anticipated, such as the black trail at the second red waypoint. Here, if the aircraft turns, the distance between the aircraft and the second waypoint will for a small time stay almost equal since the aircraft is travelling perpendicular to the waypoint. This will result in the algorithm assuming the aircraft stopped moving, and this will result in a stationary ghost. When the aircraft finally traverses the segment boundary, the distance between the aircraft and the next (third) waypoint will be smaller than anticipated, resulting in a jump of the aircraft ghost. This problem is illustrated in figure 5.4, where the speed of the ghost is given, relative to the parent aircraft speed. As one can see, for the black aircraft trajectory, the relative speed at $t = \pm 12700$ s is reduced to -400 kts before quickly jumping to > 1000 kts when the aircraft transitions to the next leg of flight. These jumps are quite large and their duration is quite long. Another method had to be found to cope with this problem.

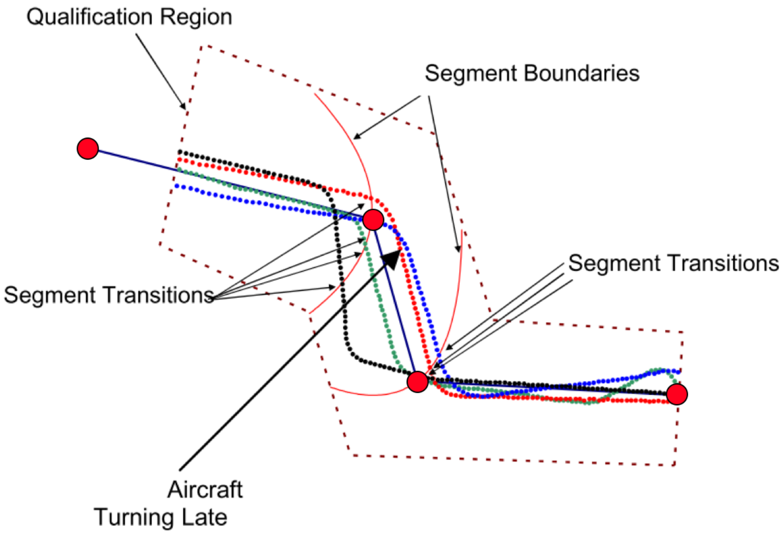


Figure 5.3: Division of air route legs between waypoints using parabolas, including four aircraft tracks and their segment transitioning points, adapted from [27]

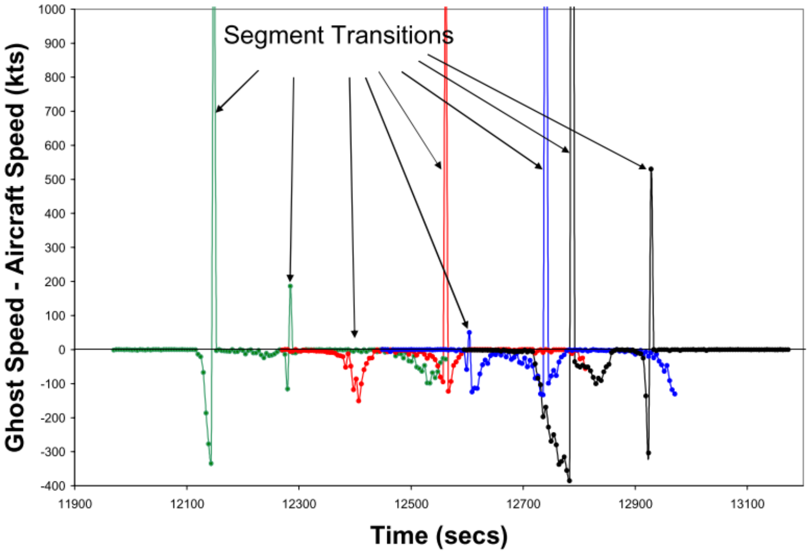


Figure 5.4: Relative speed of ghost w.r.t. parent aircraft for parabola segment decision, taken from [27]

The second method to derive the distance of the path the aircraft still had to travel uses a different way to divide the segments. For this method, the air path is divided in two kinds of segments: straight regions and turn regions. This is illustrated in figure 5.5. In this figure, the marked areas are turn segments divided by the bisector; a line drawn in a turn between the corners of the qualification region. Next, a turn centre is defined, based on bank angle or a standard predefined turn radius. This is notated with an L in figure 5.5. The aircraft location is then projected on the air route by taking the intersection between the air route and the line from the aircraft to the turning centre, as illustrated in figure 5.6. The distance to the next waypoint is then taken to be the distance between the projection and the waypoint. The lateral offset between the plane and the actual route is added as lateral offset to the route for the ghost. This method provides ghosts with a continuous groundspeed, although still showing small deviations from the parent aircraft speed for small periods of time (80 kts peak during 10 seconds). It was shown in later research that dynamics the ghosts created using this method were acceptable to ATCos [28].

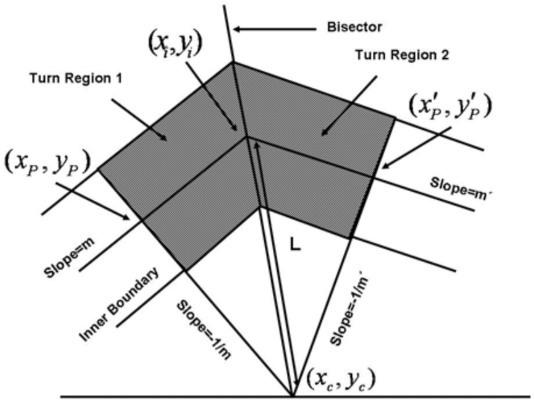


Figure 5.5: Division of air route legs between waypoints, dividing in straight and corner regions with bisector, taken from [27]

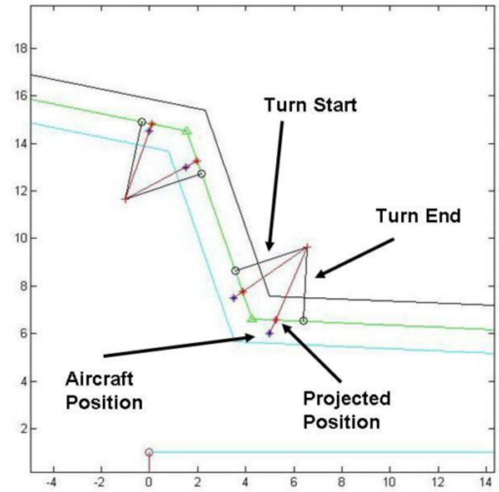


Figure 5.6: Projection of aircraft position in a turn around the turn centre, taken from [27]

For this research, ghosting is a very interesting tool to use in the concept. Ghosts allow the controller to observe future states of aircraft, a potentially very intuitive way of gaining more insight in the current state of the system. Ghosting will certainly be considered in the concept design. Next to this, the implementation of the ghosting algorithm will be considered when developing a ghosting algorithm in the future.

5.1.2. The Point Merge Concept

Another interesting research to look at is Eurocontrol's Point Merge Concept [29]. This is more of a concept than it is a display, however, it does provide insight in how problems affected by timing and spacing can be handled. A brief explanation, as well as conclusions that can be drawn from this research, will be given below. All information is taken from [29].

One of the tasks of an ATCo is to merge air traffic. Merging means bundling aircraft arriving from different directions into one stream of correctly separated aircraft. This task can cause high workload peaks in situations with a lot of traffic. To reduce workload, and to ultimately relocate the merging task to the flight deck instead of the air traffic controllers, the Point Merge Concept has been devised (see figure 5.7). In this concept, rather than the aircraft receiving instructions based on radar vectoring (where the ATCo places the speed vector of an aircraft in the right direction using voice commands), the aircraft receives a direct-to command to a merging point at a specific time. Before the direct-to command is given, the aircraft travels on a sequencing leg, ideally circular around the merging point. Sequencing legs are separated vertically for aircraft arriving from different directions, as to avoid head-on collisions. The advantage of this concept would be that controllers only have to give a direct-to command at the right time, such that when the aircraft leaves the sequencing leg, separation with the preceding aircraft is realised.

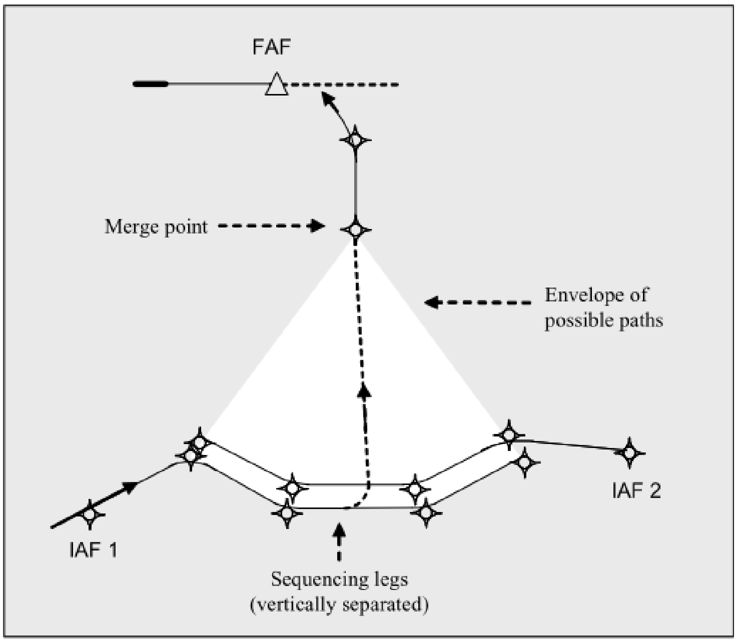

FAF
Merge point
Envelope of possible paths
IAF 1
IAF 2
Sequencing legs (vertically separated)

Figure 5.7: A point merge manoeuvre of an aircraft travelling on a sequencing leg [29]

Results of 6 experiment runs that were conducted using this concept were mostly positive. Working with the concept was reported to be 'comfortable, safe and accurate'. However, controllers did find the method less flexible than conventional radar vectoring, due to the predefined traffic routes. During the experiments, communication was also reduced significantly.

While this concept does not directly relate to TBS, its implications can be considered in the concept design. The idea of letting aircraft travel on a predefined path and only letting them of that path when separation allows it can potentially be used.

5.2. Preliminary Concept Displays

5.2.1. Display Design Considerations

From literature, we know that a display must be additional first because of regulations [19]. This is because the ATC business is highly regulated; every small change might affect system stability. Hence, the display to be designed will be designed as an addition to the current radar display.

To design a display that is effective in assisting the ARR controller, the CWA that has been performed in Chapter 4 will be considered. All five analyses will be used to create a display that gives the ARR controller insight in the degrees of freedom in the system, along with the constraints. Close attention must be paid to not showing too much degrees of freedom; this will result in cluttered displays and must be avoided.

For the work domain, the display will aim to make the work domain more transparent to the controller. By integrating data, certain attributes of the work domain (such as separation) can be visualised. This enables the ARR controller to have a more clear overview of the system. When one looks at the AH in figure 4.4, the display will try to visualise the more abstract parameters in the top of the AH.

For the control task, the display will try to facilitate more shunts and leaps in the decision making process. By linking actions and knowledge states directly to each other, the controller can skip certain cognitive actions, reducing workload.

For the strategies, the display will try to assist the controller in picking the right strategy at the right time. It will not give advisories on this; the controller must just be shown the possibilities in the system of aircraft.

For the social organisation and cooperation, the display will try to assist on more points that the computer actor currently does. At the moment, the human is responsible for almost all of the strategy information flow chart; the computer only delivers data. Whilst the final decision of picking and executing a strategy must always lie with the human controller (at least for this research), the computer can assist tin showing possible strategies that the human controller might have overlooked.

For the worker competencies, the display will try to integrate data in such a was that knowledge based behaviour is needed less. By integrating all factors in the system and presenting it as a range of options, the controller's behaviour moves more to the skill and rules domain.

5.2.2. Separation Margin Visualisation

Based on the display supposedly used by Heathrow Airport, described in section 3.2, a preliminary concept was conceived. This simple concept aims to show the necessary TBS distance and compressibility margin (based on the ORD marker in the Heathrow HMI). A difference with the Heathrow HMI is that the separation margin and the compressibility margin can be seen during the entire phase of flight, around the lead aircraft. In figure 5.8, a schematic of the display concept can be seen.

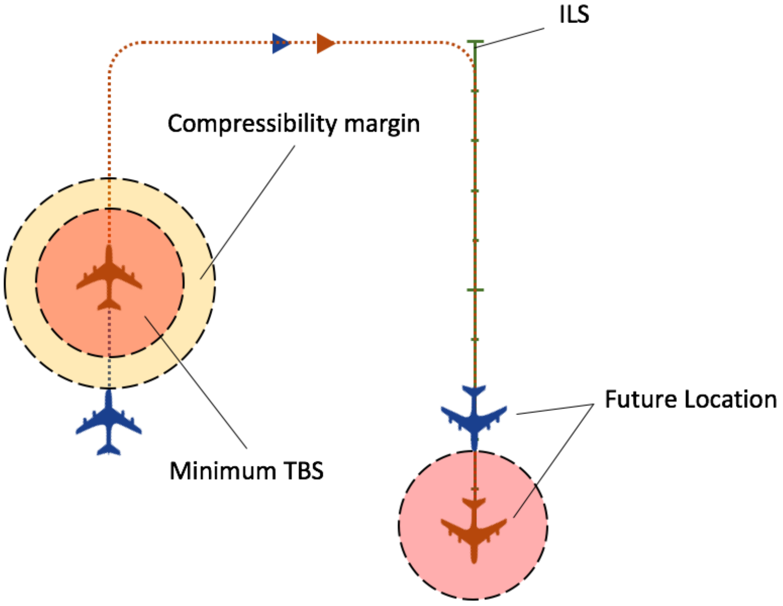


Figure 5.8: Schematic of TBS margin visualisation concept with compressibility margin and minimum TBS, beginning and final aircraft locations, no wind

In this figure, one can see two aircraft (left) that are on the downwind leg of their final phase of flight. The aircraft will, in this schematic, both follow the visualised path towards the ILS (in green, on the right of the image). The leading aircraft is shown with two circular areas around it; a red area depicting the minimum TBS needed for this specific aircraft pair and a yellow area depicting the distance that will decrease to zero when the lead aircraft has crossed the 4DME point on the ILS and slows down w.r.t. the following aircraft. The future location of the two aircraft, at the time that the leading aircraft touches down, is shown on the right of the picture. Here, the compressibility margin has reduced to zero and the separation between the aircraft is at its minimum, which will maximise the landing rate.

This concept has a few advantages. The display is simple and will minimise impact on the display of the ARR controller. This is beneficial for acceptation of the system in real life ATC; ATCos have a very difficult job and are often wary of a more cluttered radar display. The display is simple to use and the ARR controller can, in an early stage of approach, observe what the minimum TBS will be and what margin there is to tune separation.

The simplicity of this display is also one of its disadvantages; using the principles of EID the ARR controller can be assisted in a more meaningful way. The significance of the benefits of this concept w.r.t. the HMI used at Heathrow is most likely limited. Next to this, the display does not fully utilise the full capacity of the EID framework; only limitations are visualised and no options.

When introducing wind into the display, the circular areas around the leading aircraft will take some kind of egg-shape, visualised in figure 5.9. This visualisation possesses some interesting information due to the asymmetrical shape of the separation margin area around the leading aircraft. Using this shape, ARR controllers can estimate by how much the separation can be reduced in the coming phase of flight. For instance, when the leading aircraft in figure 5.9 turns to base, the asymmetrical separation margin shape will keep its orientation. This means that the separation can be reduced when the following aircraft turns to base. The same goes for when both aircraft turn into the ILS.

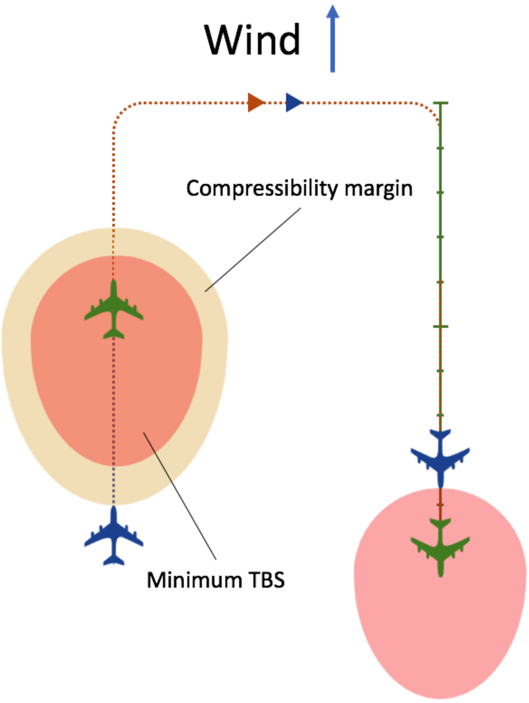


Figure 5.9: Schematic of TBS margin visualisation concept with compressibility margin and minimum TBS, beginning and projected final aircraft locations, with wind

As mentioned, these concepts can both lead to clutter since a lot of information is displayed which is not necessarily needed at all times. This can be solved by removing the areas around the leading aircraft and just using a coloured band with markers, as shown in figure 5.10.

5.2.3. Separation Reduction Possibilities Visualisation

The introduction of wind in the first concept led to the introduction of another concept. Since wind introduces an asymmetrical separation area shape, separation must be changed for every new orientation of the leading and following aircraft w.r.t. each other. This is visualised in the new concept, figure 5.11.

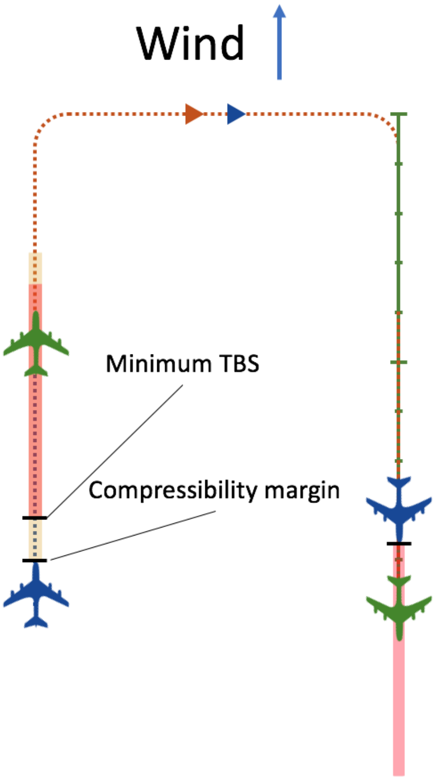

Wind
Minimum TBS
Compressibility margin

Figure 5.10: Schematic of TBS margin visualisation concept with reduced clutter, using just a coloured band and separation markers, beginning and projected final aircraft locations, with wind

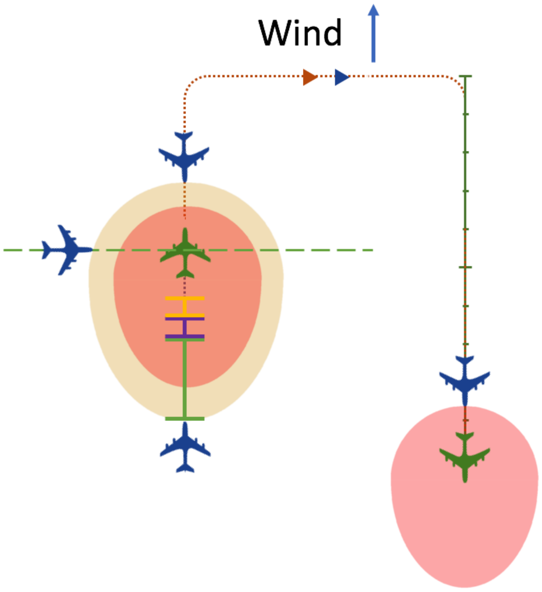


Figure 5.11: Schematic of second TBS margin visualisation concept with visualisation of separation reduction possibilities, beginning and projected final aircraft locations, with wind

As one can see, the separation between the blue following aircraft can be reduced in every new leg of flight. Currently, the aircraft are on the downwind leg, with the shown separation. However, on the base leg, the separation can be reduced. This is shown by the blue aircraft with the orientation it would have w.r.t. the green aircraft on the base leg. The green line in the separation indicator shows how much this distance differs from the current separation. As such, the ARR controller knows that the separation can be reduced in the base leg by as much as the green line shows. Similarly, the purple line shows how much separation can be reduced in the final ILS leg of flight, and the orange line shows how much separation can be reduced when the compressibility margin is reduced to zero, when the leading aircraft lands. This concept basically visualises an extended compressibility margin, where the compressibility in this case is due to the asymmetrical distribution of the separation margin due to wind.

An advantage of this display would be that it gives the controller information on future states of separation, enabling the controller to plan his/her next move. However, the result might cause cluttered displays since a lot of information is being displayed simultaneously. As such, it will be difficult to interpret the information in one observation; the controller must actively interpret the information in order to process its meaning. This is something that is ideally avoided; an ideal display is intuitive and facilitates quick responses.

5.2.4. Turn To Base Possibilities Visualisation

An entirely different display concept was developed to assist ARR controllers in deciding where to let aircraft turn to base. As discussed in section 2.5.1 and2.5.2 , the point where aircraft turn to base is an important variable in separation of aircraft. This concept aims to show all the possible solutions where the aircraft can turn to base. The idea is that when the controller selects an aircraft, the system marks the area in the display where the aircraft can safely turn to base, without causing a separation conflict in the future. This is illustrated in figure 5.12.

In this figure, the blue aircraft is selected. The leading (red) aircraft and the following (yellow) aircraft are also depicted. On the left, the current situation is given. The red and yellow aircraft are assumed to follow some predicted path, as shown. When the blue aircraft travels on its downwind leg, the controller can let it turn to base in the green area. If the aircraft turns to base when it is just in the green area, it will end up just before the red aircraft on the ILS, as depicted on the right (future situation). If the ARR controller chooses to let the aircraft turn to base at the last possible moment (the edge of the green circular area), the blue aircraft will end up just before the yellow aircraft. By varying the point at which the aircraft will turn to base, the blue aircraft's

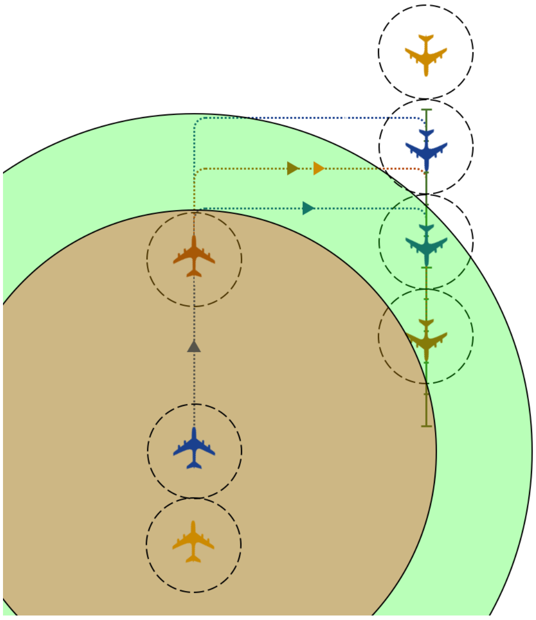


Figure 5.12: Early illustration of TTB possibility visualisation concept, beginning and projected final aircraft locations

future location can be varied in a range as shown on the right of the picture.

This concept shows all options the controller has in one observation. It assists in the decision making process of the controller and is intuitive. While the other concepts show what the separation must be, this concept assists the controller in attaining the optimum separation. It is therefore the best fit for this research. A disadvantage of this display is that the prediction of the trajectories of all other aircraft in the neighbourhood of the selected aircraft needs to be fairly accurate.

It was chosen to work out this concept in more detail, since it is expected to have the most potential and since the possibility of assisting the controller in attaining optimum separation is promising.

5.3. The Turn To Base Display

The concept described in section 5.2.4 has been worked out in more detail. In figure 5.12, the area where the aircraft can turn to base is visualised as a circle. The radius in this case is derived using the current path of the aircraft. However, it would be more interesting to visualise this circle for every heading angle the aircraft can take w.r.t. its position. For instance, it would be beneficial for the ARR controller to observe where the aircraft can turn to base if it were to keep its heading, while simultaneously being able to observe where the aircraft can turn to base would it have another heading angle entirely. This way, the display would support dog-legging as well as tromboning. Even maneuvers that combine dog-legging and tromboning together can be supported. This way, the controller is not restricted to a certain maneuver; the ARR controller remains in control at all times but is simply shown the constraints of the system.

5.3.1. Offline Simulations

To get an initial feeling of the shape of the area in this display, offline simulations in Matlab were conducted. These simulations were very basic; it was assumed all aircraft travel with a constant speed of 220 kts at sea level and the turns they make are instantaneous (no turning radius). Furthermore, wind was assumed to only affect the speed of the aircraft and not their trajectory. Minimum DBS was assumed to be 4 NM for every aircraft and minimum TBS was assumed to be 65.5 s, the time it would take an aircraft with a speed of 220 kts to travel 4 NM. The algorithm for the offline simulation evaluated a certain amount of points per a certain amount of heading angles. For this point it was evaluated if the time it would take the aircraft to traverse via this TTB point would be a 'safe arrival' time, based on the times at which the other aircraft were predicted to land. Safe points were plotted as green points, unsafe points as red.

The result of the first offline simulation can be seen in figure 5.13. In this figure, one can observe the aircraft for which the TTB possibilities are plotted at the coordinate (-8, 0), in red. Another, leading aircraft is shown at (-8, 7), in blue, along with its predicted trajectory towards the runway threshold, at (0, 0). The ILS is plotted as well, from (0, 0) to (0, 20). This plot shows the area for which the red aircraft can turn to base in green. For instance, if the red aircraft were to travel to point A, a green point, and then turn to base to the ILS, catch the ILS and land on the runway at (0, 0), the red aircraft would be separated enough from the leading blue aircraft. In this case, it would land after the blue aircraft has landed. This corresponds to the path in black. If the red aircraft were to travel to point B, a red point, after which it would turn to base, catch the ILS and land, it would not be safely time-based separated and separation between the aircraft and the blue leading aircraft would be lost. If the red aircraft would turn to base at point C, a green point again, the aircraft would end up on the ILS before the blue aircraft and the blue aircraft would be safely separated behind the red aircraft.

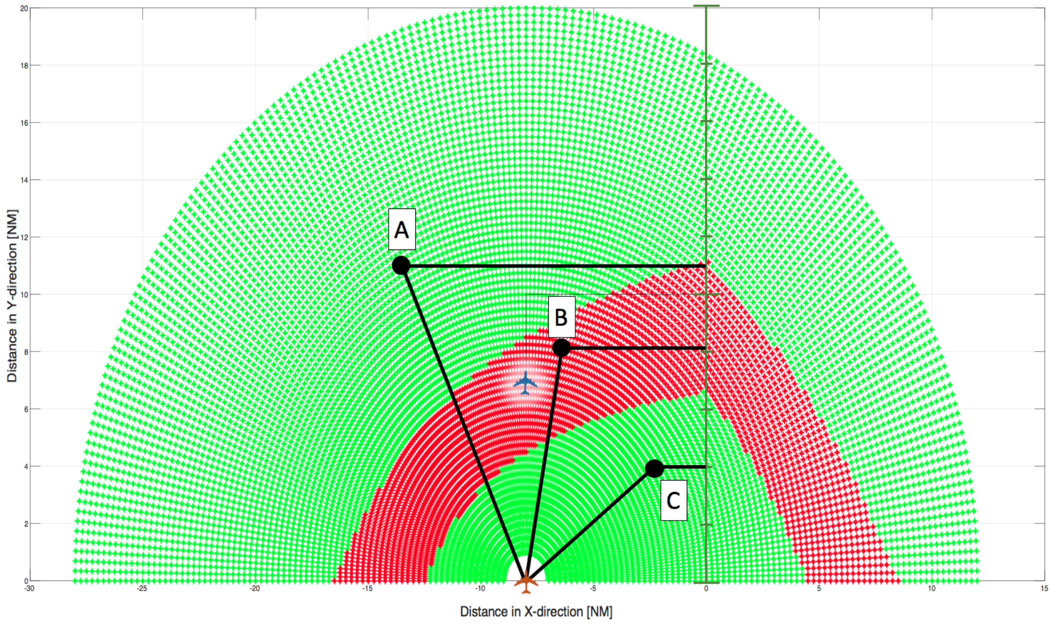


Figure 5.13: TTB display concept showing locomotion space where TTB is possible (in green), no headwind

The addition of wind can be seen in figure 5.14. Here, 50 kts headwind (to the north) is introduced. The plot for no headwind is showed in dark green, in a band under the red band in the display. As one can see, the red area shifts to the north when a southern wind is added. The shape of the constraints in the system does not change dramatically, it just moves.

The situation for two aircraft in the area can be observed in figure 5.15. Here, next to the blue aircraft, an orange aircraft is added at (20, 0), with its planned trajectory using a TTB at (20, 10). The orange aircraft results in another red area. The red aircraft can now choose to turn to base at a variety of locations resulting in it catching the ILS in front of both other aircraft, in between or behind them.

In figure 5.16, a yellow margin is plotted. This margin is plotted such that when the red aircraft chooses to TTB at a point on the border of this yellow margin, the time separation between the the aircraft will be exactly 30 seconds. This way, the margin the ARR controller has is visualised in an intuitive manner. For instance, a TTB point for the red aircraft at point A (red) would result in loss of separation, a TTB at point B (yellow) would result in the red aircraft being separated by less than 30 seconds in front of the blue aircraft, and a TTB at point C (green) would result in a separation of more than 30 seconds in front of the blue aircraft.

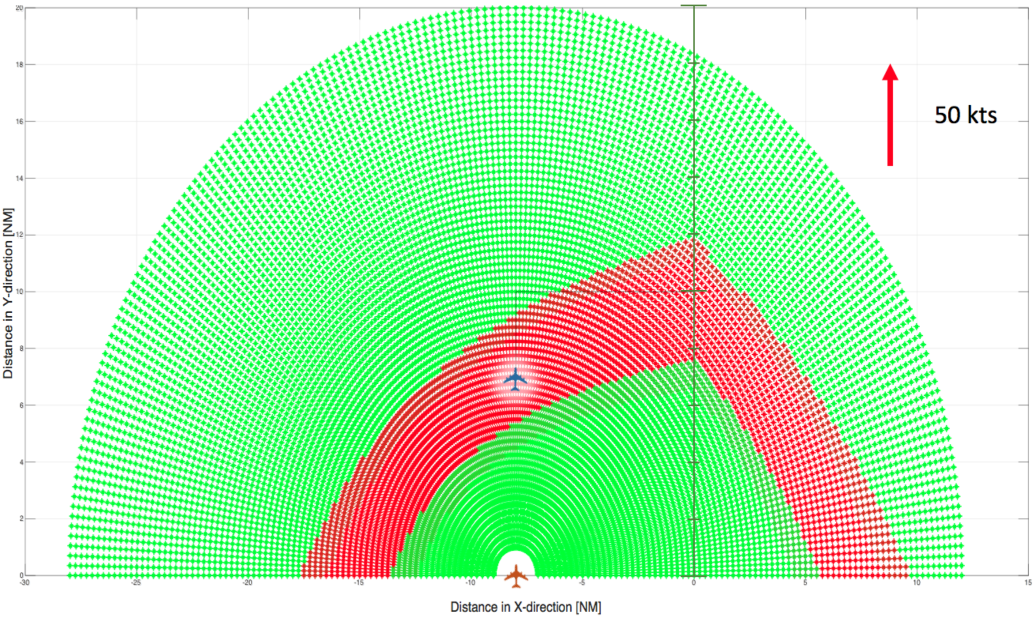


Figure 5.14: TTB display concept showing locomotion space where TTB is possible (in green), 50 kts headwind

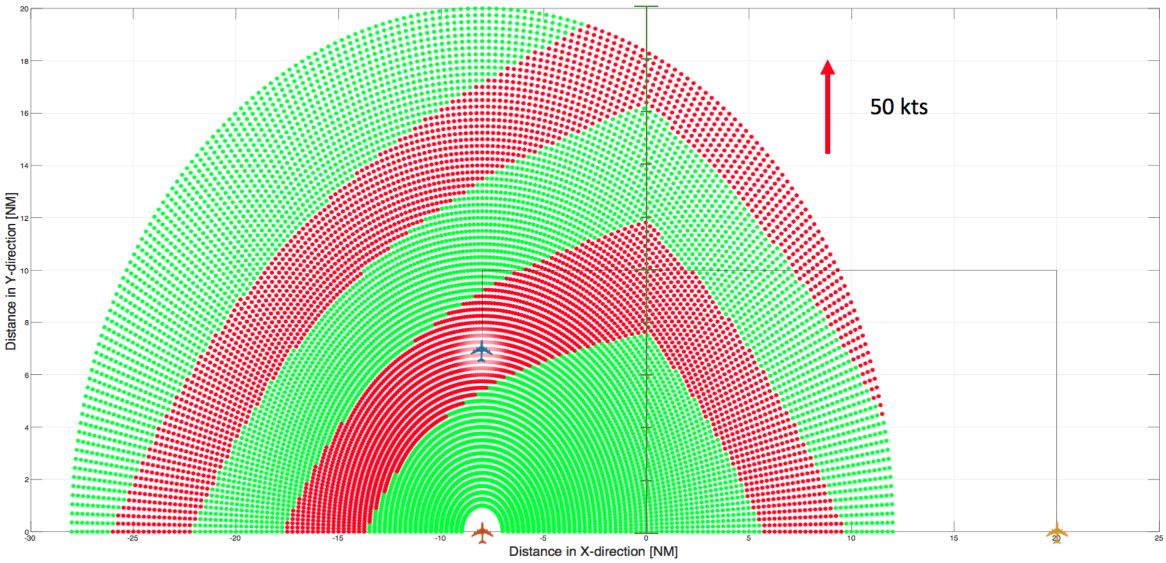


Figure 5.15: TTB display concept showing locomotion space where TTB is possible (in green), 50 kts headwind and two aircraft in traffic

5.3.2. Variations on the TTB Display

The offline simulations show some promising results. The displays looks intuitive and could potentially be really useful. Of course, when the display will be developed, possible variations might be devised to research their possible benefits.

As explained in section 5.1.1, ghosting of aircraft can be a powerful tool. It can be researched if applying ghosts to the TTB display will be beneficial. For instance, one can plot the future location of the aircraft on the ILS w.r.t. the leading aircraft when it is at the runway threshold, based on the position of the cursor. An example can be given by looking at figure 5.17. Here, the red aircraft is selected. If the user were to hover its cursor above point A, a ghost of the red aircraft projected on the ILS can be displayed. This ghost would reflect the future separation, at the moment when the blue aircraft reaches the runway threshold. This would further increase understanding of the ARR controller in the effect of using different TTB strategies.

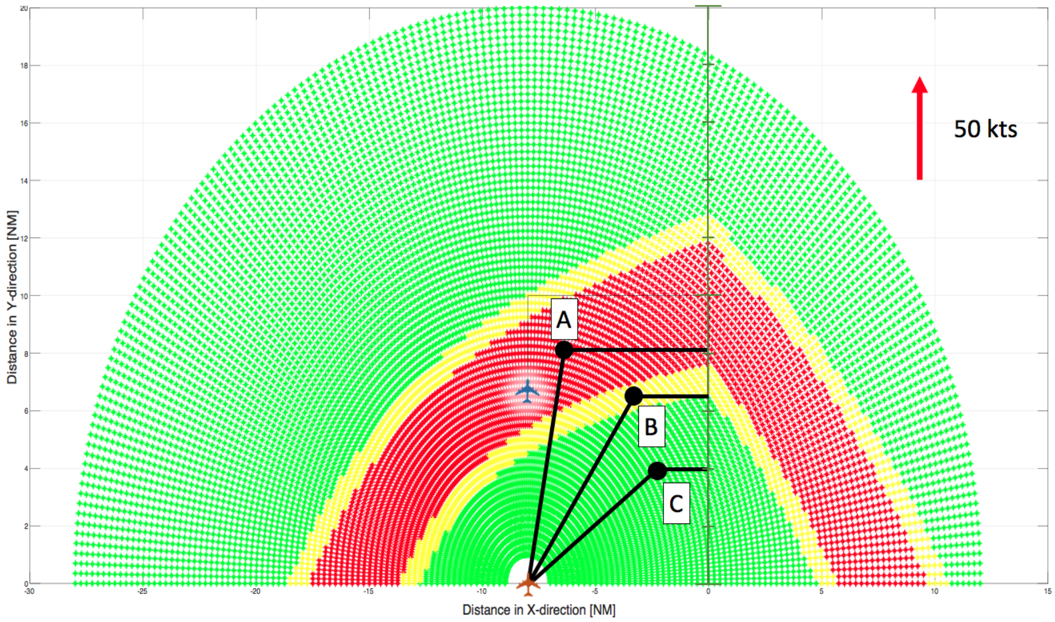

50 kts
A
B
C
Distance in Y-direction [NM]
0
2
4
6
8
10
12
14
16
18
20
Distance in X-direction [NM]
-30
-25
-20
-15
-10
-5
0
5
10
15

Figure 5.16: TTB display concept showing locomotion space where TTB is possible (in green), 50 kts headwind yellow band for 30 second margin

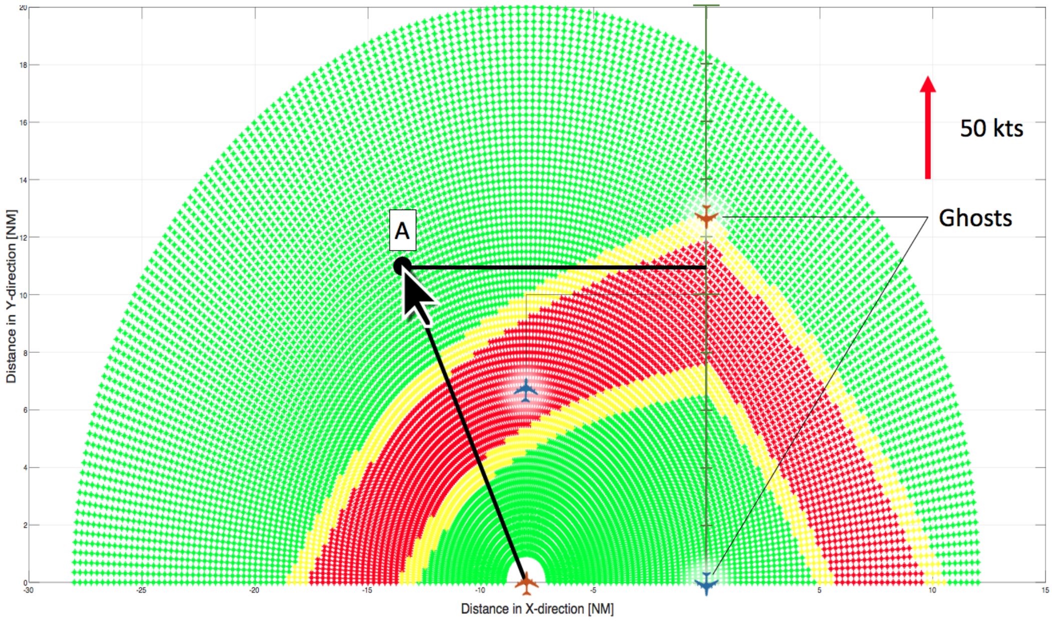

50 kts
Ghosts
A
Distance in Y-direction [NM]
0
2
4
6
8
10
12
14
16
18
20
Distance in X-direction [NM]
-30
-25
-20
-15
-10
-5
0
5
10
15

Figure 5.17: TTB display concept showing ghosting of aircraft future location while user hovers around the TTB area

Another tool that might be used is toggling of different speed options. Using the mouse wheel or keyboard, the TTB can be displayed for imaginary aircraft states. For instance, if the red aircraft in figure 5.17 flies at 220 kts IAS at the moment, the user can select 200 kts IAS with the mouse wheel to observe what the TTB display would look like if the red aircraft would fly at 200 kts IAS. This way, the controller gets more insight in his/her options.

5.3.3. The TTB Display in the Cognitive Work Analysis

The CWA (chapter 4) will be affected with implementation of the TTB display. The effect of the TTB display will be evaluated using the AH, the DL and the SRK taxonomy.

The TTB display will have its effect on the availability of some of the parameters in the AH displayed in figure 4.4. The ARR controller will be able to perceive some of the parameters that are currently not visible to him/her. Note that the AH itself does not change, merely the transparency of the system.

A first change in perception for the ARR controller would be the availability of the locomotion space. The TTB display does not show the entire locomotion space and range of possibilities, since this would be very difficult to implement in one display in an intuitive manner. However, a large portion of the locomotion state space is visualised when enabling the controller to see where a TTB is possible This relieves the controller from his/her need to infer the range of possibilities from the relatively small amount of data available on the radar screen. This would eliminate some of the need of the controller to show KBB and would result in a shift of behaviour to the SBB and RBB side.

The controller will also be assisted in the parameter 'future aircraft states', would ghosting be used. Ghosting enables the controller to actually perceive the future aircraft state, albeit just for a certain predefined scenario. Hence, the controller will also be partially relieved from the cognitive task of estimating future states.

Likewise, the TTB display integrates all traffic in the options for the TTB possibilities. The ARR controller will thus be assisted in the decision without first having to evaluate all traffic conditions.

The same goes for the parameters wind and aircraft performance. Both wind and aircraft performance are incorporated in the trajectory prediction of the TTB display such that here cognitive effort for the controller is reduced as well.

In figure 5.18, a decision ladder can be observed for the ARR controller task with the TTB display activated. Two more leaps can be identified.

Firstly, a shunt from the action 'Observe information & data' to the knowledge state 'Options' is defined. The controller selects an aircraft after he/she has reached the 'Alert' knowledge state. When he/she selects an aircraft, the TTB display instantly shows all TTB options, integrating all traffic and weather conditions in the options in one go. The controller can directly evaluate goals and pick an option.

A leap is defined from the 'Set of states' knowledge state to the 'procedure' knowledge state, since the controller may identify a procedure using the tool since the tool integrates the system state for the controller. Hence, the controller can make the leap from 'System state' to 'Procedure' starting from 'Set of states', since the system integration is done by the display.

The TTB display will also affect the SRK taxonomy set up in Section 4.6. In figure 5.19, the effect of the TTB display can be seen. The green arrows depict a shift of behaviour.

Using the TTB display, the controller can immediately see the options for an aircraft when the aircraft is selected. As such, the controller no longer has to use cognitive effort to evaluate the options of aircraft to turn to the ILS without loss of separation, because they are all given by the TTB display. The controller can simply pick an action based on the rule: 'pick a waypoint for the aircraft in the green area of the domain'. Thus, the 'Evaluate options for long term control' KBB for the 'Interpret system state' will move to the RBB domain. Likewise, the controller does not have to infer states to reach a conclusion on actions to be taken. The actions that can be taken are visualised. The 'Identify actions to be taken using inferred states' RBB will move to the SBB domain for the 'Observe information & data' action, since in this action already the system is integrated by the display. This is also visualised in the leap in the DL in figure 5.18.

For the 'Interpret target system state' action, the 'Define future aircraft streams' KBB will move to the RBB domain. This is because the TTB display makes it unnecessary to estimate streams; the display automatically integrates all traffic in the options that are visualised and thus the controller can act based on rules like 'IF the aircraft turns to base in a green point, THEN the aircraft streams will be separated sufficiently'. The 'Define future path' RBB will move to the SBB for the 'Define task' action; the picking of an aircraft path will require minimal cognitive workload since the display does all the 'heavy lifting' and will at once define the task to take. This can also be seen in the DL represented by a shunt, figure 5.18.

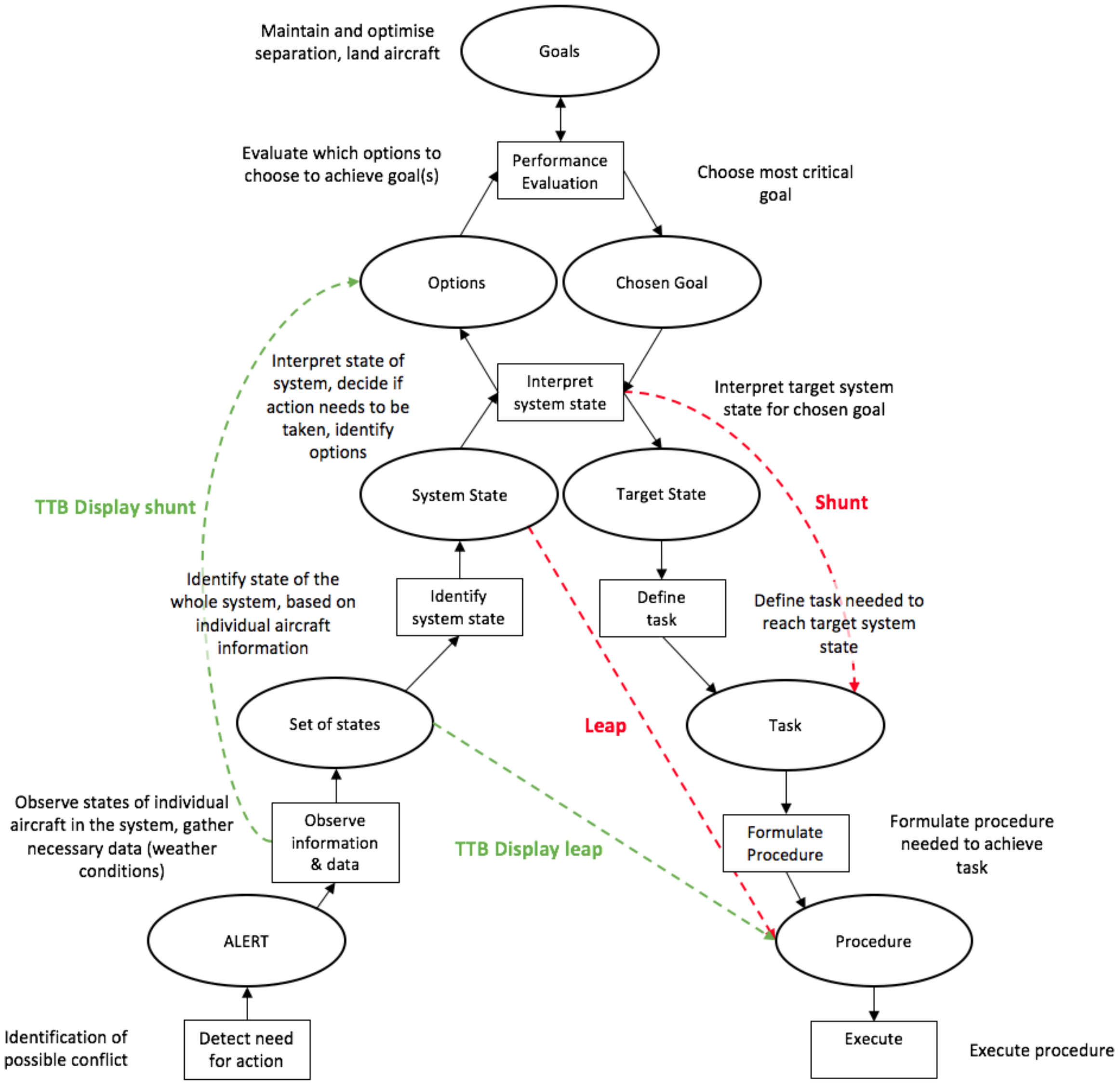

Maintain and optimise separation, land aircraft
Goals
Evaluate which options to choose to achieve goal(s)
Performance Evaluation
Choose most critical goal
Options
Chosen Goal
Interpret state of system, decide if action needs to be taken, identify options
Interpret system state
Interpret target system state for chosen goal
TTB Display shunt
System State
Target State
Shunt
Identify state of the whole system, based on individual aircraft information
Identify system state
Define task
Define task needed to reach target system state
Set of states
Leap
Task
Observe states of individual aircraft in the system, gather necessary data (weather conditions)
Observe information & data
TTB Display leap
Formulate Procedure
Formulate procedure needed to achieve task
ALERT
Procedure
Identification of possible conflict
Detect need for action
Execute
Execute procedure

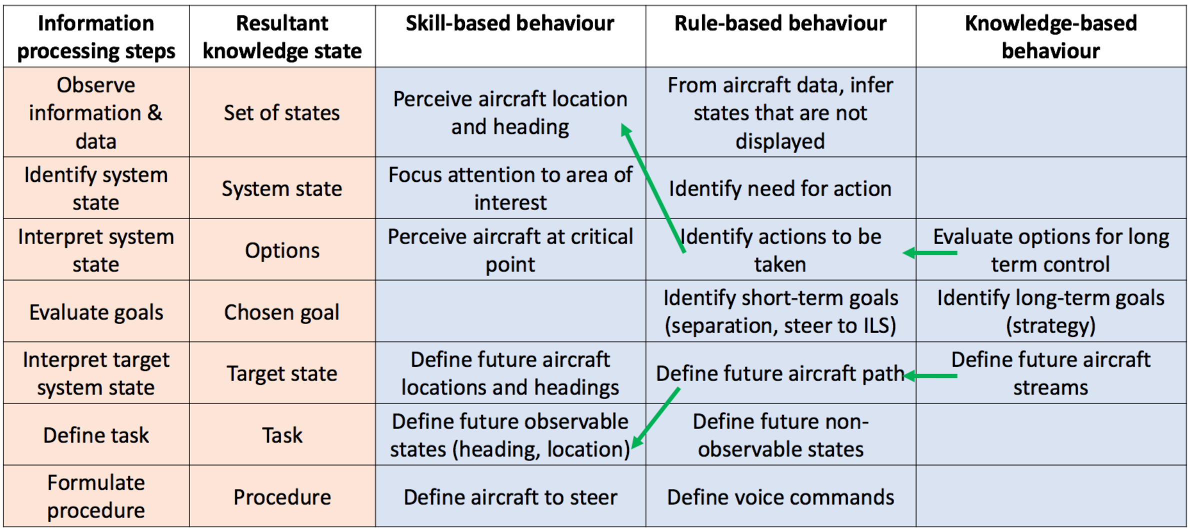
Figure 5.18: The decision ladder for the arrival controller task with implementation of the TTB display'

One can observe that the use of the TTB display would decrease controller cognitive workload and move KBB and RBB to RBB and SBB, respectively. As such, the TTB tool could provide an interesting aid for ARR controllers, reducing their workload and freeing up cognitive space for other tasks, such as dealing with unanticipated events. It will therefore also increase safety in the system.

5.3.4. Critical Analysis and Hypothesis

Next to these positive effects of the TTB display, there might also be some negative effects. It has been debated whether the use of displays that show the boundaries of constraints result in the operator picking unsafe strategies [30]. It has been shown that experienced operators make smarter and more robust decisions, and do not pick 'boundary solutions', which can be unsafe [30]. However, this does mean that the display will perhaps not be safe for training purposes, as it might teach student ATCos unsafe strategies.

Next to this, the functioning of the TTB display is highly dependant on trajectory prediction of aircraft in the sector. If the estimated arrival times of traffic are significantly off, the boundaries of the TTB will change. Next to this, the TTB display for one single aircraft will be influenced if another aircraft picks a strategy that is not



Information processing steps	Resultant knowledge state	Skill-based behaviour	Rule-based behaviour	Knowledge-based behaviour
Observe information & data	Set of states	Perceive aircraft location and heading	From aircraft data, infer states that are not displayed	
Identify system state	System state	Focus attention to area of interest	Identify need for action	
Interpret system state	Options	Perceive aircraft at critical point	Identify actions to be taken	Evaluate options for long term control
Evaluate goals	Chosen goal		Identify short-term goals (separation, steer to ILS)	Identify long-term goals (strategy)
Interpret target system state	Target state	Define future aircraft locations and headings	Define future aircraft path	Define future aircraft streams
Define task	Task	Define future observable states (heading, location)	Define future non-observable states	
Formulate procedure	Procedure	Define aircraft to steer	Define voice commands	

Figure 5.19: The SRK taxonomy for the ARR controller task, as affected by the TTB display

as one would expect. If the operator trusts the TTB display too much, the safety of a solution for one aircraft might be compromised by altering the course of another aircraft in the sector. Hence, the robustness of both the time predictor and the TTB display as assisting tool must be thoroughly tested.

A last possible problem is the acceptation of the display by controllers. ATCos are very good at what they do but they are known be reluctant to accepting new technology, since any change in their work environment can be an unstable parameter, something that is unwanted in a high risk environment. Hence, the display should be designed such that it is as transparent as possible, increasing controller understanding of the underlying physics.

The hypothesis is that operators using the TTB display will pick a strategy sooner than they would before. That is, as soon as an aircraft enters the sector, the controller can immediately pick a path for the aircraft such that it will be separated for its entire journey to the runway threshold. It is also expected that the implementation of the TTB display leads to fewer controller interventions, since last minute corrections will become redundant.

6

Preliminary Work

In this chapter, the work that has already been done will be treated. In Section 6.1, the adaptation of a ATC simulator for arrival scenario simulation will be discussed. In Section 6.2, an initial trajectory predictor will be devised.

6.1. ATC Simulator Sector X

To be able to quickly test preliminary ATC prototype displays, an environment to rapidly prototype ATC applications has been developed by Clark Borst and Rolf Klomp at Delft Technical University, Faculty of Aerospace Engineering for the Control & Operations division. In this medium fidelity ATC simulator, called Sector X, the radar display used by the ACC controllers is simulated. A screenshot of a simulated sector with two aircraft in it can be found in figure 6.1a. Note that the aircraft symbols are not to scale, the width and height of this sector is 72 NM. This size mirrors the ARR controller display which is circular with a radius of 36 NM. Also note that the images of the simulator are transferred to black and white since the original colours, black and green, do not transfer well on printing paper.

Sector X gives the user the possibility to create their own ATC scenarios. In the simulator, the user can manipulate the aircraft in the sector using either direct manipulation or using a virtual command module, displayed on the computer screen. The user can define waypoints where the aircraft should be steered towards to create realistic simulations.

6.1.1. Aircraft Visualisation in Sector X

Figure 6.2 shows some versions of aircraft in Sector X. In figure 6.2a, an aircraft travelling in a straight line can be seen. In the tag next to it, the aircraft state is displayed. From left to right, from top to bottom, the Aircraft ID (ACID), the current altitude (in FL), the target altitude (in FL), the IAS, the TAS, the heading angle (in degrees), the type of aircraft (light, heavy and medium) and the name of the waypoint where the aircraft should be steered towards. The dots behind the aircraft are so-called history dots, each dot shows were the aircraft was 5 seconds before the next dot. The aircraft speed vector is also drawn.

In figure 6.2b, the aircraft is selected by the user. Here, the outer circle represents the maximum speed, the inner circle represents the minimum speed and the dotted circle represents the current speed.

In figure 6.2c, a descending aircraft is shown. This is indicated by a downwards arrow next to the target flight level entry in the aircraft label. As can be seen in the figure, this aircraft is currently on FL 56 descending to FL 30.

In figure 6.2d, a turning aircraft is displayed. The history dots show the path the aircraft has taken. The line at 3 'o clock displays the target speed vector of the aircraft (90 degrees), whilst the line at 12 'o clock displays the speed vector of the aircraft before initiating its turn (0/360 degrees).

[!h]

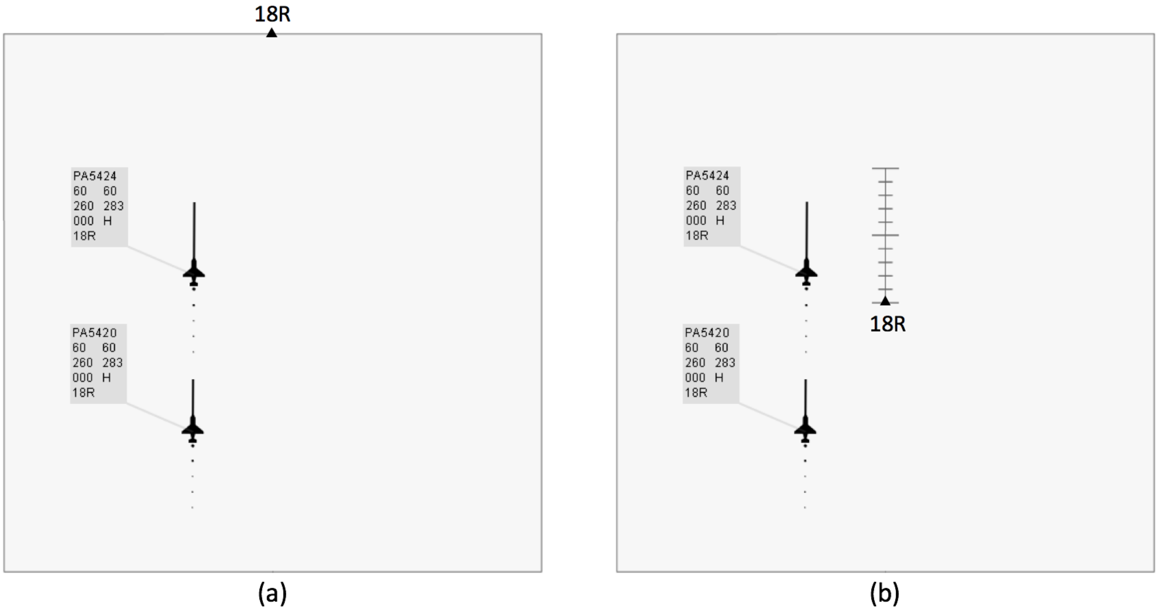


Figure 6.1: A screenshot of two aircraft in a sector with (a) a waypoint 18R and (b) an ILS for runway 18R, as shown in Sector X

6.1.2. Adapting Sector X

Sector X was designed for ACC control. As such, for this research, Sector X had to be adapted for ARR control. This meant visualising an ILS and making the aircraft able to land on this ILS.

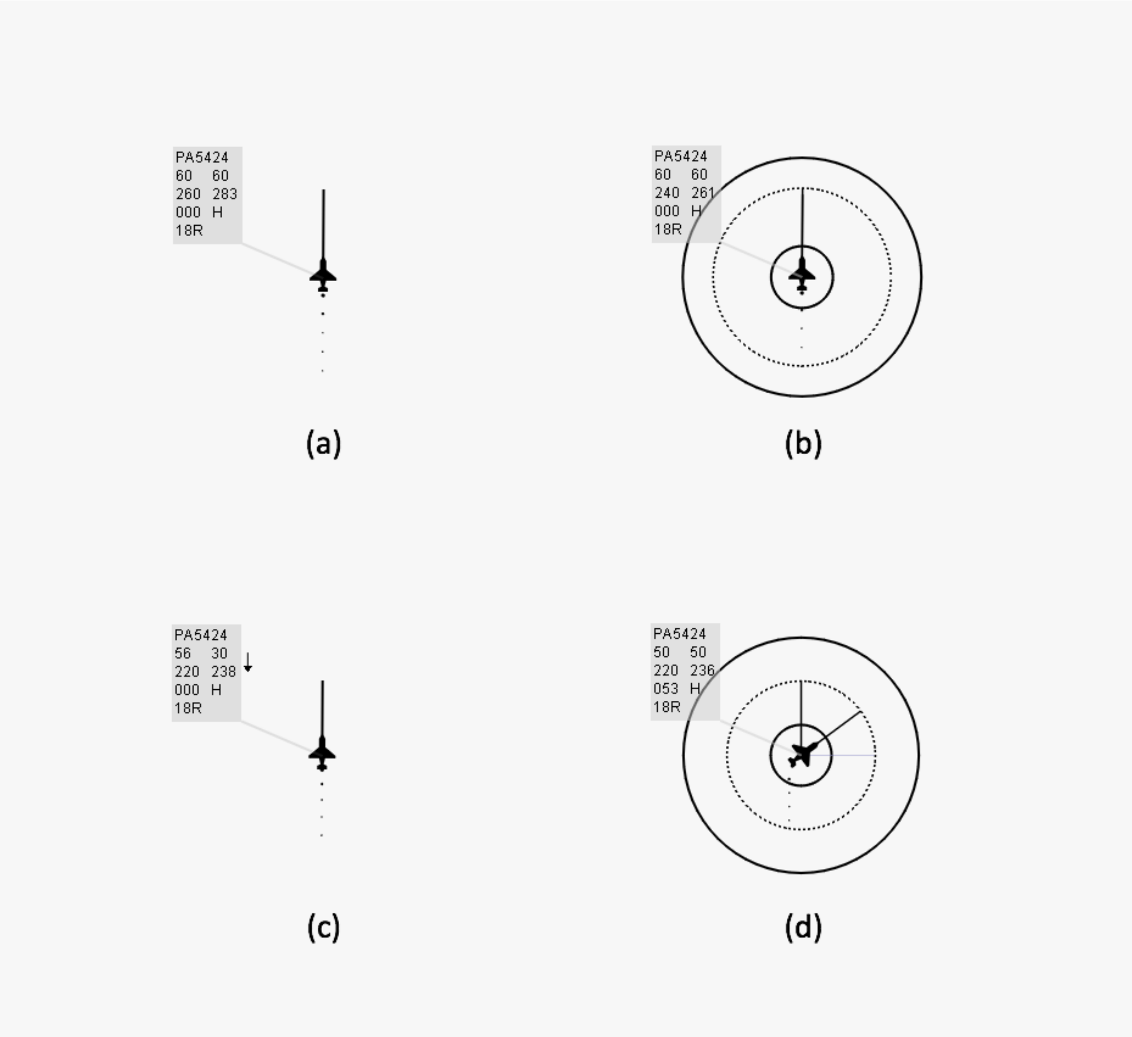
In reality, an aircraft flying on IFR can catch the localizer/glideslope on its own. The ARR controller gives the command 'cleared to catch localizer/glideslope', after which the pilot sets the autopilot to 'catch localizer/glideslope'. The autopilot then intercepts the radio signal and steers the aircraft into the localizer/glideslope. In Sector X, this function had to be added to simulate real life approach scenarios.

In the virtual Control Display Unit (CDU) that is incorporated in Sector X, a 'catch ILS' button was implemented. The resulting CDU can be observed in figure 6.3. Here, the ILS button is inactive; the current selected aircraft has not received a 'catch ILS' command or no aircraft is selected. If an aircraft is selected and the ILS button is clicked, the button will become active and the aircraft will catch the ILS when it encounters the ILS.

The algorithm to catch the localizer was defined in the following way. Every aircraft in Sector X has a turn radius, based on their maximum bank angle, as in equation 6.1. This radius is taken as the turn radius for every turn in the simulator.

$$r_{min} = \frac{V_g^2}{g \tan\left(\phi_{max}\right)} \tag{6.1}$$

When the 'catch ILS' command has been given, the aircraft checks for every timestep if the distance from the ILS where the aircraft needs to turn into the ILS has been crossed. This is depicted in figure 6.4. The timestep used in the simulator is 5 seconds, equal to the refresh rate of the aircraft locations on an ARR controller radar screen. This large timestep poses a complication in the algorithm, illustrated in figure 6.4. In this figure, an aircraft at point 1 is travelling towards the 10 NM mark of the ILS, depicted on the right, in steps of 5 seconds, via points 1 , 2 and 3. A circle with as radius the turning radius as derived in equation 6.1 has been drawn parallel to the ILS. At the point where the trajectory of the aircraft touches the circle, the aircraft should turn into the ILS to perfectly line up on the ILS. However, the aircraft will only detect its need to turn at point 3, where it already needed to start its turn somewhere between points 2 and 3. This results in overshoot and ultimately the aircraft not lining op on the ILS. This was countered by letting the aircraft detect when the


PA5424
60 60
260 283
000 H
18R
(a)
PA5424
60 60
240 261
000 H
18R
(b)
PA5424
56 30
220 238
000 H
18R
(c)
PA5424
50 50
220 236
053 H
18R
(d)

[!h]

Figure 6.2: A screenshot of an aircraft in four different situations: (a) an unselected aircraft travelling in a straight line, (b) a selected aircraft travelling in a straight line, (c) an unselected descending aircraft and (d) a selected aircraft in a turn

turn-in point was crossed, and then mapping the overshoot on the ideal turn circle. The aircraft location and heading was then revised. This new location is illustrated as point 4 in the picture.

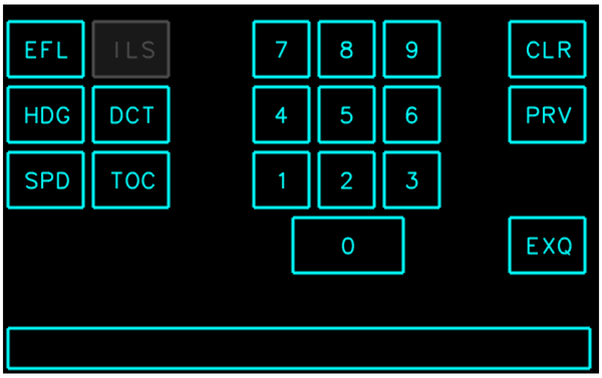

EFL
ILS
7
8
9
CLR
HDG
DCT
4
5
6
PRV
SPD
TOC
1
2
3
0
EXQ

Figure 6.3: The Control Display Unit in Sector X, adapted to include ILS catch function

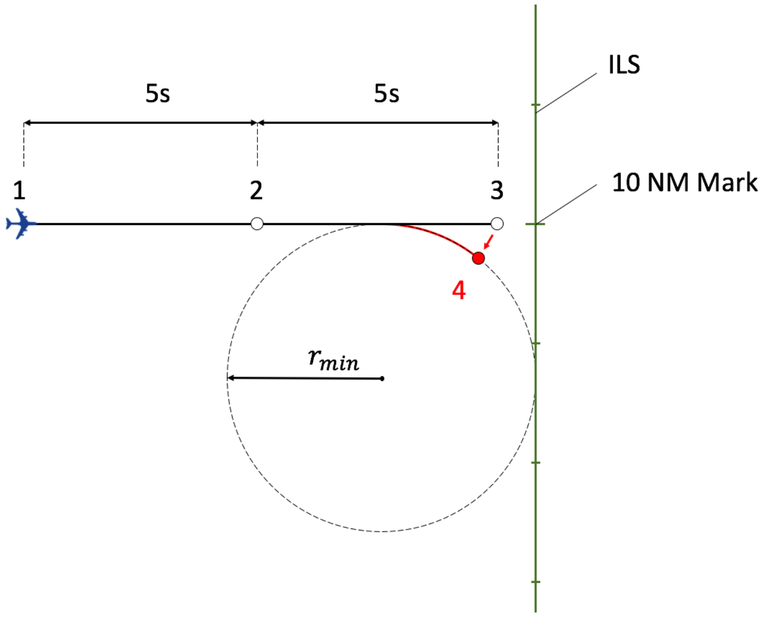


Figure 6.4: Picture of the ILS catch algorithm, illustrating compensation for overshoot due to the large timestep used in the simulation

6.2. Trajectory Prediction

Perhaps one of the most important parameters governing when an aircraft can turn to base is the estimated time that the leading aircraft will land. As such, it is essential to be able to predict the trajectory of an aircraft in four dimensions; 3 spatial and one temporal. A preliminary trajectory predictor was made to estimate the trajectory each aircraft would have in the sector. Based on the nominal trajectory and speed instructions described in section 2.5.2, this trajectory resulted in a nominal predicted arrival time for all aircraft. First, an estimator for flights arriving from the RIVER IAF was made. After this, the estimator could be extended.

The trajectory predictor worked based on the nominal trajectory as described in 2.5.2. Each aircraft was assumed to approach the ILS using the trajectory defined in figure 2.12. To estimate in which leg of flight the aircraft was, conditions were set up for each leg of flight. Details on leg estimation can be found in Appendix C.

Using the predicted leg of flight, the trajectory can be predicted. Using the speed data points as shown in figure 2.12, the Estimated Time of Arrival (ETA) could be derived. Here, the IAS was assumed to change linearly with distance until the next speed data point. For instance, if the aircraft is travelling in leg 1 with a speed of 250 kts IAS and needed to fly 30 NM to reach the beginning of leg 2, the aircraft would be assumed to decelerate with 1 kts IAS per NM. The descent profile of the aircraft was assumed to be continuous from the current altitude to an altitude of 2,000 ft at the beginning of leg 7, where the aircraft is just lined up with the ILS. To compensate for the difference between GS and IAS and the non-linearity between them, a function was written that evaluated the starting and final values of the altitude and IAS, integrated the resulting GS for several steps and returned an average GS. This speed was then used as the speed of the aircraft during that leg for time estimation purposes.

While the algorithm is not yet fully verified, initial testing provided robust behaviour. Test flights were flown and timed and no large discrepancies in the algorithm were found. For future use, the algorithm should be verified and tested more thoroughly. This will also be described in section 7.

7

Future Work

In this chapter, the work that has yet to be done is evaluated. In Section 7.1, the needed further adaption of Sector X is treated and in Section 7.2, future steps to take for the trajectory predictor are evaluated. In Section 7.3 the design and testing of the TTB display will be handled and in Section 7.4, the project planning will be given.

7.1. Finalising Sector X Adaption

As a first step, Sector X still needs some updates before real life TBS ARR scenarios can be simulated.

First of all, the RECAT separation values need to be implemented. To achieve this, new aircraft classes need to be added, since Sector X currently only works with 'Light', 'Medium' and 'Heavy' aircraft. Next, the separation values need to be added to the algorithm, such that insufficient separation based on RECAT separation values can be detected. Then, TBS alerts need to be implemented such that insufficient Time-Based separation can be detected.

Next, wind needs to be implemented in the simulator. The wind will need to interact with aircraft trajectories. At first, a uniform wind will be used. If enough time is available, a more detailed wind model may be implemented to increase simulator fidelity.

7.2. Update Trajectory Prediction

The trajectory prediction needs to be finalised. Its robustness must be tested with aircraft coming from all directions of the sector. After this, wind must be implemented in the prediction.

The integration of wind into the prediction poses a problem. For uniform wind, analytic analysis can be used. However, when an advanced wind model is used where the wind is not uniform, integration must be used. This poses a problem for the trajectory prediction since it is now impossible to analytically derive where an aircraft should turn to end up at a certain point in space. Hence, instead of a start-to-finish approach, as described in 6, a finish-to-start approach can be used. This would mean rewriting the trajectory predictor.

After finalising the trajectory predictor, it must be verified. First, some scenarios will be executed to check if the time estimation comes near the time it takes for the aircraft to traverse this scenario, in real time. After these real time simulations, using analytic analysis in Matlab, the model will be verified thoroughly.

7.3. Display Design and Testing

When the trajectory prediction is verified and Sector X finalised, the TTB possibilities display can be built. Using the shader program, a program that allows the user to perform calculations using the video-card in their computer to speed up calculations, the area where aircraft can safely turn to base will be plotted. This will be done by evaluating the estimated arrival time between the leading aircraft and the following aircraft, taking each pixel as a TTB possibility.

After the TTB display is finalised, it will be tested in real time. Participants will be taken either from the LVNL or from the TU Delft. The results will be analysed and a paper will be written.

7.4. Project Planning

For this project, the planning will be as follows. After the hand-in of this preliminary report, 7 weeks will be taken for adapting and finalising Sector X. 6 weeks will be taken for programming the TTB display and integrating it into the Sector X design. In 5 weeks, the experiment will be set up, conducted and the results will be analysed. In the last 6 weeks, the final thesis will be written and the final presentation given. For a detailed Gantt chart of the project planning, see Appendix D.

8

Concluding Remarks

In this chapter, some concluding remarks will be given. A conclusion is given in Section 8.1. Preliminary answers to the research question and subquestions will be given in Section 8.2.

8.1. Concluding Remarks

In this report, the work domain of an ARR controller has been extensively researched. The problem of TBS w.r.t. DBS has been treated. A problem that will most likely occur with TBS is that the gains in runway throughput will be less than the theoretically predicted gains. This is why a display was needed to assist the controller in obtaining optimum TBS.

Concepts were derived to assist the ARR in obtaining optimum TBS. Concepts included displays showing the TBS separation minima around the aircraft with several variations, and a display assisting the ARR controller in deciding where to let an aircraft turn to base. It was chosen to develop this display further.

Next to this, some preliminary work was done on an existing Rapid Prototyping Environment (RPE), developed at the TU Delft for ATC simulations. The simulator was adapted for arrival scenarios; an ILS object was implemented and a function for aircraft to catch and land on the ILS was made. An initial trajectory predictor was devised based on nominal arrival paths, obtained from expert interviews. This trajectory predictor has yet to be verified.

In the coming phase of this research, the RPE will be finalised. Wind will be integrated and TBS enabled. The trajectory predictor will also be adapted to account for wind in the sector. Next, the TTB display will be developed and designed This will be an iterative process including input from ARR controllers at the LVNL. At the end of this research, the TTB display will be tested and results reported.

The hypothesis is that operators using the TTB display will pick a strategy sooner than they would before. That is, as soon as an aircraft enters the sector, the controller can immediately pick a path for the aircraft such that it will be separated for its entire journey to the runway threshold. It is also expected that the implementation of the TTB display leads to fewer controller interventions, since last minute corrections will become redundant. These hypotheses will be tested in the tests that will be conducted.

8.2. Preliminary Answers to Research Question

The research question at hand was:

'How can the Human Machine Interface for an Air Traffic Arrival Controller be improved to best assist the controller in obtaining optimum time-based separation between aircraft during the final approach phase?'

In this report, it became evident that there are a number of ways in which the ARR controller can be assisted. The ARR controller can be assisted using a simple approach of a ghost aircraft with a TBS marker projection, as is the case for Heathrow Airport, see section 3.2. However, this would only assist the controller at the end of the final approach and it would not show the controller the options in the entire locomotion state space of the work domain.

To achieve this, a cognitive work analysis has been performed. Using this analysis, along with the earlier work domain analysis, multiple concepts have been designed, as can be seen in chapter 5. Finally, one concept was chosen, the TTB display, and designed in more detail. This research will continue by further developing the TTb display.

Subquestions for this research were:

1. What are current methods and best practices to maintain safe separation during the final approach phase?
2. How is optimum time-based separation defined?
3. What problems arise when using time-based separation instead of distance-based separation, and what information does the controller need to solve these problems?
4. How is this information best visualised?

Current methods to maintain separation are described in section 2.5. ARR controllers can give heading, speed and altitude commands. Using these commands, the controllers are able to increase or decrease separation by either altering the speed of an aircraft or revising its path, as described in section 4.4.

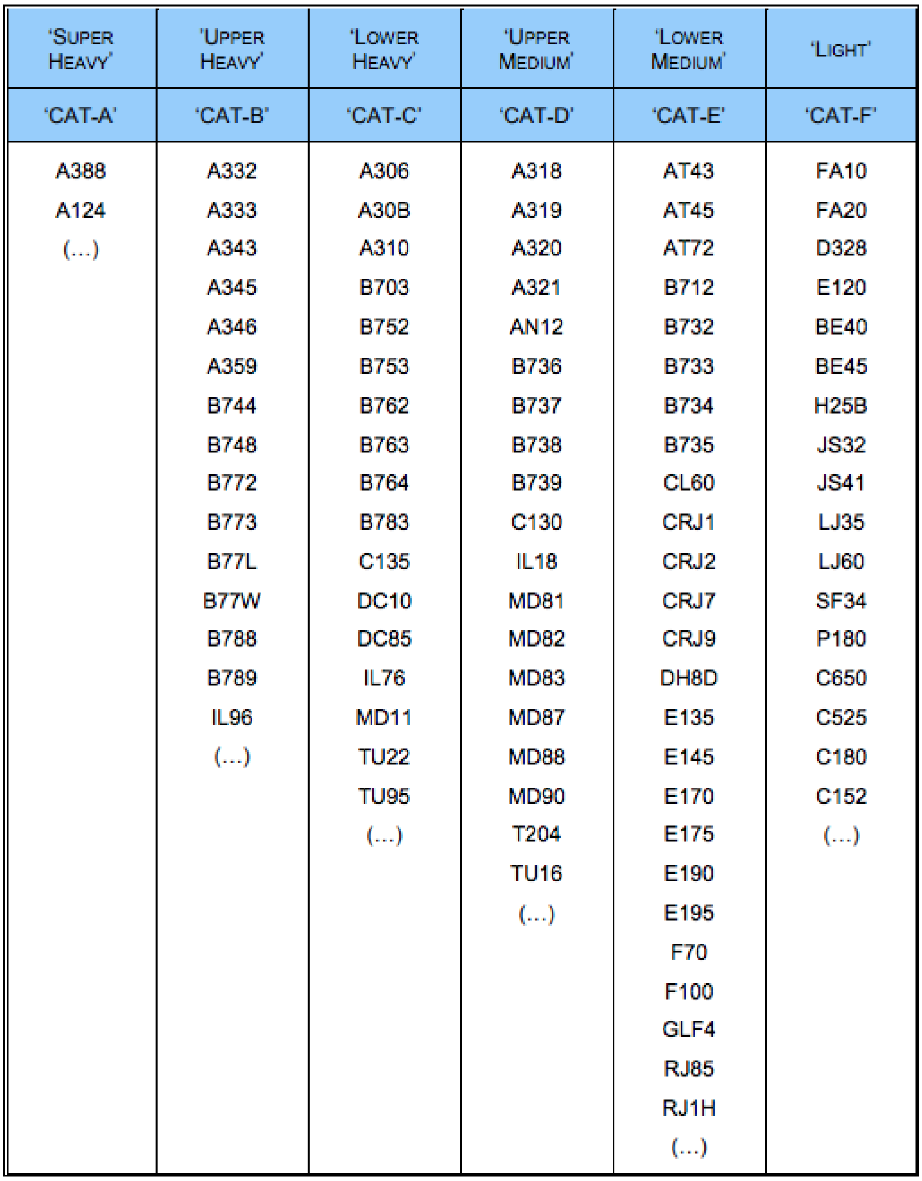
Optimum TBS is defined as the TBS where landing rate is maximum. As such, the controller needs to be assisted in devising strategies where separation is almost exactly at the minimum TBS.

Problems that arise when using TBS are the invisibility of time on the current radar display in use by ATCos. The concepts presented in this report aimed to increase visibility of the TBS in some way. What information is exactly needed will become evident in the testing phase of the research, this is an iterative process. This also goes for the visualisation; at the moment it is not yet known what visualisation will work. This will hopefully become clear in the design part of this thesis.

A

Aircraft in RECAT-EU Categories

Taken from [8].



'Super Heavy'	'Upper Heavy'	'Lower Heavy'	'Upper Medium'	'Lower Medium'	'Light'
'CAT-A'	'CAT-B'	'CAT-C'	'CAT-D'	'CAT-E'	'CAT-F'
A388	A332	A306	A318	AT43	FA10
A124	A333	A30B	A319	AT45	FA20
(…)	A343	A310	A320	AT72	D328
	A345	B703	A321	B712	E120
	A346	B752	AN12	B732	BE40
	A359	B753	B736	B733	BE45
	B744	B762	B737	B734	H25B
	B748	B763	B738	B735	JS32
	B772	B764	B739	CL60	JS41
	B773	B783	C130	CRJ1	LJ35
	B77L	C135	IL18	CRJ2	LJ60
	B77W	DC10	MD81	CRJ7	SF34
	B788	DC85	MD82	CRJ9	P180
	B789	IL76	MD83	DH8D	C650
	IL96	MD11	MD87	E135	C525
	(…)	TU22	MD88	E145	C180
		TU95	MD90	E170	C152
		(…)	T204	E175	(…)
			TU16	E190	
			(…)	E195	
				F70	
				F100	
				GLF4	
				RJ85	
				RJ1H	
				(…)	

B

Airspeed Definitions

The airspeed definitions in this appendix are taken from [31].

B.1. Indicated Airspeed

The Indicated Airspeed (IAS) is the airspeed which the pilot sees on the airspeed indicator. It is derived from comparing the total pressure and the static pressure. Total and static pressure can be measured using a Pitot tube. The IAS can then be derived using Bernoulli's equation, given in equation B.1.

$$p_t = p_s + \frac{1}{2}\rho V_{IAS}^2 \tag{B.1}$$

Where p_t is the total pressure in Pascal and p_s is the static pressure in Pascal. Subsequently, the relation between the IAS and the static pressure and air density at sea level, p_0 and ρ_0, is given by equation B.2. Note that in this equation, the term $p_t - p_s$ can be seen as the impact pressure as measured by a pitot tube.

$$V_{IAS} = \sqrt{7\frac{p_0}{\rho_0}\left[\left(\frac{p_t - p_s}{p_0} + 1\right)^{\frac{1}{3.5}} - 1\right]} \tag{B.2}$$

B.2. Calibrated Airspeed

When the IAS is corrected for position and instrument error, it is called the Calibrated Airspeed, or CAS. Since these errors are mostly small, it can be assumed that the CAS is equal to the IAS, equation B.3.

$$V_{CAS} \approx V_{IAS} \tag{B.3}$$

B.3. Equivalent Airspeed

The Equivalent Airspeed (EAS) is the CAS corrected for altitude. It is thus the airspeed at which the aircraft would fly, would it fly at sea level. The EAS can be derived using equation B.4.

$$V_{EAS} \cong \sqrt{7\frac{p_s}{\rho_0}\left[\left(\frac{p_t - p_s}{p_s} + 1\right)^{\frac{1}{3.5}} - 1\right]} \tag{B.4}$$

B.4. True Airspeed

The True Airspeed (TAS) is the airspeed of the aircraft relative to the atmopsphere it is flying in. The TAS can be derived from the EAS by simply correcting for the density, like in equation B.5. The TAS can be derived from the IAS by using equation B.6.

$$V_{EAS} = V_{TAS}\sqrt{\frac{\rho}{\rho_0}} \tag{B.5}$$

$$V_{TAS} = \sqrt{\frac{2\gamma}{\gamma-1}\frac{p}{\rho}\left((1+X)^{\frac{\gamma-1}{\gamma}}-1\right)} \qquad \text{with} \qquad X = \frac{p_0}{p}\left[\left(1+\frac{\gamma-1}{2\gamma}\frac{\rho_0}{p_0}V_{IAS}^2\right)^{\frac{\gamma}{\gamma-1}}-1\right] \tag{B.6}$$

B.5. Groundspeed

The groundspeed (GS) is the speed of the aircraft's projection on the ground. The only difference between the TAS and the GS is the wind speed at the aircraft location. The wind speed vector can be added to the TAS to obtain the GS, as in equation B.7.

$$V_{GS} = V_{TAS} + V_{wind} \tag{B.7}$$

C

Trajectory Prediction in Sector X: Leg Estimation

In Sector X, booleans were set up such that the leg of flight could be identified by a unique combination of these booleans. The booleans that were used along with the conditions for them to be true can be found in table C.1. Which booleans are used to identify each leg of flight can be seen in table C.2. Note that boolean notation is used; the AND opreator requires both booleans before and after it to be true, the OR operator requires one of either booleans to be true and an exclamation mark in front of a boolean requires the boolean to be false instead of true.

In the algorithm, the legs are checked in order from 1 to 7. This allows the algorithm to compare the leg that is being evaluated with only legs that are after that leg. For instance, if the algorithm tests if the aircraft is in leg 4, the possibility of the aircraft being in legs 1 - 3 is already eliminated, and the algorithm only has to compare the situation of the aircraft with legs 4 - 7. The algorithm stops evaluating legs when an aircraft is identified to be in a leg. In table C.2 one can see the effect of this type of evaluation. For leg 4, only booleans 'turning' and '!caughtILS' have to be true, since the very fact that the possibility of the aircraft being in leg 4 is being evaluated already implies that the aircraft is not in leg 1 - 3. Hence the algorithm only needs to test if the aircraft is turning ('turning' boolean) and if the aircraft is not in leg 6 ('!caughtILS' boolean). Similarly, for leg 6 the algorithm only has to evaluate if the aircraft is in a turn to know it is in leg 6.

Table C.1: Explanation of booleans used for identifying phase of arrival flight

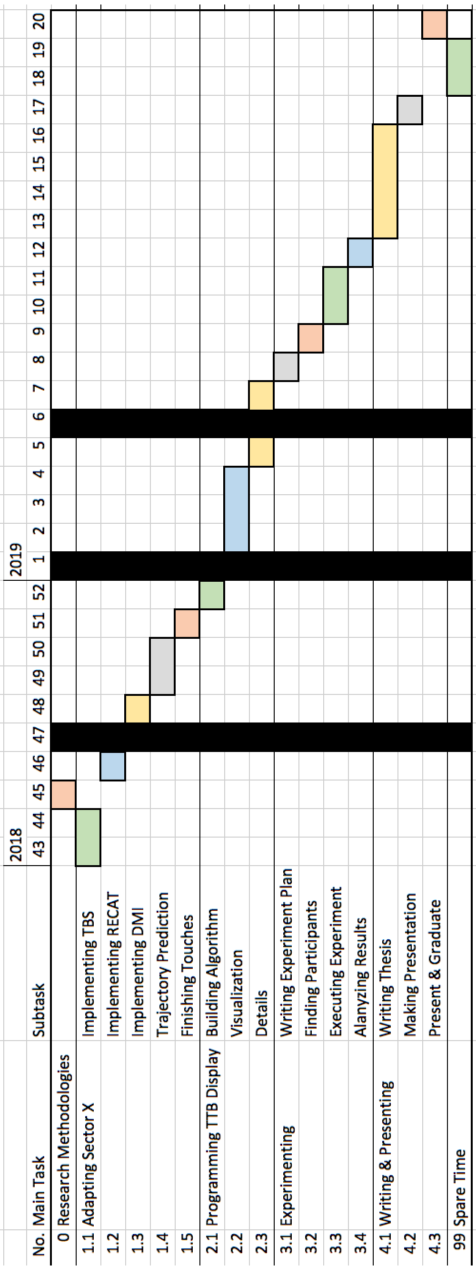
Boolean	**Explanation**
turning	The aircraft target heading is not equal to the current heading, indicating the aircraft is executing a turn
behindTTBPoint	Aircraft is behind the ideal turn to base point, when the aircraft would turn to base it will cross the ILS at more than 10 NM from the runway threshold
turnedToBase	The aircraft speed vector points toward the ILS, where the ILS extends 20 NM from the runway threshold
hasCaughtILS	The aircraft has received the 'catch ILS' command and crossed its ideal turn in point
closeToILS	The distance to the ILS is less than a certain predefined value. For this simulation, 3 NM is used, since data pointed out that most aircraft turn into the ILS (start leg 4) between 3 and 1 NM from the ILS.
hasTurnedToILS	The angle between the aircraft speed vector and the ILS is less than 30 degrees and the aircraft is close to the ILS (less than 3 NM)

Table C.2: Booleans that are used to identify a flight leg, per leg

Leg	**Booleans**
1	!turning AND !behindTTBPoint AND !turnedToBase AND !caughtILS
2	turning AND !closeToILS AND !hasCaughtILS
3	!turning AND hasTurnedToBase AND !closeToILS AND !hasCaughtILS
4	turning AND !caughtILS
5	!turning AND !caughtILS
6	turning
7	- (Last option, if leg 7 is evaluated, the AC is in leg 7)

D

Planning Gantt Chart



			2018										2019																			
No.	Main Task	Subtask	43	44	45	46	47	48	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	Research Methodologies				🟧		⬛						⬛					⬛														
1.1	Adapting Sector X	Implementing TBS	🟩	🟩			⬛						⬛					⬛														
1.2		Implementing RECAT				🟦	⬛						⬛					⬛														
1.3		Implementing DMI					⬛	🟨					⬛					⬛														
1.4		Trajectory Prediction					⬛		⬜	⬜			⬛					⬛														
1.5		Finishing Touches					⬛				🟧		⬛					⬛														
2.1	Programming TTB Display	Building Algorithm					⬛					🟩	⬛					⬛														
2.2		Visualization					⬛						⬛	🟦	🟦	🟦		⬛														
2.3		Details					⬛						⬛				🟨	⬛	🟨													
3.1	Experimenting	Writing Experiment Plan					⬛						⬛					⬛		⬜												
3.2		Finding Participants					⬛						⬛					⬛			🟧											
3.3		Executing Experiment					⬛						⬛					⬛				🟩	🟩									
3.4		Alanyzing Results					⬛						⬛					⬛						🟦								
4.1	Writing & Presenting	Writing Thesis					⬛						⬛					⬛							🟨	🟨	🟨	🟨				
4.2		Making Presentation					⬛						⬛					⬛											⬜			
4.3		Present & Graduate					⬛						⬛					⬛														🟧
99	Spare Time						⬛						⬛					⬛												🟩	🟩	

Bibliography

[1] National Air Traffic Services. Time based separation, 2018. URL `https://www.nats.aero/tbs/`.

[2] To70 Aviation Consultants. kdc - time based separation, presentatie aan kdc groot.

[3] NATS. Enhanced time based separation (etbs), 2017. URL `http://www.nats.aero/wp-content/uploads/2017/03/Full_eTBS_PresentationV1.pdf`.

[4] EUROCONTROL. Time-based separation (tbs) solution and controller tool for final approch, 2016. URL `https://www.eurocontrol.int/sites/default/files/publication/files/time-based-separation-factsheet-2016.pdf`.

[5] Eurocontrol. Fir/uir lower chart europe, 2018. URL `http://www.eurocontrol.int/sites/default/files/content/documents/nm/cartography/04012018-firuir-lower-airspace-ectl.pdf`.

[6] AIS Netherlands. Electronic ais publications, 2018. URL `http://www.ais-netherlands.nl/aim/2018-08-02-AIRAC/html/index-en-GB.html`.

[7] International Civil Aviation Organization. Wake turbulence aspects of Airbus A380-800 aircraft. 2008.

[8] Eurocontrol. RECAT-EU, European Wake Turbulence Categorisation and Separation Minima on Approach and Departure. 2018.

[9] International Civil Aviation Organization. ICAO doc 4444, fourteenth edition. 2001.

[10] Julien Scavini. B767familyv1.0.png, 2011. URL `https://commons.wikimedia.org/wiki/File:B767FAMILYv1.0.png`.

[11] Julien Scavini. A340family2v1.0.png, 2012. URL `https://commons.wikimedia.org/wiki/File:B767FAMILYv1.0.png`.

[12] Gooogle LLC. Google maps, 2018. URL `https://www.google.nl/maps/`.

[13] Onderzoeksraad voor Veiligheid. Neergestort tijdens nadering, boeing 737-800, nabij amsterdam schiphol airport, 2009. URL `https://www.onderzoeksraad.nl/uploads/items-docs/714/Rapport_TA_NL_aangepast.pdf?s=B66BB60C83923D122FB3CB4B7A403F90A088792C`.

[14] National Air Traffic Services. Time-based separation at heathrow, 2018. URL `https://nats.aero/blog/2014/02/explaining-time-based-separation-heathrow/`.

[15] Clark Borst, John M. Flach, and Joost Ellerbroek. Beyond ecological interface design: Lessons from concerns and misconceptions. *IEEE Transactions on Human-Machine Systems*, 45(2):164–175, 2015. ISSN 21682291. doi: 10.1109/THMS.2014.2364984.

[16] Jens Rasmussen and Kim J. Vicente. Coping with Human Errors through System-Design - Implications for Ecological Interface Design. *International Journal of Man-Machine Studies*, 31(5):517–534, 1989. ISSN 00207373. doi: 10.1016/0020-7373(89)90014-X.

[17] Jens Rasmussen. Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13(3):257–266, 1983. ISSN 21682909. doi: 10.1109/TSMC.1983.6313160.

[18] Neelam Naikar. *Work Domain Analysis, Concepts, Guidelines and Cases*. CRC Press, Taylor & Francis Group, LLC, Boca Raton, USA, 2013.

[19] Marinus M. Van Paassen, C. Borst, J. Ellerbroek., M. Mulder, and J. M. Flach. Ecological Interface Design for Vehicle Locomotion Control. *IEEE Transactions on Human-machine systems*, X(Xx):1–16, 2018. ISSN 2168-2291. doi: 10.1109/THMS.2018.2860601.

[20] Brian Hilburn, Carl Westin, and Clark Borst. Will Controllers Accept a Machine That Thinks Like They Think ? The Role of Strategic Conformance in Decision Aiding Automation. *Air Traffic Control Quarterly*, 22(2):115–136, 2014. ISSN 1064-3818. doi: 10.2514/atcq.22.2.115.

[21] Carl Westin, Clark Borst, and Brian Hilburn. Strategic Conformance: Overcoming Acceptance Issues of Decision Aiding Automation? *IEEE Transactions on Human-Machine Systems*, 46(1):41–52, 2016. ISSN 21682291. doi: 10.1109/THMS.2015.2482480.

[22] Kim J. Vicente and Jens Rasmussen. Ecological Interface Design: Theoretical Foundations. 22(4):589–606, 1992. ISSN 21682909. doi: 10.1109/21.156574.

[23] Kim J. Vincente. *Cognitive Work Analysis, Toward Safe, Productive, and Healthy Computer-Based Work*. CRC Press, Taylor & Francis Group, LLC, Boca Raton, USA, 1999.

[24] Ann M. Bisantz. *Applications of Cognitive Work Analysis*. CRC Press, Taylor & Francis Group, LLC, Boca Raton, USA, 2009.

[25] Paul Macwilliams and Donald Porter. An Assessment of a Controller Aid for Merging and Sequencing Traffic on Performance-Based Arrival Routes. 2007.

[26] D. Anand and The MITRE Corporation. Display aid for air traffic controllers, 1989. URL `https://patentimages.storage.googleapis.com/c8/67/12/8c264ba1165faa/US4890232.pdf`. US Patent 4,890,232.

[27] Arthur P. Smith and Thomas A. Becher. A study of SPACR ghost dynamics applied to RNAV routes in the terminal Area. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, 1(January 2005), 2005. doi: 10.1109/DASC.2005.1563330.

[28] Paul V. MacWilliams, Arthur P. Smith, and Thomas A. Becher. RNP RNAV arrival route coordination. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, 2006. doi: 10.1109/DASC.2006.313682.

[29] Ludovic Boursier, Bruno Favennec, Eric Hoffman, Aymeric Trzmiel, François Vergne, and Karim Zeghal. Merging Arrival Flows Without Heading Instructions. *USA/Europe Air Traffic Management Research and Development Seminar*, (July):1–8, 2007.

[30] Rolf Klomp, Clark Borst, René Van Paassen, and Max Mulder. Expertise Level, Control Strategies, and Robustness in Future Air Traffic Control Decision Aiding. *IEEE Transactions on Human-Machine Systems*, 46(2):255–266, 2016. ISSN 21682291. doi: 10.1109/THMS.2015.2417535.

[31] Prof. Dr. Mustafa Cavcar. *Airspeed Definitions*. School of Civil Aviation, Anadolu University, Eskisehir, Turkey, 2004.

III

Part 3: Book of Appendices

1

Experiment Manual

Dear Participant,

First of all I would like to thank you for taking part in this experiment. The experiment will take approximately 3 hours to complete. First, you will be given time to read this manual and practice some scenarios, which will take about 2 hours. Afterwards, you will conduct 4 measurement runs of 15 minutes each. An approximate time schedule of the experiment can be found in table 1.1. After each (practice) scenario, you are asked to fill out a workload rating form. At the end of this test, you are asked to complete a small survey about the experiment and the proposed display.

Table 1.1: Time Schedule of This Experiment

Time (H:M)	Activity
0:00	Reading: Approach Control in Sector X
0:10	Practice Run 1: Approach Control in Sector X
0:15	Reading: Approach Control Guidelines
0:20	Practice Run 2: Approach Control in Sector X
0:30	Reading: The NATS Display
0:45	Practice Run 3: The NATS Display
1:00	Practice Run 4: The NATS Display
1:15	Reading: The Ideal Turn-In Point Display
1:30	Practice Run 5: The Ideal Turn-In Point Display
1:45	Practice Run 6: The Ideal Turn-In Point Display
2:00	Test Run 1
2:15	Test Run 2
2:30	Test Run 3
2:45	Test Run 4
3:00	Survey

1.1. Test Setup

In this experiment, you will test approach (APP) scenarios with different display tools. Aircraft will fly into the sector and you, as the APP Controller, will have to guide them toward the runway, using the ILS. The Area Control (ACC) simulator, Sector X, has been adapted for APP control. In Section 1.2.3, the differences between the APP simulator and the ACC simulator which you are used to will be explained. During this section, you will be given two short practice scenarios to get used to the APP version of Sector X. After these quick lessons, two displays will be presented; the NATS display (Sections 1.3 and 1.4). After each display presentation you will be given two practice scenarios to get used to that display. At the end of this manual, in Section 1.5, some final tips and tricks will be given, which you can use during your test runs. After you have read this manual, you will execute 4 test runs of 15 minutes each, concluding the experiment.

Please note that for illustration purposes, the colours for the pictures in this manual are different from the simulator. In the simulator, the background is black and all aircraft are green; in print this results in unclear images. Also note that for the images in this manual, an unusually strong wind is present. This is done in order to show the effects of wind, in the simulator these effects will be less distinct.

1.2. Approach Control in Sector X

In this section, the differences between Area Control (ACC) and Approach Control (APP) will be explained. First, the goal of an APP scenario will be given. Second, the differences in simulator control will be laid out after which some guidelines for APP control will be given.

1.2.1. Goal of the Simulation

The goal of an APP scenario differs from the goal in an ACC scenario. In an APP scenario, the goal is to guide every aircraft towards the runway threshold in a **safe and efficient** manner. For the controller, this means achieving the **largest possible throughput** (amount of aircraft landing per unit of time), whilst **maintaining separation**. Next to this, some guidelines are defined, which an APP Controller (and you in this experiment) must try to adhere to.

1.2.2. Difference in Aircraft Symbol and Label

In Figure 1.1, an aircraft as visualised in the simulator is given. The square in the middle depicts the aircraft location. The dots behind the aircraft are the so called ’history dots’; these dots visualise the track which the aircraft has taken. The circle around the aircraft has a radius of 1.5 NM, this area can be used to assess separation, further explained in figure 1.2.6.

The label next to the aircraft contains some vital information, as can be seen in figure 1.2. In the first line of the label, the aircraft ID (ACID) is given. This is a unique name identifying the aircraft. On the second line, the altitude is given in flight levels (FL) and the Indicated Airspeed (IAS) is given in kts. On the bottom line of the label, the heading (HDG) in degrees is given and next to it the type of the aircraft. In the scenarios you will perform, only two types of aircraft will be present, the Medium (M) and the Heavy (H) type.

Please note that you can click and drag the aircraft labels to move them to a more convenient location on the screen.

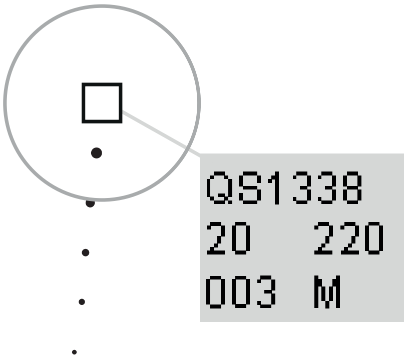


Figure 1.1: An aircraft with label as visualised in the simulator

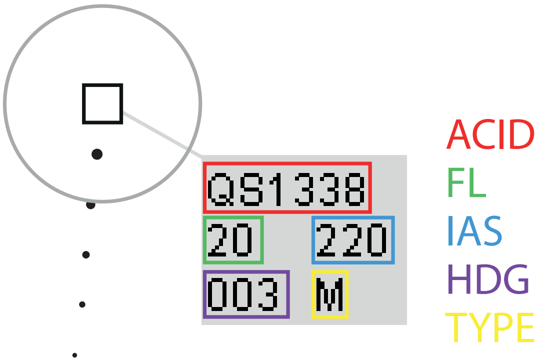


Figure 1.2: An aircraft with label as visualised in the simulator, with label legend

1.2.3. Difference in Simulator Control

The control of the Sector X version adapted for Approach operations (APP-Sector X) is very similar to the version for Area control (ACC). Aircraft can be given commands using the Control Display Unit (CDU) screen, depicted in Figure 1.3. Altitude (EFL), Heading (HDG) and Speed (SPD) commands can be given by selecting an aircraft, clicking the corresponding command button, inputting the numerical value of your command using the number buttons and clicking the execute (EXQ) button. Using the Clear (CLR) button, one can delete messages put into the CDU. The ’DCT’, ’TOC’ and ’PRV’ buttons can be ignored for the scenarios in this experiment. Commands can be previewed in the preview section on the bottom of the CDU screen; the ACID of the selected aircraft will be displayed along with the commands that are not yet executed.

In comparison to the Sector X version for ACC control, there are two main additions in the APP version: the Instrument Landing System (ILS) in the simulator area and the ’ILS’-button in the Command unit.

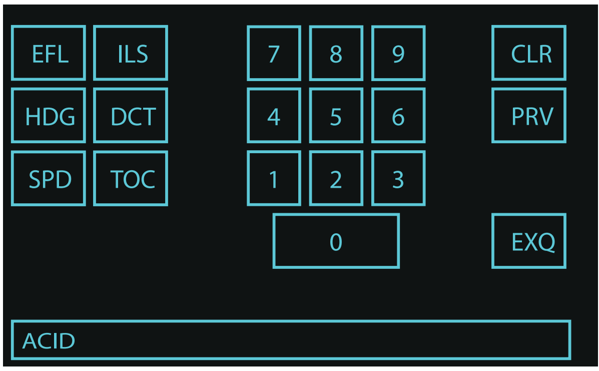


Figure 1.3: The CDU Screen with an Active ILS Button

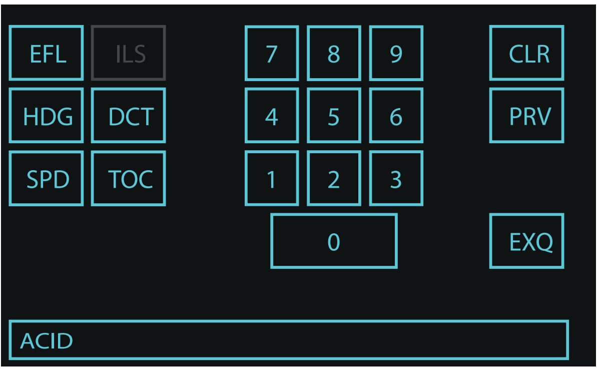


Figure 1.4: The CDU Screen with an Inactive ILS Button

In reality, the ILS is a radio beacon stretching from the runway threshold to 25 NM before the runway threshold, at an upwards angle of 3 degrees. The ILS consists of a Localizer (LOC) to guide the aircraft in lateral direction and a Glideslope (G/S) for vertical guidance.

The ILS in the simulator is modelled after an ILS as visualised at the NLR (Air Traffic Control The Netherlands), as can be seen in Figure 1.5. The ILS is visualised as a green line. There is a green dot at every 2 NM of the ILS, as well as a green orthogonal line at every 10 NM. These dots can be used to get a sense of scale in the simulation. Aircraft in the simulation can intercept the ILS to initiate an automatic instrument landing manoeuvre. The Approach controller must first clear the aircraft for ILS interception. In the simulator, this is done using the ’ILS’-button in the command panel.

To be able to catch the ILS, the aircraft must first get permission to do so from the controller. In the simulator, this is done by selecting an aircraft and subsequently clicking the ’ILS’-button in the CDU. When an aircraft is selected, the button shows the ILS status; when it is active the aircraft is cleared to catch the ILS, when it is inactive the aircraft is not cleared to catch the ILS. In figure 1.3, the CDU is shown with an active ILS button, whereas in Figure 1.4 it is inactive. The ’ILS"-button can be used as a switch; clearance can be given by making the button active and withdrawn by making it inactive.

It is important to note that aircraft can only intercept the LOC if it has a heading angle of 30 degrees or less w.r.t. the LOC. This is irrespective of the flight path angle. This is illustrated in Figure1.6 Furthermore, the aircraft can only intercept the G/S when it is approached from below. For an aircraft flying at FL20, which is the standard altitude at which to intercept the ILS, this means that the aircraft must catch the ILS at a minimum distance of 6.3 NM from the runway threshold.

Once the ILS is caught by an aircraft, the aircraft will automatically follow the ILS path. This means that it will adjust its heading to stay on the LOC, such that drift due to wind is counteracted. Furthermore, the aircraft will descend automatically once it catches the G/S (at 6.3 NM from the runway threshold for an aircraft flying at FL20). In real life, aircraft are free to decelerate to their own landing speed starting at 4 NM before the

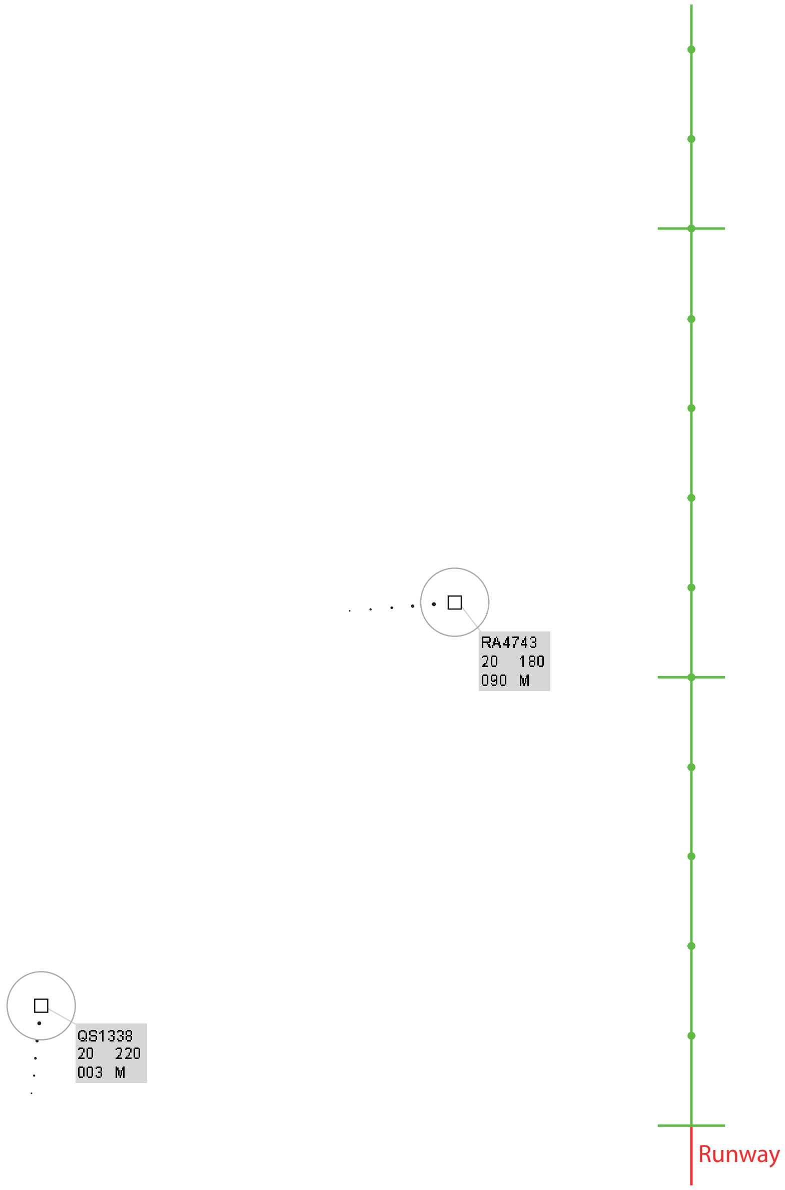


Figure 1.5: The ILS as Visualised in the Simulator. The runway is given in red; the runway is not visualised in the simulator.

runway threshold (also called 4 Distance Measuring Equipment or 4DME). In the simulator, this is modelled as all aircraft automatically decelerating towards 130 kts starting at 4DME.

1.2.4. Practice Scenario 1

Please now complete the first practice scenario. In this scenario, you will be given 5 minutes to steer a few aircraft onto the ILS. Note that the scenario is sped up. For this scenario, the focus should lie on practising steering aircraft onto the ILS and getting a feel of where one can steer the aircraft into the ILS.

The steps for steering an aircraft onto the ILS are summarised:

1. Send aircraft toward the ILS
2. When at sufficient proximity to the ILS, steer the aircraft into the ILS with an angle of incidence of 30 degrees or less (for aircraft approaching from the right this means a heading angle of 150 degrees or more, for aircraft approaching from the left this corresponds to a heading angle of 210 degrees or less.)

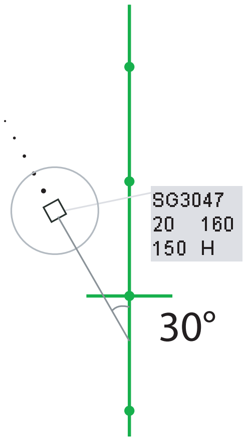


Figure 1.6: Detail of Aircraft catching the ILS at 30 degrees

3. Clear the aircraft to catch the ILS using the 'ILS'-button.

1.2.5. Approach Controller Guidelines

For approach controllers, some guidelines are defined by the LVNL. These guidelines are taught by the LVNL to new APP controller recruits and it is expected that APP controllers (and you in this experiment) try to follow these guidelines as much as possible. These guidelines relate to the attributes path, speed and altitude.

In figure 1.7, an aircraft with a possible approach path is given. Note that the approach path an aircraft may take may look very different depending on the point where the aircraft enters the sector, the points at which they turn and the angles with which they turn. However, an approach path can almost always be divided into 7 parts, or legs. These leg are depicted in Figure 1.7. In leg 1, the aircraft travels alongside the ILS. In this leg, the heading vector does not point towards the ILS, the aircraft is travelling towards the point at which it will turn towards the ILS. Turning toward the ILS is called 'Turning to Base', and the point at which the aircraft starts this turn is called the Turn To Base (TTB) Point. The TTB Point is the start of leg 2, which is the leg in which the aircraft is turning. As soon as the aircraft has completed its turn it is said to be in leg 3, which consists of straight flight toward the ILS. Leg 4 starts as soon the aircraft turns into the ILS, with a target heading angle of 30 degrees or less with respect to the ILS. The point at which the aircraft turns into the ILS is called the Turn-To ILS (TTI) Point. Leg 5 starts as soon as the aircraft stops turning and is cleared to catch the ILS. When the aircraft catches the ILS beacon and starts to turn automatically, it is said to be in leg 6 and leg 7 starts when the aircraft has successfully caught the ILS and starts its instrument landing procedure. Note that leg 1 and 2 may be skipped when an aircraft enters the sector while directly heading for the ILS, in which case the aircraft is in leg 3 from the beginning. Also note that an aircraft being in a leg of flight does not impose any restrictions on it; the Approach Controller can change the path of an aircraft as he/she sees fit.

The guidelines concerning flight speeds are linked to the legs of flight. From leg 1 onward, it is considered to be best for a controller to command the aircraft to fly at a speed no higher than 220 kts. From leg 2 onward this speed is 180 kts and from leg 7 onward this speed is 160 kts. For you as a participant this simply means giving an aircraft a speed command of 220 kts somewhere in leg 1, giving a speed command of 180 at the same time as commanding it to turn to base and giving it a speed command of 160 kts the moment it is on the ILS. Of course, these speed commands should only be given if the aircraft is flying at a higher speed at that moment. Note that you are free to vary the speed as you see fit, however, since this is an approach scenario, **speed increase commands should be avoided at all cost** since the aircraft needs to decrease its total potential and kinetic energy.

The guidelines for altitude are not of primary importance for this experiment. To simplify the experiment, all aircraft enter the sector at FL20. Since FL20 is considered to be the best practice to catch the ILS, giving altitude commands will not be necessary in this experiment. However, you are free to vary the altitude of aircraft, for instance to ensure separation, see section 1.2.6. Note that, since this is an approach scenario, **altitude increase commands should be avoided at all cost** since the aircraft needs to decrease its total po-

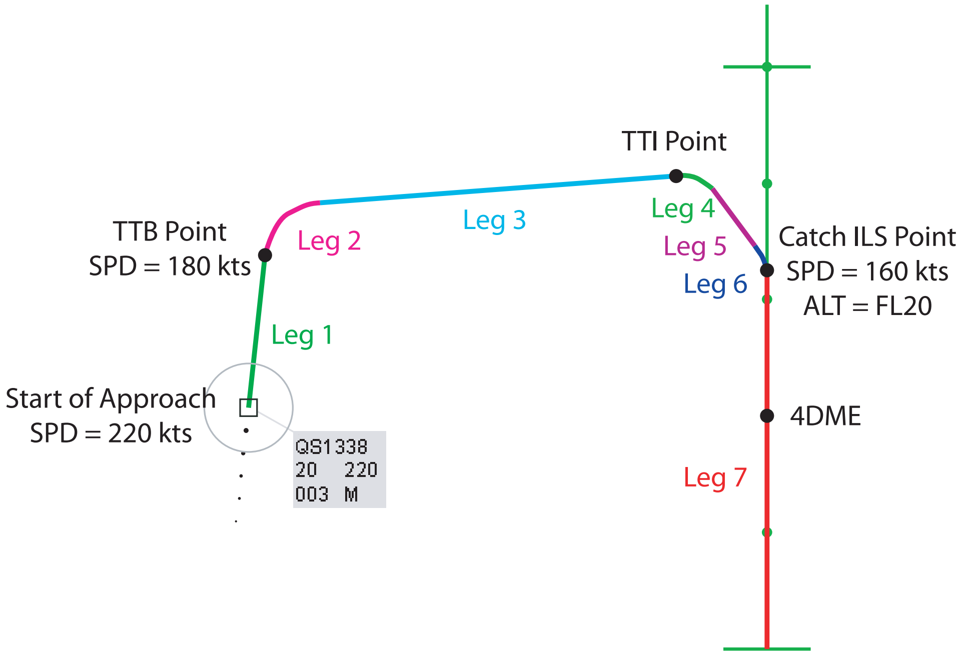


Figure 1.7: A possible Approach path for an aircraft with all its legs of flight

tential and kinetic energy.

1.2.6. Distance-Based Separation

Next to adhering to the guidelines as described in section 1.2.5, the controller must also make sure that all aircraft maintain separation. In a approach scenario, separation is defined as every aircraft keeping a horizontal distance of at least 3 NM from every other aircraft at the same altitude. Aircraft may decrease this separation distance when they are sufficiently vertically separated; this is the case when the aircraft have an altitude difference of at least 10 flight levels.

For your convenience, all aircraft in this experiment have a circular area with a radius of 1.5 NM visualised around them, as can b seen in Figure 1.1. To maintain horizontal separation, simply make sure that these circles never overlap, or if they do, that the respective aircraft are sufficiently separated in altitude.

1.2.7. Practice Scenario 2

Now complete the first practice scenario. In this scenario, aircraft have different IAS settings. Try to catch the ILS whilst maintaining separation and adhering to the guidelines. The scenario takes 10 minutes to complete and is again sped up.

For each aircraft, the steps to be taken are summarised:

1. Set IAS to 220 when the aircraft is in leg 1
2. At the TTB Point, give a heading command and set the IAS to 180
3. At the TTI point, give a heading command and clear the aircraft to catch the ILS
4. As soon as the aircraft is on the ILS, set the IAS to 160

Table 1.2: Time-Based separation minima for different wake pairs, in seconds, for an average IAS of 160 kts with 5 kts headwind.

[NM] **Leader**	**Follower** **Heavy**	 **Medium**	 **Light**
Heavy	90	113	135
Medium	68	68	113
Light	68	68	68

1.3. Time-Based Separation and the NATS Display

1.3.1. Time Based Separation

In addition to the separation as described in section 1.2.6, another separation minimum applies when aircraft fly on the ILS.
When aircraft flies directly behind another aircraft, it has to avoid a conflict with the wake vortices created by the leading aircraft. At most airports, this is done by prescribing a certain distance that a follower aircraft needs to keep from a leading aircraft, depending on the types of both aircraft (Distance-Based Separation of DBS). Here, aircraft can have the type ’Light’ (denoted with a L), ’Medium’ (denoted with an M) and ’Heavy’ (denoted by an H). The general rule is that the lighter the follower aircraft and the heavier the leading aircraft, the larger the distance the follower aircraft needs to keep from the leading aircraft. This is because light aircraft are more influenced by wake vortices and because heavy aircraft create more wake vortices. A follower and leading aircraft are also called a wake pair. Note that the scenarios in this experiment only contain aircraft of type Medium and Heavy, resembling the traffic mix at Schiphol Airport.

However, a problem with DBS arises when large headwinds are present. Since the groundspeed of aircraft is reduced by the headwind, the time it takes for an aircraft to cover the separation distance is increased, reducing the landing rate. This can be solved in part by applying Time-Based Separation (TBS). Instead of keeping a specified distance between every wake pair, with TBS a specific time is kept between the aircraft, mitigating the effects of headwinds on the landing rate. In this experiment, TBS is used.

The separation times for every wake pair are given in table 1.2. **You do not have to learn these separation times** as they will be visualised in the simulator later on, however it is good to understand the influence of type on separation time.

1.3.2. The NATS Display

To assist the Approach controllers in separating aircraft using TBS, the National Air Traffic Services (NATS) in the United Kingdom have developed a display tool. This tool will henceforth be referred to as the NATS Display. A simplified version of this tool was implemented in Sector X. The tool consists of two markers, the TBS Marker and the Optimised Runway Delivery (ORD) Marker. These markers are visualised in Figures 1.8 and 1.9.

The TBS Marker is a marker on the ILS behind an aircraft, where the follower aircraft should always stay behind. If an aircraft crosses the TBS Marker of the aircraft before it, this counts as a loss of separation. The algorithm behind the display predicts where an aircraft will catch the ILS and plots the TBS marker on the ILS, taking the time it will take the aircraft to get onto the ILS into account. Hence, it is possible that an aircraft that is in leg 3 already has a TBS marker for its follower aircraft visualised on the ILS, behind its projected location on the ILS. To calculate the distance behind the leading aircraft of this marker, the predicted average groundspeed of the follower aircraft is taken and multiplied by the separation time.

In the simulator, the TBS marker is visualised as a red dot on the ILS, as can be seen in figure 1.8. Here, aircraft RA4743 must stay behind the red TBS marker, caused by the leading aircraft, SG3047. To clarify which marker belongs to which aircraft, the TBS marker which the aircraft should stay behind will be highlighted when hovering over the aircraft with the computer mouse. In this case, hovering over aircraft RA4734 will highlight the red TBS marker.

The second marker that is plotted is called the Optimised Runway Delivery Marker, or ORD, depicted in blue

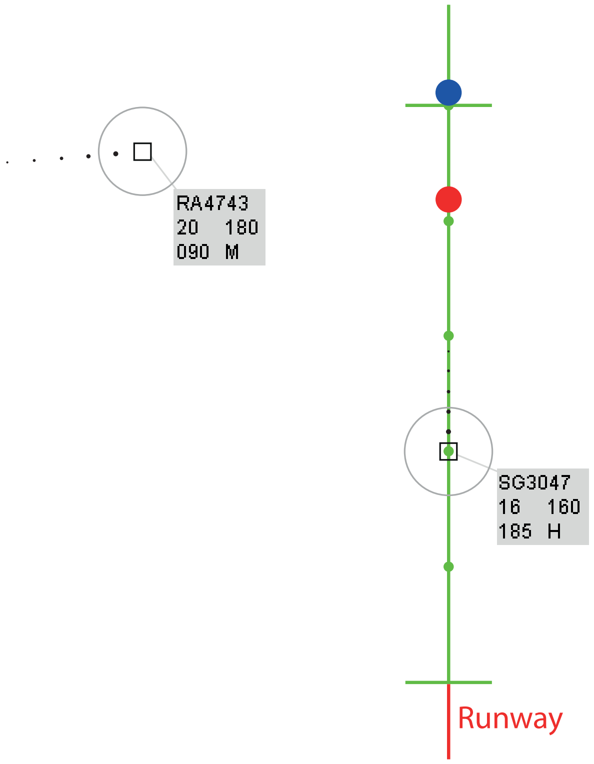


Figure 1.8: The TBS and ORD Marker for an Aircraft Pair

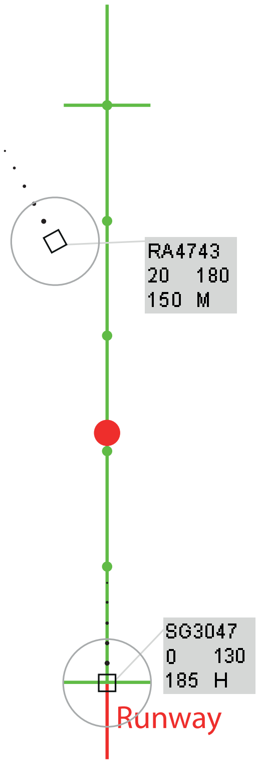


Figure 1.9: The TBS and ORD Marker for an Aircraft Pair at the time the leading aircraft lands

in Figure 1.8. This marker depicts how much the follower aircraft will gain on the leading aircraft. The algorithm sums all differences in groundspeed during the final approach, which results in a distance that will be gained on the leading aircraft if the follower aircraft would proceed with its predicted trajectory and speed profile. Note that here the speed profile as described in the guidelines in section 1.2.5 are used to predict the distance the follower aircraft will gain on the leading aircraft. The idea is that the controller can aim at the ORD marker on the ILS. If the aircraft catches the ILS on the ORD marker of the leading aircraft, the follower aircraft will be on the TBS marker of the leading aircraft as soon as the leading aircraft lands. This is thus the optimal solution, resulting in the highest throughput.

Note that the ORD will thus creep toward the TBS marker, until they are at the same place as can be seen in Figure 1.9. This phenomenon is called the compression effect. If the follower aircraft correctly follows the predicted trajectory and speed profile, it will creep toward the TBS marker in exactly the same manner. This process will start as soon as the groundspeed of the leading aircraft is lower than the groundspeed of the follower aircraft, and stops if these aircraft have the same groundspeed, or if the leading aircraft lands.

In Sector X, the ORD marker is visualised as a blue dot on the ILS, just like the TBS marker. Just like the TBS marker, it will light up when the computer mouse is hovered over the follower aircraft.

1.3.3. Practice Scenario 3 & 4

Please complete practice scenarios 3 & 4. In these scenarios, try to use the ORD and TBS markers to maintain separation whilst maximising throughput. Try to adhere to the guidelines. Note that from now on, wind is present in the simulator.

1.4. The Ideal Turn-In Point Display

As you might have noticed by now, the path an aircraft takes is a significant variable in the time it takes for an aircraft to land. The largest variables in the path an aircraft takes are the points where the aircraft turns toward the ILS (Turn To Base or TTB Point), the point where an aircraft turns into the ILS (Turn To ILS or TTI Point) and the target heading given at these points. Of lesser importance for the arrival time of an aircraft are speed and altitude, since only deceleration and descend commands should ideally be given. A display was made to assist approach controllers in early decision making, taking the path as the variable of interest. The display visualises possible TTB and TTI points, along with possible turn headings, such that aircraft will be safely separated on the ILS. This display will be referred to as the Ideal Turn-In Point Display, or ITIP. The ITIP consists of two versions of the same display; one for the ideal TTB point and one for the ideal TTI point.

For this experiment, the TBS marker from the NATS Display will also be used in the ITIP Display. This is to get a feeling of the required separation of aircraft in real time.

1.4.1. The Ideal TTB Point

When a selected aircraft flies in leg 1, the TTB part of the ITIP Display is shown. The display for a selected aircraft is shown in figure 1.10. Here, Aircraft QS1338 is selected. The main display feature is an array of green and red dots. Each dot represents a future location of the aircraft, if it would maintain its current heading and adhere to the ideal speed profile as described in Section 1.2.5. The timestep for this display is the same as the traffic refresh rate for the simulator: 5 seconds. Each dot in the display represents a Turn To Base possibility. The colour of the point is green when the algorithm predicts that the trajectory resulting from turning to base at this point is feasible. Here, feasible means that the aircraft catches the ILS from below and the aircraft will be separated on final approach from its leading and follower aircraft. A red TTB Point indicates some sort of trouble; either the resulting trajectory will result in loss of separation on final approach or the aircraft has to catch the ILS from above.

The trajectory the aircraft is predicted to take is shown in purple. The algorithm assumes that the user will take the quickest feasible route i.e. it assumes the user to pick a TTB Point such that the total time travelled will be minimised, whilst making sure the trajectory is feasible. The user can override this prediction by clicking on one of the dots. When clicking on a TTB point, the algorithm will update the prediction. Simultaneously, the displays for all other aircraft will be updated with this new information. This way, the user can clarify their plan to the algorithm so that the user may see if and how a certain path change will affect the solution space for other aircraft. If the aircraft travels beyond the predicted turn-in point, the algorithm will assume the user made a mistake and it will assume that the aircraft will turn-in at the next timestep. Hence, it will 'push' the prediction forward.

Next to the predicted route, the heading angle command that should be given at the TTB point is shown. This serves as a reminder to the user. The user can vary this heading angle by hovering over a turn in point and using the scroll wheel to adjust the heading angle. Scrolling upwards will increase the heading angle (clockwise) and scrolling down will decrease the heading angle (counter-clockwise), up to some limits. When adjusting the heading angle, the algorithm will again assess separation at landing and path feasibility, changing the colour of the drawn path accordingly. Hence, if a TTB point is red, the user can search for a heading angle which would result in that point being green. The user can update the chosen angle by clicking. This will update the predicted heading angle for the TTB, as well as the display. This updated angle is now used as the heading angle for every TTB point, allowing the user to assess what a change in angle would do for the solution space of the TTB points. The predicted turn in angle is visualised in text to avoid the user commanding a different heading angle than was previously set.

In the simulator, the user can hover the computer mouse over all the dots of the TTB part of the display. Doing so will plot the predicted trajectory, were the aircraft to turn to base at this point, as can be seen in figure 1.11. The trajectory will be green if it is possible for the aircraft to take this trajectory and if taking it does not cause a loss of separation on final approach. This is the case for the situation in figure 1.11. Next to plotting the predicted trajectory, the algorithm will also visualise a ghost aircraft on the ILS. The location of this ghost aircraft is equal to the location at which the selected aircraft will be at the time its leading aircraft lands. A red line is also visualised on the ILS, representing the minimum separation the aircraft must have with respect to its leading aircraft. This way, the user can assess the resulting separation on final approach, were the aircraft to turn to base at the selected TTB Point.

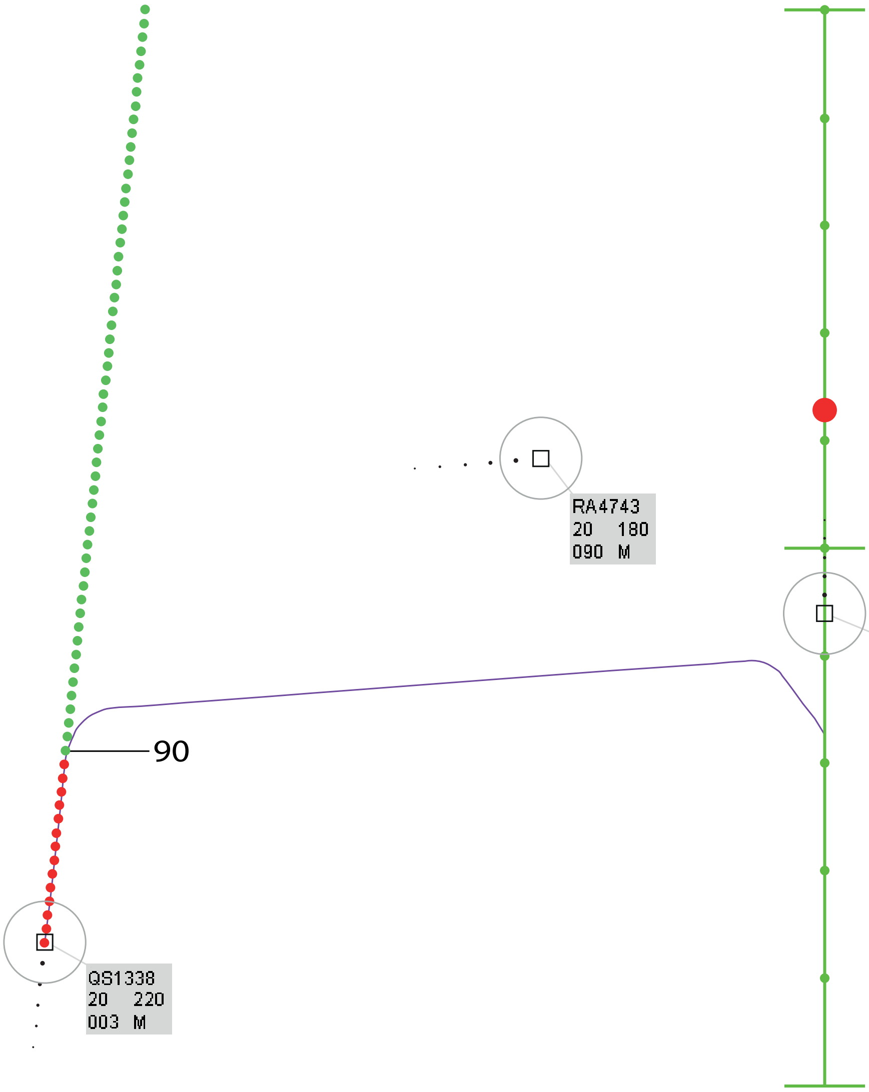

RA4743
20 180
090 M
SG3047
20 160
185 H
90
QS1338
20 220
003 M

Figure 1.10: The ITIP Display for a selected aircraft in leg 1

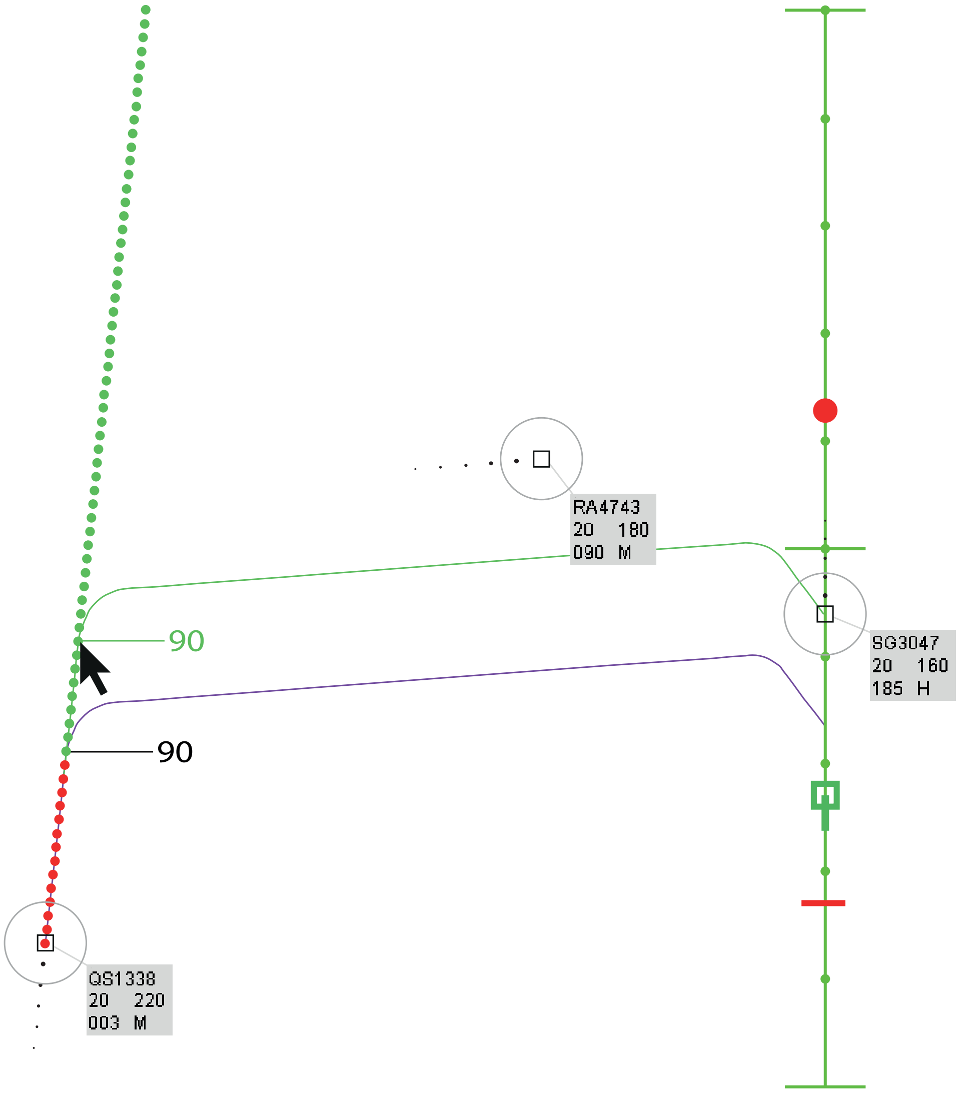


Figure 1.11: The ITIP Display for a selected aircraft in leg 1, when hovering over a TTB Point with the computer mouse

To maximise throughput, aircraft must be as close to the TBS marker of a leading aircraft as possible, at the moment the leading aircraft lands. Using the ghosting tool, one can investigate which solution is optimal, when deciding which aircraft to send to the ILS and with which path.

If the plotted trajectory would result in a loss of separation on final approach, or if the trajectory is unfeasible because of the aircraft having to catch the ILS from above, the trajectory will be red. This situation can be seen in figure 1.12. Here, the predicted trajectory resulting from a TTB at the selected point would result in the aircraft having to catch the ILS at less than 6.3 NM from the runway threshold. In the simulator, if an unfeasible trajectory is plotted when hovering over a point, the algorithm highlights the cause of the trajectory being unfeasible. Here, the ILS is highlighted. As one can see, the ghost aircraft is still before the TBS marker of the leading aircraft when it would land, signalling that for this trajectory separation is not a problem.

The same goes if the reason of a trajectory not being feasible is another aircraft. This situation is given in figure 1.13. In this figure, aircraft RA4743 has moved somewhat to the left, delaying its arrival. A TTB at the selected point would result in a loss of separation, as visualised by the ghost aircraft being behind the separation marker.

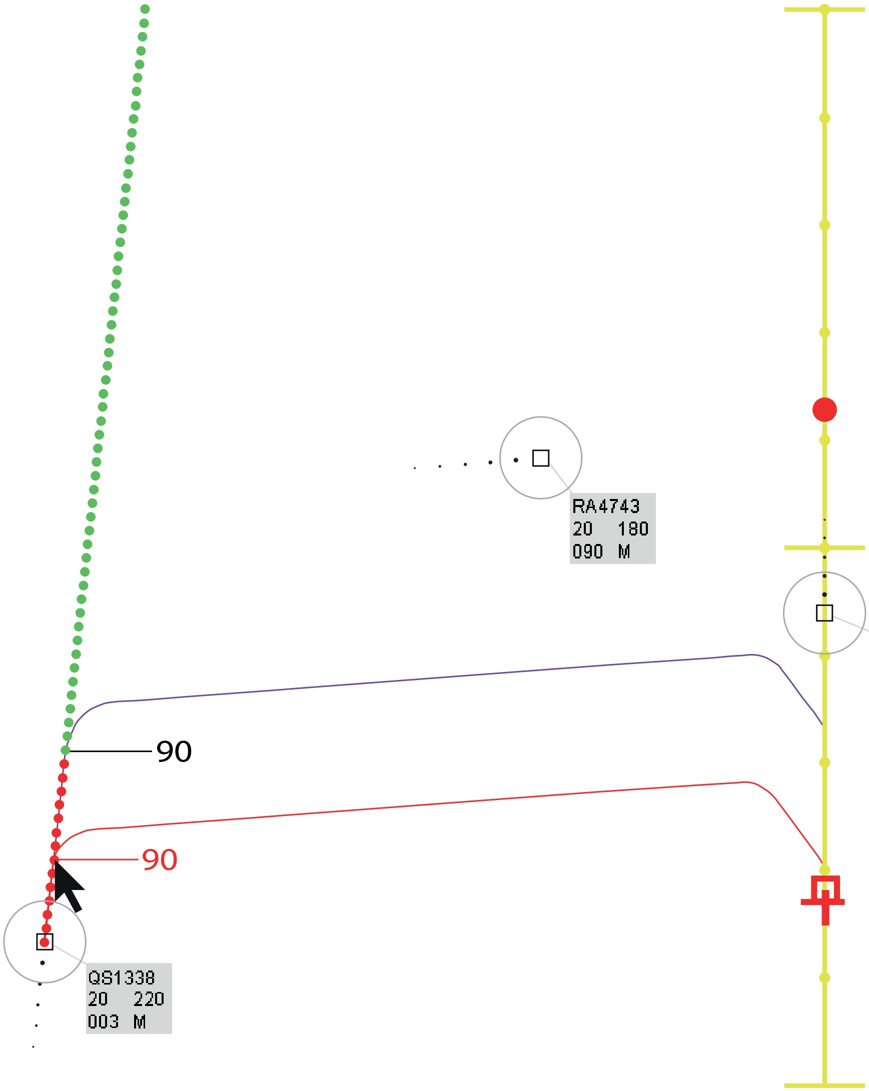

RA4743
20 180
090 M
SG3047
20 160
185 H
90
90
QS1338
20 220
003 M

Figure 1.12: The ITIP Display for a selected aircraft in leg 1, when hovering over a TTB Point resulting in an unfeasible trajectory with the computer mouse

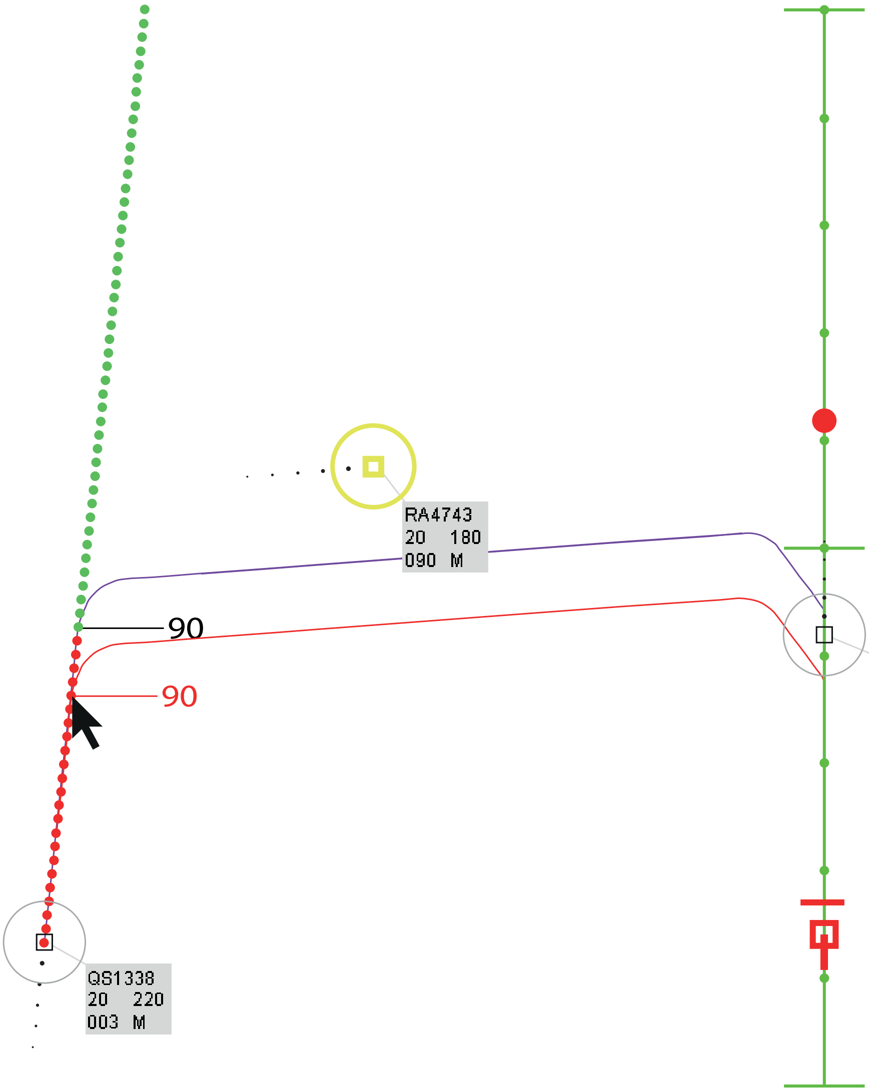

RA4743
20 180
090 M
90
90
SG3047
20 160
185 H
QS1338
20 220
003 M

Figure 1.13: The ITIP Display for a selected aircraft in leg 1, when hovering over an unfeasible TTB Point with the computer mouse

1.4.2. The TTI Display

The TTI Part of the display is largely the same as the TTB Display. The only major change is that it is visualised for aircraft flying in leg 3. The boundaries with which the user can change the turn in angle are such that the angle with respect to the ILS can only be 30 degrees or less, up to 5 degrees. Next to these changes, the cause of points that are red will again be highlighted, the resulting location at the moment the lead aircraft lands will be visualised as a ghost and the user can update the trajectory prediction by clicking TTI points and scrolling to adjust the turn in angle.

Note that this display is used for fine-tuning the (relatively) rough decisions that were made when picking the TTB point. Also note that, when choosing a TTB point for an aircraft that is just feasible, the solution space for the TTI point of aircraft that are in leg 2 or higher can be seriously affected, sometimes even reducing the possible TTI points to two or even one. Keep this in mind whilst picking your strategies.

1.5. Final Tips & Tricks for the ITIP Display

In this section, some final tips and tricks will be given. Keep these in mind during the simulations!

Always remember the goal of these scenarios: to maximise throughput whilst maintaining separation and adhering to the guidelines. Note that maintaining separation is the most important goal, a loss of separation should never occur!

Note that, while the ITIP Display assumes that the user will follow the guidelines, it is sometimes more efficient to leave the IAS of an aircraft unchanged until the user wants to interact with this aircraft, in order for the aircraft to be able to turn to base sooner. However, do keep in mind that the display only displays correct information as soon as the correct speed commands are given.

In general, for a perfect trajectory, the following commands should be given (in this order):

- A speed command of 220 kts at the moment the user wants to start using the ITIP Display
- A speed command of 180 and a heading command as planned at the TTB Point
- A cleared to catch ILS command **(do not forget this!)** and heading command at the TTI Point
- A speed command of 160 kts at the moment the aircraft catches the ILS

Again, it should be noted that the user may choose not to follow these steps but the user should keep in mind that flying at higher speeds than expected by the algorithm changes the solution space.

A good tactic for aircraft that are entering the sector directly heading for the ILS can be to put these aircraft in leg 1 first by giving them a heading command parallel to the ILS. This is because the TTI part of the ITIP Display is meant for small adjustments only, whilst the TTB portion of the display is used for initial planning and greatly impacts the arrival time of an aircraft.

You can use figure 1.7 during the tests to remember the decision points of interest.

1.5.1. Practice Scenario 4 & 5 and Test Runs

Now complete Practice Run 4 5. Focus on trying out all aspects of the ITIP Display.

After the practice scenarios, you may start the test runs. Make sure you ask any questions you may still have before starting the test runs, as communicating during the test runs is a possible confound. The order in which you will receive the displays is random. During the experiment, you may use the figures and lists in this manual. Good luck!

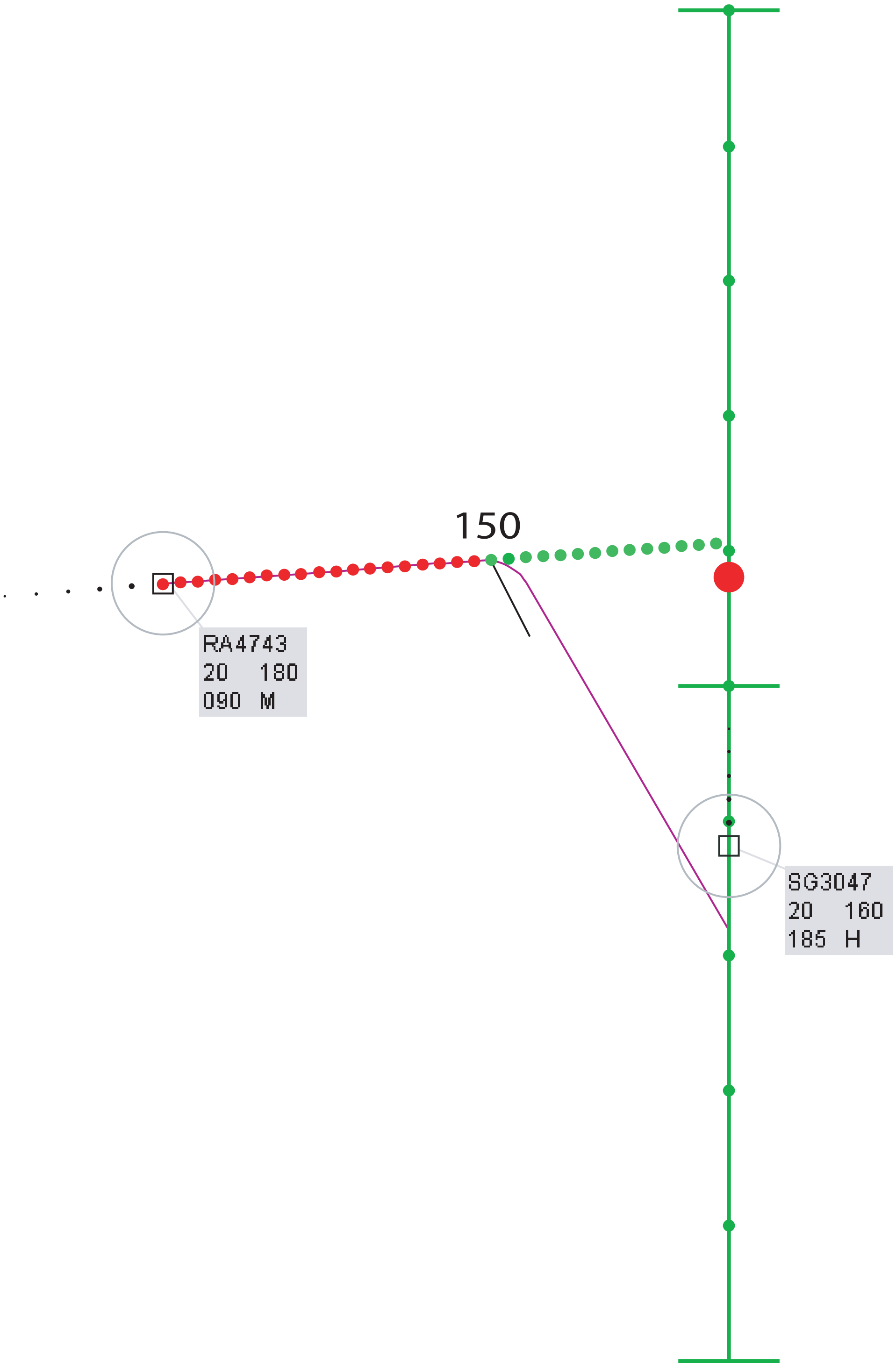

150
RA4743
20 180
090 M
SG3047
20 160
185 H

Figure 1.14: The ITIP Display for an aircraft in leg 3

2

Additional Results

2.1. Numerical Results

In Figures 2.1, the average length of aircraft path in nautical miles is given. Due to every participant completing all the scenarios, this comparison can be made. Here, a reduced path length would indicate a more efficient average solution. As one can see, there is a small increase in the path length for the ITIP display scenarios, however the difference is relatively small.

In Figure 2.2, the number of aircraft landed per scenario can be seen. Participants were able to land more aircraft using the ITIP display, indicating more throughput and thus a more efficient solution. Again, the difference is small.

In Figures 2.3 and 2.4, the number of clicks in the TTB and TTI parts of the display can be found, respectively. As one can see, the TTB click function was used less than the TTI click function. This is in line with observations from the experiment; participants used the click in TTI function to probe for the impact of certain solutions, since this updated prediction was used for the feasibility of route calculation in displays for other aircraft.

In Figures 2.5 and 2.6, the number of scrolls in the TTB and TTI part of the display can be observed. It is interesting to note that only four participants used the scroll function at all. The scroll function was used more in the TTB part which was to be expected since the impact of a turn-in heading angle change is more significant for the TTB.

The small use of the clicking and scrolling functions in the display further underscores the observation that participants mainly used the display in a rule-based fashion; they did not experience incentive to explore. Participants noted that they felt they would have been able to get more use out of the display with more training; perhaps the incentive to explore was not present due to the participants still learning the general approach task.

In Figures 2.7, the number of trajectories that were drawn is given. As can be seen, participants made more use of this option for the harder scenarios. This would indicate that the participants used the display more actively in the harder scenarios.

In Figure 2.8, the number of DBS conflicts can be found. No clear trend is visible; it seems like all the DBS conflicts were just random mistakes.

2.2. Survey Results

2.2.1. Survey Questions

Participants were asked to complete a survey after the experiment. In the survey they were asked to input a level of agreement to some statements varying from strongly disagree to strongly agree. After these rating

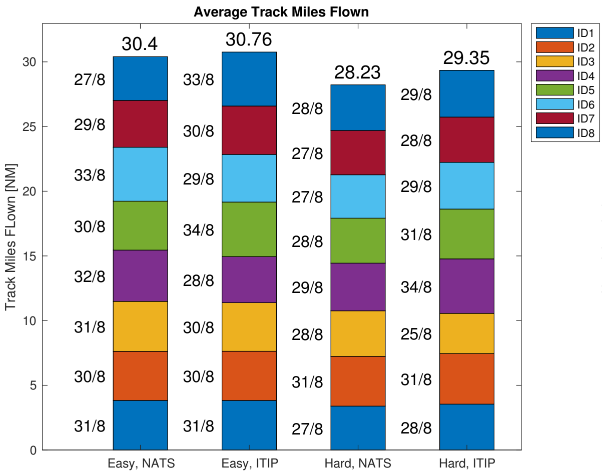


Figure 2.1: Average track miles flown per aircraft in nautical miles, per scenario

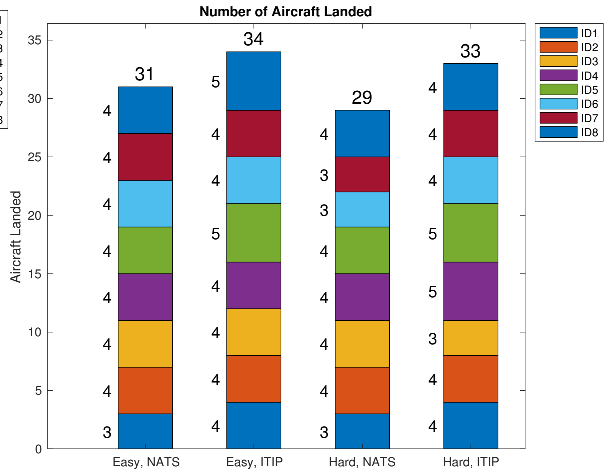


Figure 2.2: Number of aircraft landed, per scenario

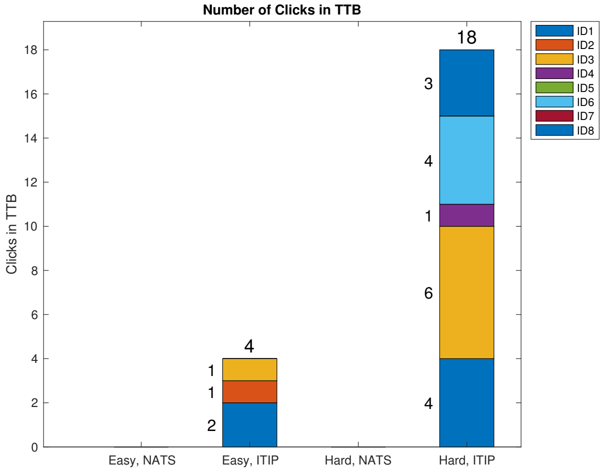


Figure 2.3: Number of clicks in TTB display, per scenario

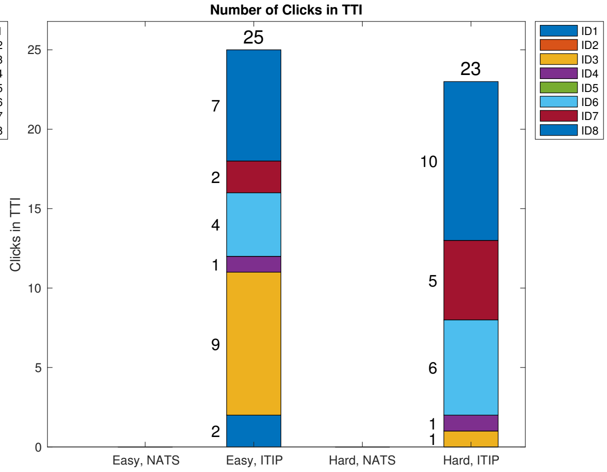


Figure 2.4: Number of clicks in TTI, per scenario

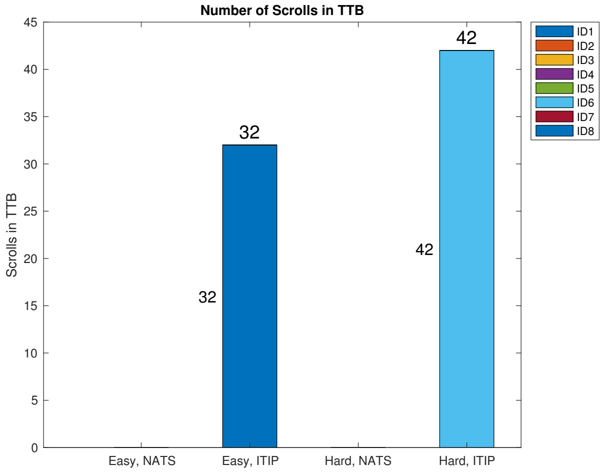


Figure 2.5: Number of scrolls in TTB display, per scenario

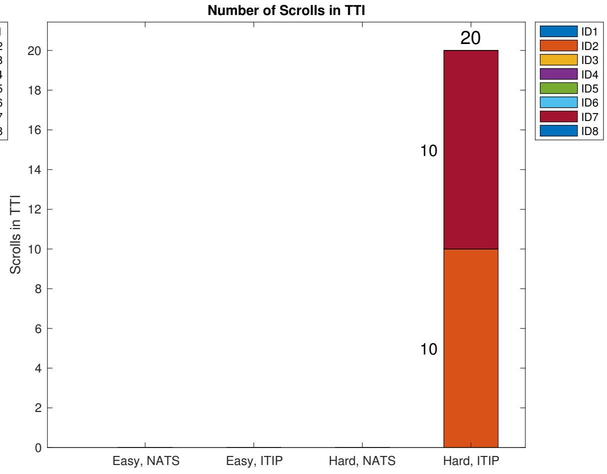


Figure 2.6: Number of scrolls in TTI, per scenario

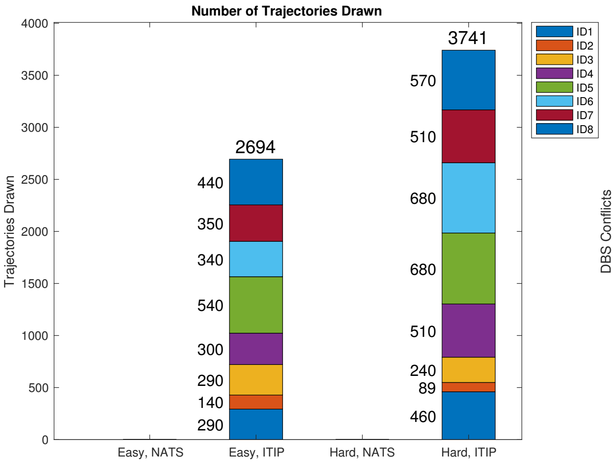


Figure 2.7: Number of trajectories drawn, per scenario

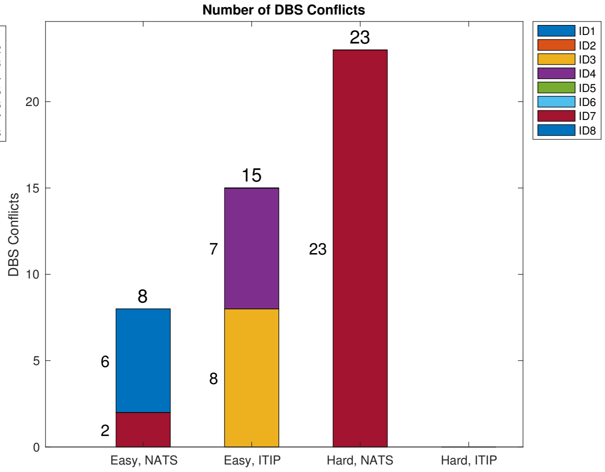


Figure 2.8: Number of DBS conflicts, per scenario

questions, some open questions were asked.

The rating questions in the survey were as follows:

1. The ITIP Display is a helpful tool for APP ATC;
2. I made use of the ITIP Display when making control decisions;
3. The NATS Display helped me maximise throughput;
4. The ITIP Display helped me maximise throughput;
5. The ITIP Display helped me maintain Time-Based Separation;
6. The NATS Display helped me maintain Time-Based Separation;
7. The ITIP Display was more useful to me than the NATS Display;
8. The use of the ITIP Display was clear to me;
9. Using the ITIP Display, I was confident of my control decisions;
10. I trusted the ITIP Display when making decisions;
11. The ITIP Display increased my understanding of the scenario;
12. The ITIP Display made me see solutions that would not have been apparent without it;
13. I feel that using the ITIP Display has trained my APP control skills.

The open questions in the survey were as follows:

1. What did you find most useful about the NATS Display?;
2. What did you find most useful about the ITIP Display?;
3. Which elements in the ITIP Display did you use? (participants could pick from a list and add their own options);
4. Which element(s) did you find most useful and why?;
5. Did you use different in tactics for the different displays? If yes, what was that difference?;

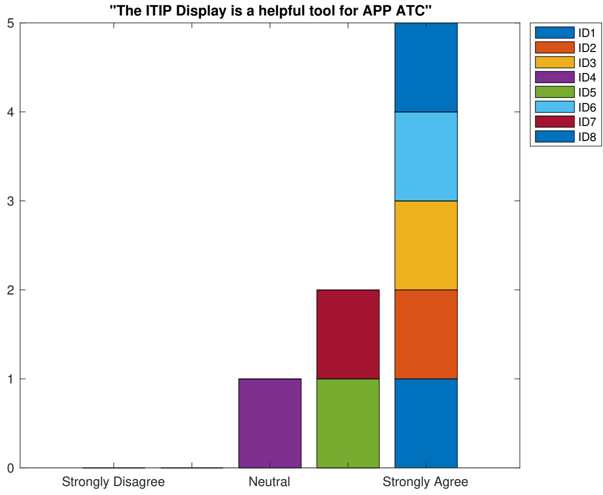


Figure 2.9: Survey answers to the 1^{st} statement

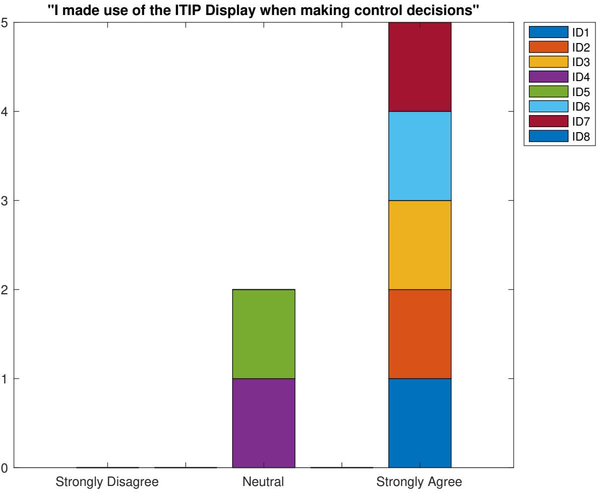


Figure 2.10: Survey answers to the 2^{nd} statement

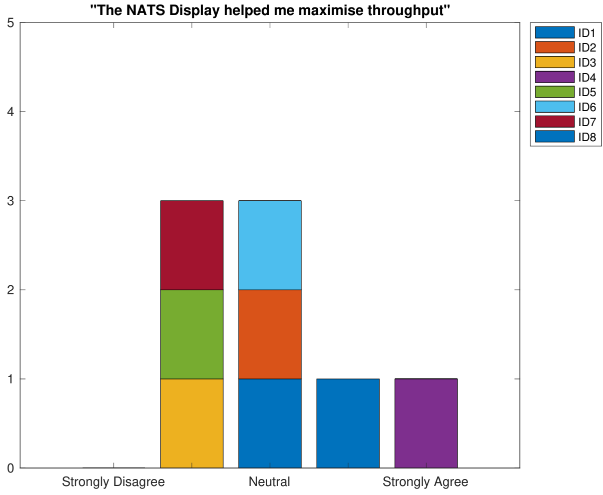


Figure 2.11: Survey answers to the 3^{rd} statement

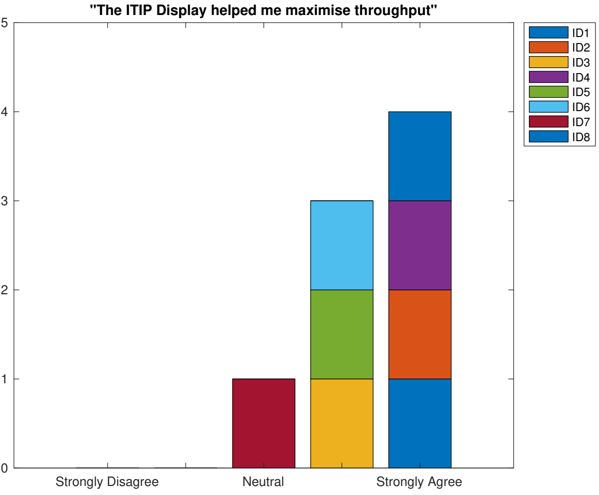


Figure 2.12: Survey answers to the 4^{th} statement

6. What would you want to improve about the NATS Display?;
7. What would you want to improve about the ITIP Display?.

In addition to the test participants, the survey was also given to professional APP controllers at the LVNL. For this LVNL version of the survey, rating questions 1 and 13 were omitted and the following questions were added:

1. Automation will influence the way ATCos work within 10 years (rating);
2. With further research, the ITIP Display should be (partly) implemented in future APP control (rating);
3. The ITIP Display cluttered the radar screen (rating);
4. Do you have any other comments? (open).

2.2.2. Results of Survey Rating Questions for Test Participants

In Figures 2.9 - 2.21, the given answers to the rating questions, as given by the pest participants are shown. As can be seen in the plots, the participants were very positive about the ITIP display. They indicated liking the ITIP display more than the NATS display, noting that they trusted the display and were confident of their

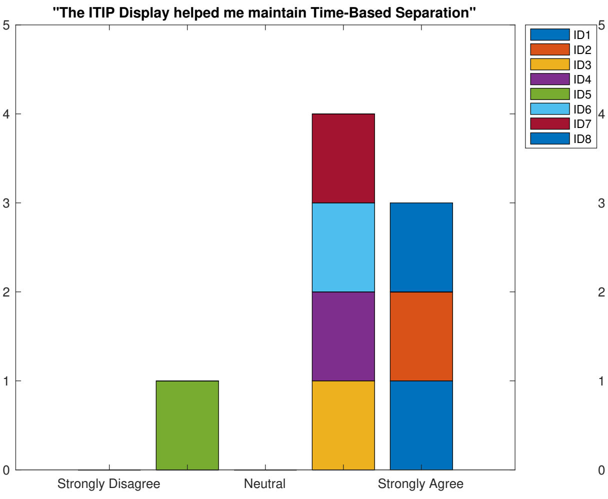


Figure 2.13: Survey answers to the 5^{th} statement

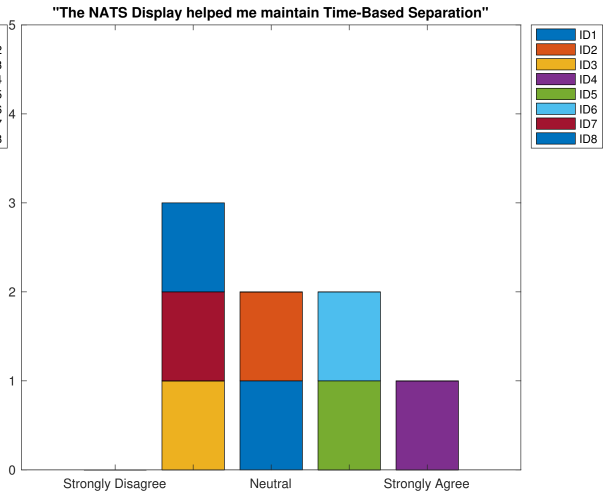


Figure 2.14: Survey answers to the 6^{th} statement

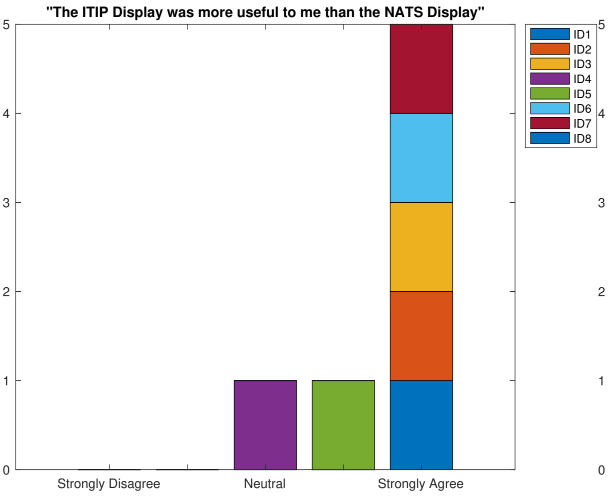


Figure 2.15: Survey answers to the 7^{th} statement

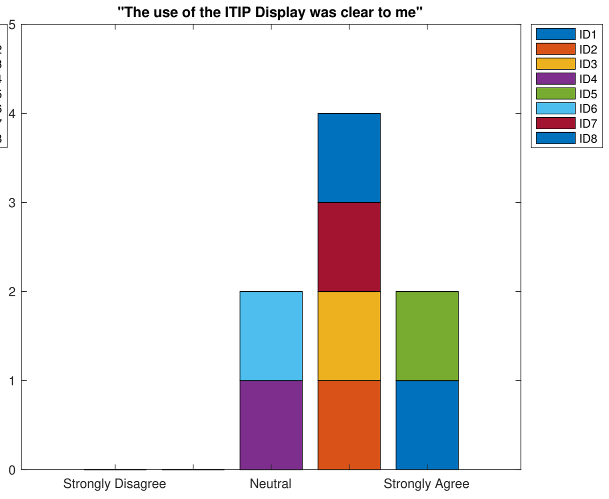


Figure 2.16: Survey answers to the 8^{th} statement

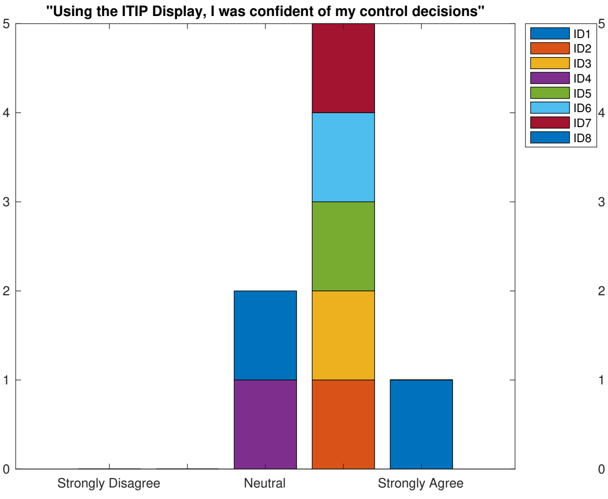


Figure 2.17: Survey answers to the 9^{th} statement

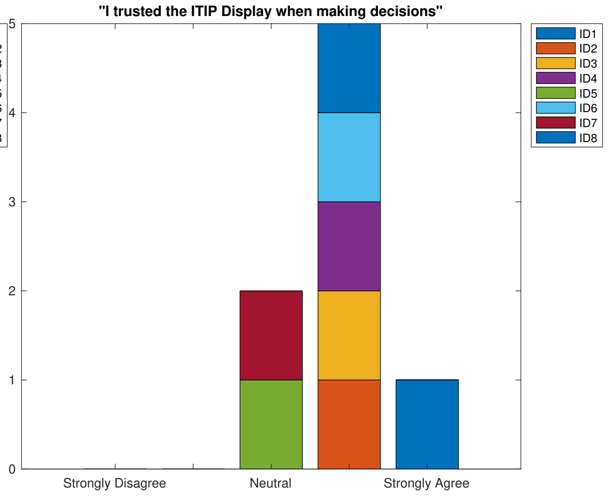


Figure 2.18: Survey answers to the 10^{th} statement

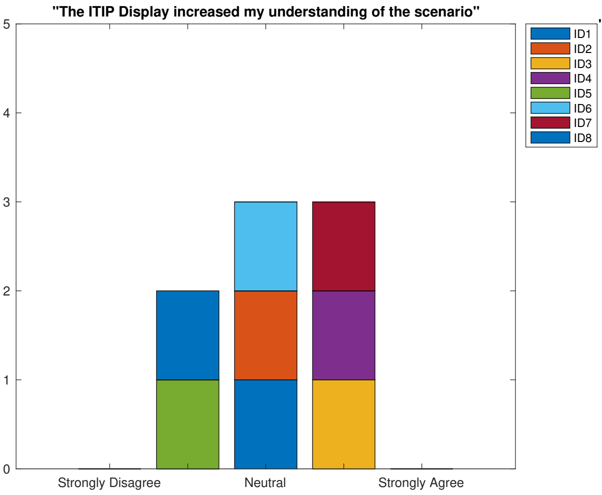

"The ITIP Display increased my understanding of the scenario"

Figure 2.19: Survey answers to the 11^{th} statement

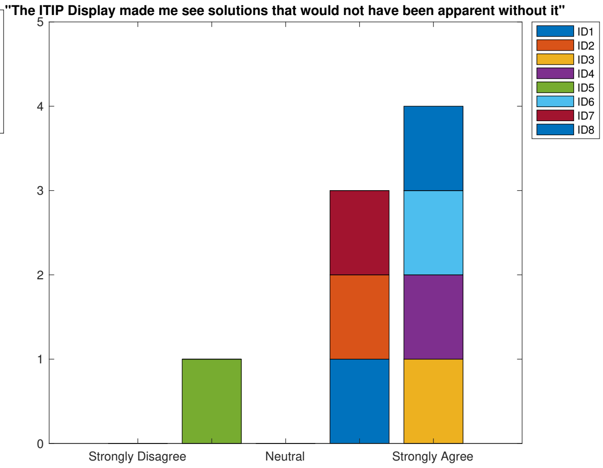

"The ITIP Display made me see solutions that would not have been apparent without it"

Figure 2.20: Survey answers to the 12^{th} statement

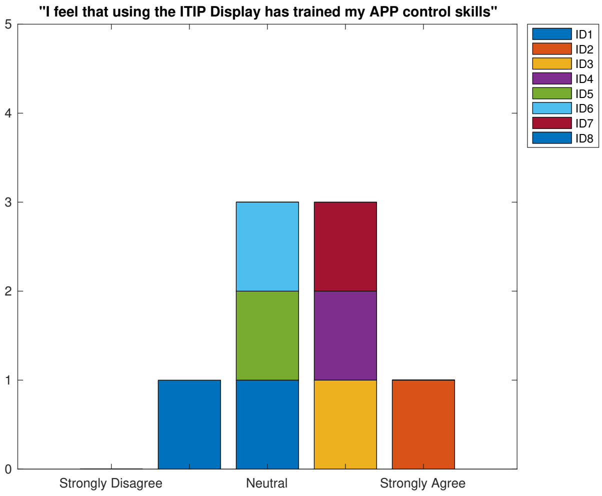

"I feel that using the ITIP Display has trained my APP control skills"

Figure 2.21: Survey answers to the 13^{th} statement

decisions while using it. The ITIP display did not increase scenario understanding and participants felt the display did not increase their general APP controller skill. Especially the scenario understanding is of importance; further research should focus on whether the display decreases situation awareness which is to be avoided.

2.2.3. Results of Survey Open Questions for Test Participants

To the question *'What did you find most useful about the NATS Display?'*, the participants answered the following:

1. *'The blue ORD marker, I used it to time the actions toward intercepting the ILS.'*
2. *'Knowing the dynamic distances for TBS.'*
3. *'The "ghost dots" are useful, perhaps with more training I would have been able to better use these.'*
4. *'The red and blue dots.'*
5. *'The TBS and ORD marker are useful to use as a target to steer aircraft towards.'*
6. *'blue dots.'*
7. *'confirmation of spacing when already on ILS.'*
8. *'Blue target to aim for as an intercept point, although the development of the scenario in the final stage of the approach was sometimes still surprising.'*.

To the question *'What did you find most useful about the ITIP Display?'*, the participants answered:

1. *' Almost everything, I heavily relied on the green dots and mostly used the indication of the optimal route toward the ILS. It was most useful in timing the TTB moment, but also the ILS interception. I hardly used the scroll probes to inspect other trajectories to perhaps further maximise my throughput. '*
2. *' Accurate timing advice. '*
3. *' Everything! Helps with planning, separation, ordering/sequencing, seeing solutions that are not clear without it. '*
4. *' The TTB and TTI advisory lines with the red and green dots. '*
5. *' The ITIP helped me validate (and optimise) the control strategies that I had planned. '*
6. *' ability to foresee when an action is needed. '*
7. *' Showing early turn in opportunities. '*
8. *' Exploring the problem and possible solutions. '*.

To the question *'Which elements in the ITIP Display did you use?'*, the participants answered:

1. *' The green and red dots for the TTB Decision point, The green and red dots for the TTI Decision point, The purple predicted trajectory, The trajectories plotted when hovering over a TTB point, The trajectories plotted when hovering over a TTI point. '*
2. *' The green and red dots for the TTB Decision point, The green and red dots for the TTI Decision point, The purple predicted trajectory. '*
3. *' The green and red dots for the TTB Decision point, The green and red dots for the TTI Decision point, The purple predicted trajectory, The trajectories plotted when hovering over a TTB point, The trajectories plotted when hovering over a TTI point, The ghost of the aircraft when hovering over a TTB point, The ghost of the aircraft when hovering over a TTI point. '*
4. *' The green and red dots for the TTB Decision point, The green and red dots for the TTI Decision point. '*

5. *‘ The green and red dots for the TTB Decision point, The green and red dots for the TTI Decision point, The purple predicted trajectory, The trajectories plotted when hovering over a TTB point, The ghost of the aircraft when hovering over a TTB point, I did not use the TTB and TTI heading select at all. ’*
6. *‘ The green and red dots for the TTB Decision point, The green and red dots for the TTI Decision point, The ghost of the aircraft when hovering over a TTB point, The ability to change TTB heading by scrolling. ’*
7. *‘ The green and red dots for the TTB Decision point, The green and red dots for the TTI Decision point. ’*
8. *‘ The green and red dots for the TTB Decision point, The green and red dots for the TTI Decision point, The ghost of the aircraft when hovering over a TTB point, The ghost of the aircraft when hovering over a TTI point, Selecting a different turn-in point to make room for another aircraft. ’.*

To the question *‘Which element(s) did you find most useful and why?’*, the participants answered:

1. *‘ I mainly used the display in a rule-based fashion; whenever the aircraft was on a green dot, I gave the TTB command. Likewise for the TTI moment. Very rarely did I look at the ghost or tried another intercept trajectory using the scroll function. As such, most useful are the green en red dots with the optimal TTB and TTI indicators. ’*
2. *‘ Change from red to green points because it aids in critical timing. ’*
3. *‘ The green/red dots, as it shows you beforehand when actions should be taken and this helps you to prioritise your control actions. ’*
4. *‘ TTB and TTI decision points provide comfort feeling when taking complex decisions. ’*
5. *‘ I found the TTB information the most useful, because that can be used to make an efficient initial planning. Final fine-tuning can be done later by using speed. ’*
6. *‘ green dots. ’*
7. *‘ green/red dots, for timing decisions. ’*
8. *‘ Decision points because they allowed me to time turning commands. Counting the number of red dots for different aircraft gave me a clue to the sequencing of commands (who should I tell to turn first?). ’*

To the question *‘Did you use different in tactics for the different displays? If yes, what was that difference?’*, the participants answered:

1. *‘ Yes, for the ITIP I relied on the displayed features. For the NATS, I tried to estimate myself when a TTB was suitable and then used the flight label to time the TTI moment: I dragged the label over the ILS and when the blue dot intersected the flight label connector, I gave the TTI command. ’*
2. *‘ Yes, the NATS display led me to work more tactically, whereas the ITIP display allowed me to plan more ahead. ’*
3. *‘ Yes. Without ITIP you are more conservative, you are more working towards assuring separation. This is also the downside of ITIP, you are pushing the limits. ’*
4. *‘ Yes, in the NATS display, my last exercise, I applied tromboning which standardised the turn commands. This relieved my workload. ’*
5. *‘ Not fundamentally different, however, I used the information on the NATS display as direct control targets, whereas I used the ITIP display more to validated my control strategies. ’*
6. *‘ yes, NATS more busy with planning, used bigger margin, ITIP more busy with optimisation and timing. ’*
7. *‘ no. ’*

8. *‘ ITIP display was still used mechanically, so without adding any situational awareness. I can imagine this would change with more experience. NATS was much closer to conventional control and just provided some useful assistance without changing the control strategy (from conventional control). ’*

To the question *‘ What would you want to improve about the NATS Display?’*, the participants answered:

1. *‘ Support for timing the TTB moment. ’*
2. *‘ Help for the TTB. ’*
3. *‘ Some more info about "advised sequence" would be helpful. ’*
4. *‘ – ’*
5. *‘ The display additions themselves are very minimalistic (which is good), however, the actual control strategies and control operations (speed, alt, heading, etc) are left completely up to the controller. At least a simple "Turn to Base" cue would be helpful. ’*
6. *‘ marking of the sequence in label? with a very busy airspace, matching up ac to dots was tricky. ’*
7. *‘ make it so that you don’t have to extrapolate yourself where your aircraft and the blue dot converge. ’*
8. *‘ I had trouble with the predictability (I thought I was doing fine and still got a conflict). This could be my lack of understanding or deficient control technique. ’*

To the question *‘What would you want to improve about the ITIP Display?’*, the participants answered:

1. *‘ Integrate the ability to warn and resolve conflicts (loss of separations) when aircraft did not yet received a TTB. Now, conflict detection and resolution before TTB must be done yourself, but once aircraft received a TTB command, they were taken into account in the red dots for aircraft that were still flying parallel to the ILS. ’*
2. *‘ HMI elements, to articulate some elements more than others, for some elements perhaps application of more visibility rules and or on-request only. ’*
3. *‘ Perhaps include a safety buffer such that when you push towards the limits you are still safe. ’*
4. *‘ "I used few of the information. The information overload provided a higher learning curve (for me). Less routing information can be considered. I liked the green and red indications. The ITIP display made it hard to combine a tromboning technique because the sequence was disturbed when guiding aircraft far out to the ILS. Not providing advise for aircraft to far out can be considered." ’*
5. *‘ I noticed that I controlled very close to the boundaries of safe control with the ITIP Display resulting in a number of slight separation overruns. Perhaps showing the margins more explicitly for a given control action would prevent that (i.e., show separation margins in time and distance for a given control action). ’*
6. *‘ declutter? and maybe setting a timer/blinking when a turn point selected. ’*
7. *‘ I had the feeling at least once that a green area shrunk when it shouldn’t have. Also, although I may have a sequence in my head, the display only incorporates this when I maneuver an aircraft. It would be nice if you could see the effect of sequencing choices ahead." ’*
8. *‘ There was no assistance to sequence two aircraft who are both on down wind on two sides of the localizer. To get guidance I had to turn one of them to base leg. Secondly a future feature might be to queue commands, so select a turn-in point and automatically let the system send that command to the aircraft over data link. This would lower my workload in the situations where I was just counting down red dots until I could send the command. ’*

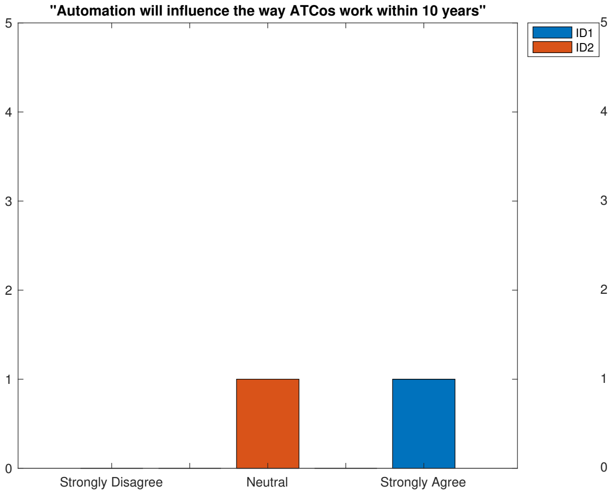


Figure 2.22: Survey answers of professional controllers to the 1^{st} statement

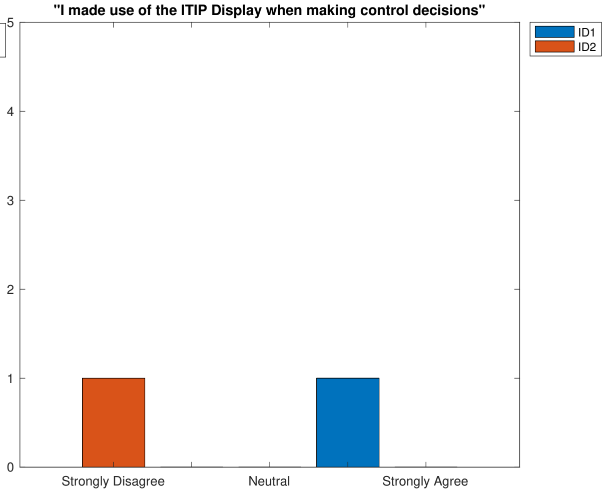


Figure 2.23: Survey answers of professional controllers to the 2^{nd} statement

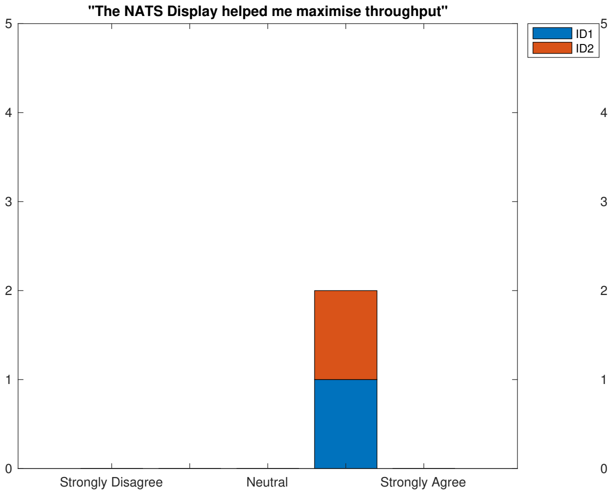


Figure 2.24: Survey answers of professional controllers to the 3^{rd} statement

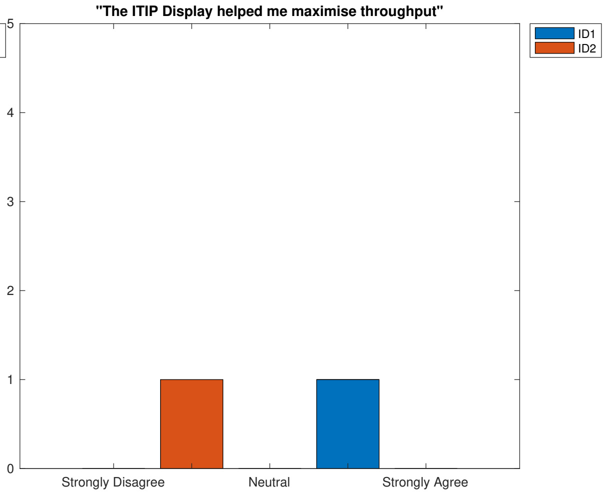


Figure 2.25: Survey answers of professional controllers to the 4^{th} statement

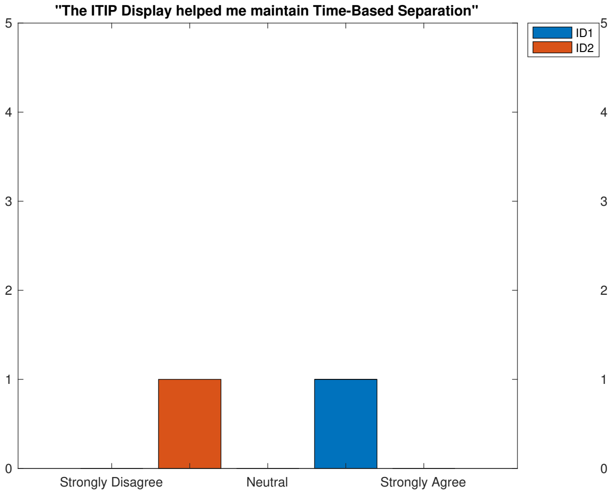


Figure 2.26: Survey answers of professional controllers to the 5^{th} statement

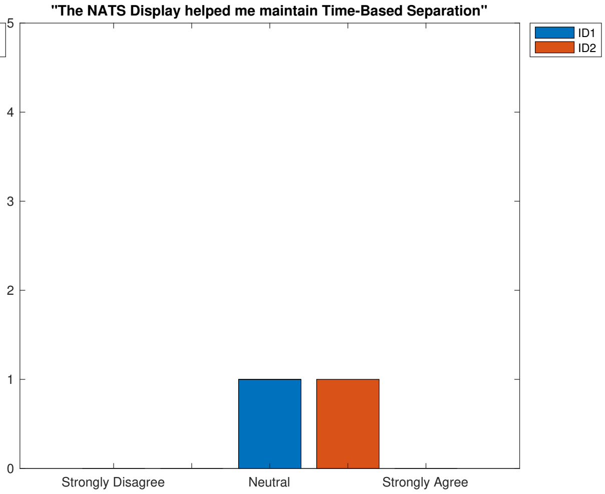


Figure 2.27: Survey answers of professional controllers to the 6^{th} statement

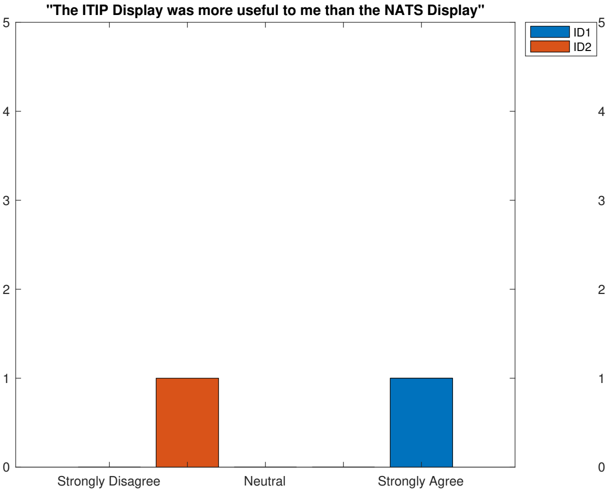


Figure 2.28: Survey answers of professional controllers to the 7^{th} statement

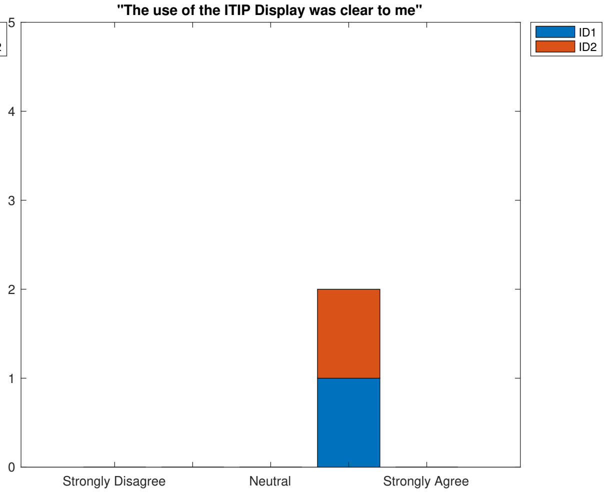


Figure 2.29: Survey answers of professional controllers to the 8^{th} statement

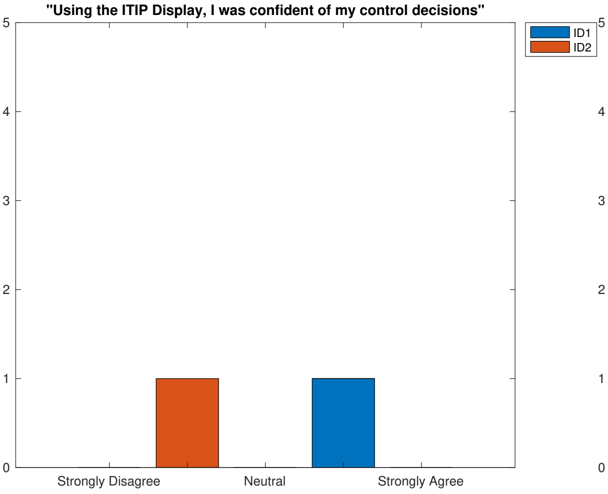


Figure 2.30: Survey answers of professional controllers to the 9^{th} statement

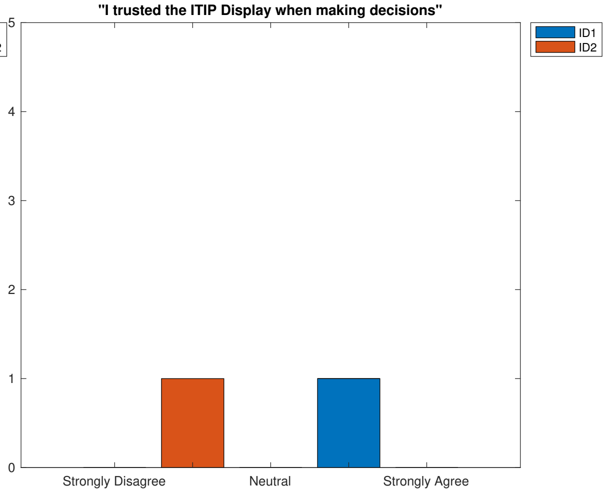


Figure 2.31: Survey answers of professional controllers to the 10^{th} statement

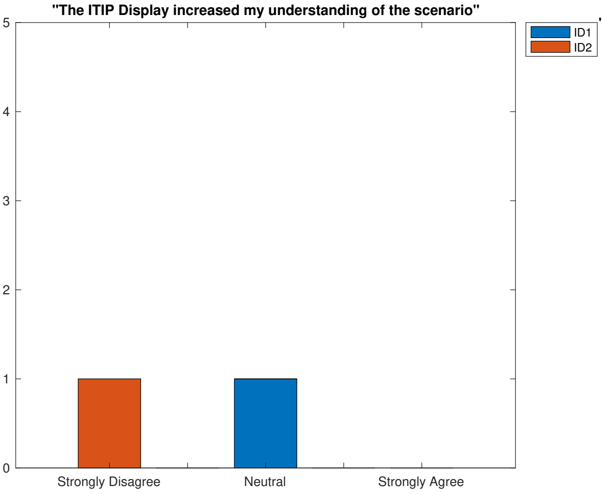


Figure 2.32: Survey answers of professional controllers to the 11^{th} statement

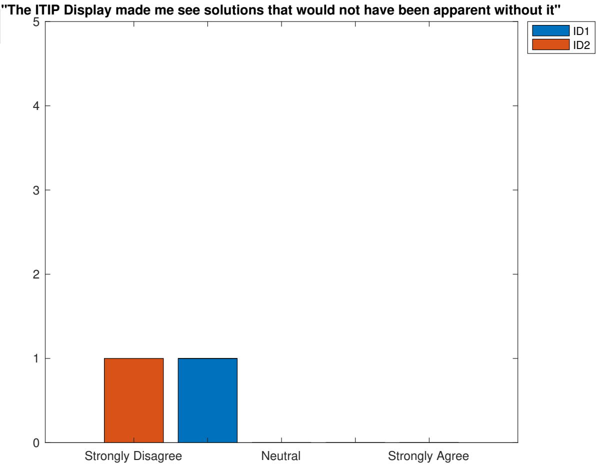


Figure 2.33: Survey answers of professional controllers to the 12^{th} statement

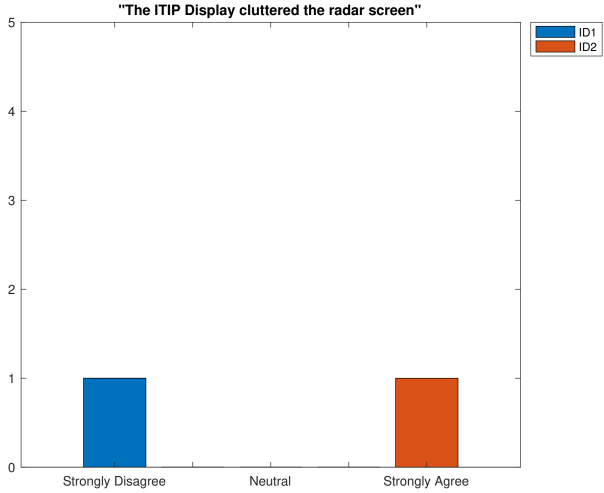


Figure 2.34: Survey answers of professional controllers to the 13^{th} statement

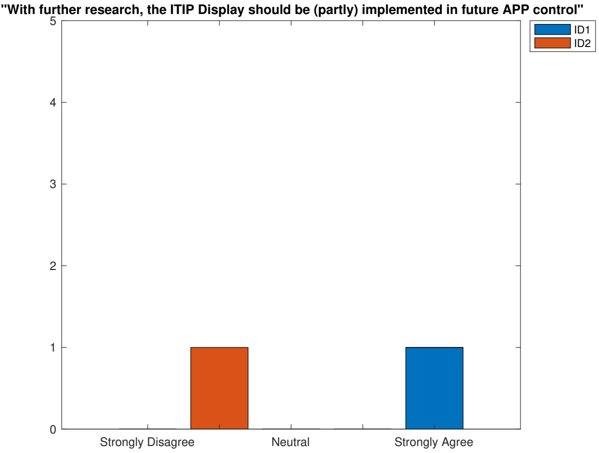


Figure 2.35: Survey answers of professional controllers to the 14^{th} statement

2.2.4. Results of Survey Rating Questions for Professional Controllers

In Figures 2.22 - 2.35, the answers of the professional ATCOs to the rating questions can be found. It is clear that the two ATCOs were of different opinion; one was very positive about the display while the other was more conservative. It should be noted that the more conservative controller also has a more conservative stance toward ATCo support in general, as can be seen from the answers to the question *'Automation will influence the way ATCos work within 10 years'* .

2.2.5. Results of Survey Open Questions for Professional Controllers

To the question *'What did you find most useful about the NATS Display?'*, the professional ATCos answered:

1. *' I have seen the NATS tool live at Heathrow approach and for me as a controller is works really straight forward. Aim at the moving line, that's it. Bu there is no help when to turn for example that's why I find ITIP even better. '*

2. *' Simple indicator, following the controllers decisions. '*

To the question *'What did you find most useful about the ITIP Display?'*, the professional ATCos answered:

1. *'The green and red dot indication really helps to make you decide when to turn to base or final.'*

2. *'TTI dots.'*

To the question *'Which elements in the ITIP Display did you use?'*, the professional ATCos answered:

1. *'The green and red dots for the TTB Decision point, The green and red dots for the TTI Decision point, The trajectories plotted when hovering over a TTB point, The trajectories plotted when hovering over a TTI point, The ghost of the aircraft when hovering over a TTB point, The ghost of the aircraft when hovering over a TTI point, The ability to change TTB heading by scrolling, The ability to change TTI heading by scrolling.'*

2. *'The green and red dots for the TTI Decision point.'*

To the question *'Which element(s) did you find most useful and why?'*, the professional ATCos answered:

1. *'I mostly used the green en red dot indicators for turn to base and final.'*

2. *'-'*

To the question *'Did you use different in tactics for the different displays? If yes, what was that difference?'*, the professional ATCos answered:

1. *' I only did one exercise with the ITIP so cant really say that. I mostly used my controller insight and used ITIP to confirm or as a help when to turn base and final.'*
2. *'No.'*

To the question *'What would you want to improve about the NATS Display?'*, the professional ATCos answered:

1. *'Purely based on what I have seen in real life I wouldn't change a thing.'*
2. *'Initially nothing.'*

To the question *'What would you want to improve about the ITIP Display?'*, the professional ATCos answered:

1. *'To make it even better, as a 2.0 version maybe, speed advise would be a great extra feature.'*
2. *'Change ghost blips to small lines.'*

To the question *'Do you have any other comments?'*, the professional ATCos answered:

1. *'Not really. Great experimental tool which could be a great benefit in real life to support or help controllers optimise the final approach.'*
2. *' At the moment I don't need additional support, just the TBS indicator. More support leads to a more passive controller and might delay corrective actions if needed.'*

2.2.6. Discussion of Survey Results

In general, two things can be concluded from the survey results. Firstly, the test participants were mainly very positive about the display. They noted that they made frequent use of the display and that it helped them maximise throughput, even more so than the NATS display. Secondly, for the professional ATCos, two opinions were heard; on the one hand the ATCo who felt more positive toward more ATCo support was very positive about the ITIP display, while the somewhat more conservative controller was more sceptical. It should be noted that this controller was sceptical toward extra automation and controller support in general, as can be seen from the controller's answers to the rating question *'Automation will influence the way ATCos work within 10 years'* and the open question *'Do you have any other comments?'*, both of which indicate a conservative stance. These survey answers were in line with frequently heard opinions at the LVNL; on the one hand there are ATCos who welcome more automation and support and on the other hand there are ATCos who are more conservative and sceptical of automation and support. Negative feedback of these controllers mostly focused on this conservative stance toward automation and extra controller support than on the concept of the ITIP display itself.

The positive stance of test participants toward the display can partly be explained by their limited level of experience with regards to the APP control task. Naturally, non-professional controllers will welcome support more since they are new to the task and need every support they can get. As such, it might also be interesting to look at whether the ITIP display can be used in training of new ATCos.

3

Code Layout

3.1. Aircraft Class

In the Aircraft class for the ITIP display function, a lot of use is made from datalogs. Here, a datalog is one ArrayList containing secondary ArrayLists which contain floats. A datalog is equal to an aircraft trajectory, with each secondary arraylist corresponding to a point in the future along that trajectory. In a datlog point the following floats are stored:

0. Heading angle in Degrees
1. Altitude in feet
2. Indicated Airspeed in knots
3. x location in nautical miles
4. y location in nautical miles
5. Predicted time it will take the aircraft to get to this point in seconds

Here, the heading angle is taken as positive clockwise with true north being a heading of 0 degrees, x location is the lateral distance on the screen from the centre and y location is the vertical distance on the screen from the centre.

3.1.1. ITIP Data Update Methods

The data needed for the ITIP display is derived in two main methods: updateTTBDisplayData(dt_sec) and updateTTIDisplayData(dt_sec). Both methods have the same working principle; they store trajectory data for each turn-in-point and assess trajectory feasibility. Because these methods function basically in the same way, only the updateTTBDisplayData(...) will be explained here.

updateTTBDisplayData(dt_sec) is called in the aircraft.tick(...) method, if the aircraft is in leg 1. It takes the timestep dt_sec as an input, in the simulator 5 seconds is widely used. The method does not return anything; its function is to store the data needed for the TTB part of the ITIP display in two ArrayLists: TTBLandTimeList and TTBDataLogList.

The TTBDataLogList is an ArrayList containing datalogs. As such, each ArrayList in TTBDataLogList contains an entire trajectory. Each datalog in TTBDataLogList corresponds to a TTB point, the datalog contains the entire trajectory if the aircraft were to TTB at this point. These final points of these datalogs contain the land times for each TTB point, these are used to create the TTBLandTimeList. The TTBDataLogList is used for the plotting of trajectories when the computer mouse hovers over a TTB point.

TTBLandTimeList is an ArrayList, where each ArrayList contains two floats and corresponds to one TTB point. The first float contains the predicted time it would take for the aircraft to land if the aircraft were to TTB at this point. The second float signals if this trajectory is considered feasible; the float will be equal to 1 if the trajectory is feasible and 0 if the trajectory is unfeasible. This list is used to decide the colour of the TTB points.

The updateTTBDisplayData() method works as follows. First, the path that the aircraft would have if it was to keep its heading angle is integrated using the input dt_sec as timestep. For this, the method integrateStraightPath(...) is used. Next, using the getLeg(...) methods described in section 3.1.5 and the mergeDataLogs(...) method described in Section 3.1.7, a datalog is created for that TTB point. This datalog is stored and used to determine the predicted land time. These landing times are stored in TTBLandTimeList and their feasibility is assessed using the isSeparated(...) method. The trajectory feasibility is stored as a numbered boolean in TTBLandTimeList.

The last function of this method is to set the predicted turn-in point. This is done by assessing landing time and feasibility. The TTB point with the shortest time to land is chosen, after its feasibility has been verified. Setting this point as the predicted turn-in point is done by storing the turn-in distance as measured from the runway, parallel to the ILS in the local variable TTBDist. This

3.1.2. updateTP(dt_sec)

The updateTP(...) method updates the Trajectory Prediction (TP) for the aircraft. It is ran in the tick function of the aircraft class, as well as when the user updates the TP by clicking in the display.

The method first checks in which leg of flight the aircraft is using the getLegOfApproach() method. It then applies the relevant getLeg(...) methods, in the right order and with the current aircraft state as first state. The datalog resulting from this integration procedure is stored in the local variable tpDataLog, which can be retrieved by other classes using the getTPDataLog() method. The landing time, which is the time of the last point in the datalog, can be retrieved by the getLandTime() method.

3.1.3. updateNATSMarkerData()

updateNATSMarkerData() stores all the data necessary for plotting of the NATS display in local variables. First, the method checks if a lead aircraft is present, using the getLeadAcIf(...) function from the ILS class. If this is the case, the x and y position of the lead aircraft are stored in the local variables x_lead_NATS and y_lead_NATS. The time it takes for the lead aircraft to get on the ILS, if this aircraft has not yet caught the ILS, is stored in time_to_ILS_NATS. The distance where the NATS TBS marker should be as seen from the leading aircraft is stored in length_TBS_NM_NATS and the distance of the ORD marker behind the TBS marker is stored in the length_ORD_NM_NATS local variable, using the getHarmonica(...) method.

getHarmonica(lead, follow, callTime) returns the ORD distance in NM behind the TBS marker. It does this by integrating all groundspeed differences between the leading aircraft provided in the lead variable and the follower aircraft provided in the follow variable. The callTime variable is used as a failsafe; it is increased by one every time the function calls itself and the function quits trying after this variable reaches five.

getNATSData() returns an arraylist with all the local variables set in the updateNATSMarkerData() method.

3.1.4. ILS Related Methods

catchILS(dt_sec) is a method that is ran for every aircraft tick, if the aircraft local variable catchILSOn is true. Here, catchILSOn is set by the user in the Command Display Unit (CDU). The method catchILS() checks if the aircraft needs to steer into the ILS. It does this using the method startTurnILS(...), which returns true when the aircraft should start to turn. When the startTurnILS() function returns true, the ILS turn-in is initiated; the catchILS(...) function sets the target heading to the angle of the ILS and the local variable caughtILS to true. When the aircraft is on the ILS, the local boolean variable onILS is set to true, after which for each tick the heading agle is set such that the aircraft remains on the ILS. As a failsafe, the position of the aircraft is set to the location of its projection on the ILS. When the aircraft encounters the ILS, the altitude is set to the altitude of the ILS at that point. When the aircraft crosses the 4DME point, the target IAS is set to 130 kts and when the aircraft touches down, the local variable landedILS is set to true. The landedILS variable being true is the trigger for the aircraft 'deleting' itself; all functions interacting with the aircraft class only do so when the aircraft has not landed and the tick function of the aircraft itself will not run if the aircraft has landed. When the aircraft lands, the catchILS() function will also unsubscribe the aircraft from the ILS using the unsubscribe(...) method.

startTurnILS(dt_sec) returns true when the aircraft should start turning to catch the ILS. The turn radius is calculated and the method derives whether the aircraft has passed the point where a turn in would result in it being lined up with the ILS.
isILSCatchPoint(x, y, alt) returns true if the point provided by the x-coordinate in x, the y-coordinate in y and the altitude in alt is a valid point to catch the ILS. As such, the algorithm assesses whether the point is lined up with the ILS and whether the altitude is not above the ILS at that specific coordinate.

3.1.5. getLeg Methods

There are multiple getLeg(...) methods, each one of them uses an initial state, provided in a single datalog point (ArrayList with floats) called startPos and the integration timestep, dt_sec. All getLeg(...) return a datalog for their portion of the path.

getLeg12(dt_sec, startPos, angTTBExtra) uses the integrateStraightPath(...) to integrate the constant heading angle path of leg 1 and consecutively it uses integrateCurvedPath(...) for the varying heading angle path for leg 2. The integrateStraightPath(...) method is stopped as soon as it reaches the TTB distance (variable TTBDist), set by the user or the updateTTBDisplayData(...) method. The point where this happened is stored in a local variable TTBPoint, which is used in the visualisation of the TTB angle. If the aircraft has already crossed its TTB distance, only leg 2 will be integrated in this method, since the algorithm assumes a TTB will happen for that timestep. For leg 2, the integrateCurvedPath(...) method will be stopped as soon as the aircraft TTB angle is reached, which is a heading angle of 90 with respect to the ILS, plus the extra TTB angle, provided by angTTBExtra, which is a local variable that is changed by the user and reset as soon as the aircraft starts a turn.

getLeg2(dt_sec, startPos, angTTBExtra) basically does the same as getLeg12(...), except for that is does not take leg 1 into account. This method is used for building the data log with a datalog for leg 1 already provided, which is the case when the user hovers over a TTB point with the computer mouse.

getLeg34(dt_sec, startPos) operates much in the same way as the previous getLeg(...) methods but now for legs 3 and 4. The only differences are that the assumed TTI angle now is 30 with respect to the ILS, that the angTTIExtra is used instead of the angTTBExtra, that the TTI Distance, stored in the local variable TTIDist, is used and that the TTI point is set in local variable TTIPoint.

getLeg4(dt_sec, startPos, angTTIExtra) basically does the same as getLeg34(...), except for that is does not take leg 3 into account. This method is used for building the data log with a datalog for leg 3 already provided, which is the case when the user hovers over a TTI point with the computer mouse.

getLeg56(dt_sec, startPos) operates much in the same way as the previous getLeg(...) methods but now for legs 5 and 6. The major difference is that the point at which the aircraft turns to catch the ILS is derived iteratively by the method iterateCurveOnStraight(...), if the aircraft is in leg 6. The integration stops as soon as the heading angle is equal to the angle of the ILS.

getLeg78(dt_sec, startPos) integrates leg 7, which is split into leg 7 and 8 in the code. Here, leg 7 is the leg from ILS catch to 4DME and leg 8 is the leg from 4DME to the runway threshold. This method uses the integrateStraightILS(...) method for the integration, which integrates a path with a constant flight path angle. As such, the ILS path is integrated with a heading angle negating the effects of wind, keeping the aircraft on the ILS.

3.1.6. Path Integration Methods

integrateStraightPath(mode, dt_sec, x_start, y_start, alt_start, IAS_start, HDG_start, alt_end, IAS_end, t_start, dist_stop_nm, keepFirst) integrates the path of an aircraft for a constant heading angle. It does this starting from the state provided in the variables x_start, y_start, alt_start, IAS_start and HDG_start which are the location in x in NM, location in y in NM, altitude in ft and heading in degrees, respectively. The distance covered at which the integrator should stop is governed by dist_stop_nm and in which way this distance is used is governed by the mode variable, which is an integer ranging from 1-5. In the integration, the getWindVec_kts(...) function from the Wind class is used to obtain the speed and direction of the wind at that point. For each timestep, the averages of all variables are used to integrate the states. This is done because the timestep may be quite large (5 seconds) in ATC simulation. For the purpose of landing accurate time prediction, for all integrators a separate dt_sec is used of 1 second. The keepFirst variable is a boolean stating if the first point in the integration should be kept or discarded, as the position of the integrated piece in the total path dictates different requirements for this.

Mode 1 for this method stops the integration if the distance from the first integration point to the last is larger than dist_stop_nm. Mode 2 quits integrating as soon as the length of the vector from the starting point to the last point, projected on the ILS is larger than dist_stop_nm. The third mode stops the integration as soon as the distance from the runway threshold to the integration point location on the ILS is larger than dist_stop_nm. The fourth method stops the integration as soon as the distance of the integration point to its projection on the ILS is smaller than a certain distance. The fifth mode stops integrating as soon as a projection on the ILS of a point is either 25 NM away from the runway threshold or smaller than 6 NM, used for the TTB display.

integrateStraightPathILS(dt_sec, x_start, y_start, alt_start, IAS_start, HDG_start, IAS_end, t_start, dist_stop_nm, keepFirst) integrates the straight path on the ILS. This method differs from the main integrateStraightPath(...) method since the heading angle for this flight path is set such that the aircraft travels in a straight line i.e. on the ILS. The IAS_end also does not have to be set since altitude is governed by the ILS at this point.

integrateCurvedPath(mode, dt_sec, x_start, y_start, alt_start, IAS_start, HDG_start, alt_end, IAS_end, HDG_end, t_start, HDG_stop, keepFirst) integrates the path of an aircraft for a varying heading angle. The heading angle at which the integrator should stop is governed by HDG_stop. The rate at which the heading angle changes is governed by the maximum bank angle of the aircraft.

iterateCurveOnStraight(dt_sec, dataLog1, alt_end, IAS_end, HDG_end) again integrates a curved path in the same way as the integrateCurvedPath(...) method does. A difference is that it takes the last point of a datalog which is put in (dataLog1), integrates a curved trajectory on that and determines the distance between that point and the ILS. It then tries other turn-in points to derive the optimal turn-in point; it will pick the turn-in point on the input datalog where the aircraft catches the ILS best.

3.1.7. Miscellaneous Methods

mergeDataLogs(dataLog1, dataLog2) returns a datalog consisting of dataLog1 and dataLog2 combined. It is used top combine all the pieces of datalog to form one total path, for instance in the updateTP(...) method.

isSeparated(dataLog, leg_min) returns true if the aircraft is separated with all other aircraft flying in a leg after leg_min, if the aircraft were to travel with the path provided in dataLog. The function checks all aircraft in the scenario and, if that aircraft has not landed and has a leg number higher than leg_min, separation is checked using the function checkSeparation(..). If a conflict is predicted, the method puts the conflicting aircraft in acConflictList, a local ArrayList with all intruding aircraft that will be in conflict with the aircraft calling isSeparated(...). This list is used when highlighting the conflict aircraft when hovering over an unfeasible TTB point with the computer mouse.

checkSeparation(acInt dataLog, callTime) returns true if the aircraft calling this method is separated from aircraft acInt, if the calling aircraft will travel using the path defined in dataLog. The method obtains the landing time for both aircraft, as wel as the minimum separation applying for this aircraft pair using the getTBSTime(...) method. Then, the landing times of both aircraft are checked to see if the aircraft are separated. Here, if the landing time of one aircraft is smaller than the other, it taken to be the leading aircraft of the pair. The callTime variable is used as a failsafe and troubleshooting variable, where it represents the amount of times the function has recursively called itself.

getLegOfApproach() returns the leg the aircraft is currently in as an integer. Returns 0 if the leg of flight can not be identified. This can be for numerous reasons; the aircraft flies away from the ILS, the aircraft is outside of the sector or the predicted approach is deemed unfeasible.

setOverruleHighlighting(bool) sets the internal boolean l_overrule_highlighting to the value given in bool. When this variable is true, mouse movement will not affect aircraft highlighting in the GLAircraft class.

3.2. ILS class

The ILS class is an object with a position, a heading angle and a length. These variables can be set and retrieved using multiple setters and getters. Some frequently used vectors are defined as well, such as the vector locating the base of the ILS and the vector from the base to the top of the ILS. The ILS is drawn in the GLILS class.

The methods with which the aircraft catches and follows the ILS are located in the Aircraft class. However, the ILS class does have one extra function; it is an information hub where all data for all aircraft is known, such that aircraft can obtain approach-crucial information from the ILS class. The main piece of information being the landing order of aircraft.

3.2.1. Landing Order Methods

subscribe(ac) subscribes the aircraft ac to the ILS. This means that the predicted landing time of the aircraft will be stored in a locally stored ArrayList called landTimeList and the aircraft itself will be saved in a locally stored ArrayList called acList. All aircraft are subscribed to the ILS on their first tick and as such these two lists will contain multiple landing times and aircraft. The subscribe(...) method then calls the sortByLandTime() method, which sorts both lists by landing time.

unsubscribe(ac) unsubscribes aircraft ac from the ILS; it removes it and its landing time from acList and landTimeList, respectively. This function is called by the aircraft when it lands.

sortByLandTime() sorts the locally stored lists landTimeList and acList. Here, the first aircraft to land first will be at the top of the list. The sorting is done using a bubble sorting algorithm. The method does not return both lists, it simply alters them.

sortByLandTime(acList, landTimeList) is equal to the sortByLandTime() function, except that it uses the acList and landTimeList that the user put i. This is used when making fictional lists; for instance for previewing a situation where an aircraft gets another landing time (hovering with computer mouse). The sortByLandTime(acList, landTimeList) method sorts the input lists and returns the acList.

getLandingOrder(ACID) returns the index of the aircraft with name ACID in the acList.

updateLandTime(ac) updates the landing time in landTimeList for aircraft ac. It uses getLandingOrder(...) to obtain the position of the aircraft in acList and alters the value at this location in landTimeList to the landing time of the aircraft. After an update, the lists are sorted using sortByLandTime().

getLeadAcIf(acIn, lead, tLand) returns the aircraft that is predicted to land before the aircraft acIn, if acIn would have landing time tLand. In this prediction, only aircraft that are in leg 2 or further are considered. The lead variable governs how many places in front of acInt the to be returned aircraft must be e.g. for the leading aircraft a lead of 1 is used and for the aircraft in front of the leading aircraft a lead of 2 is used.

getFollowAcLeg2Min(acIn) returns the first aircraft that will land after aircraft acIn, taking only aircrft in leg 2 or further into consideration.

getLeadAcIf(acIn, follow, tLand) returns the aircraft that is predicted to land after the aircraft acIn, if acIn would have landing time tLand. In this prediction, only aircraft that are in leg 2 or further are considered. The follow variable governs how many behind acInt the to be returned aircraft must be.

3.2.2. Miscellaneous Methods

setHighlighted(bool) sets the internal boolean highlighted to the value given in bool. When this boolean is true, the GLILS class will draw a larger, yellow ILS.

3.3. GLILS class

The GLILS class was added top draw the ILS.

autoPaintScaled(drawable) draws the ILS using the GLAutoDrawable ‘drawable’. The color is set to green, except for when the highlighted boolean for the ILS is set to true, in which case the ILS will be drawn in yellow.

3.4. GLITIP class

The GLITIP class draws all the ITIP display data on the screen. Each Aircraft is linked to a GLITIP class. All the computations are done in the Aircraft class to minimise processor load as the graphics refresh rate is very high. In the code, distinction is made between the TTB and TTI part of the display. All methods with the same name except for ’TTB’ or ’TTI’ have exactly the same functions but for their respective part of the dipslay.

autoPaintScaled(drawable) draws the display data using the GLAutoDrawable ‘drawable’. It is called by the autoPaintScaled(...) method located in the GLAircraft class if and only if the aircraft is selected. The method first draws the path of the TP using the drawTrajectory(...) method and the Aircraft tpDataLog. Then, the TTB or TTI display is drawn using the drawDisplay(...) method, depending on the leg of flight. The trajectory resulting from the user hovering over a TTB or TTI point is also drawn using the drawHoverTrajectory(...) method, depending on the local booleans drawTTBTrajectory and drawTTITrajectory which are set in the resetDisplay() method. Next, all conflict aircraft listed in the ArrayList highlightedAcList are highlighted and finally, the TBS marker behind the leading aircraft and behind itself are drawn using drawTBSLeadMarker(...) and drawTBSFollowMarker, respectively.

drawTrajectory(dataLog, gl) draws the path stored in the dataLog variable on the screen using using the Graphics Library interface defined by gl.

drawDisplay(landTimeList, dataLog, poly, gl) draws the TTB or TTI dots using the Graphics Library interface defined by gl. It does this by drawing a point at every point in the dataLog variable, using the separation information supplied in landTimeList to correctly colour the dots. During the drawing, a polygon around all the drawn points is stored in the variable supplied by poly. This polygon is used for assessing whether the mouse is hovering over one of the drawn points.

resetDisplay() is a method that updates local booleans such that the right paths are drawn and the right aircraft are highlighted. Using the mouseInTTB() and mouseInTTI() methods, the method checks whether the computer mouse is hovering over green or red points from the ITIP display and stores whether a trajectory should be drawn in the local booleans drawTTBTrajectory and drawTTITrajectory. When a path resulting from a mouse hover should be drawn, the updateHoverLog() method is ran. The resetDisplay() method is ran every time the user moves the mouse (in the mouseMoved(...) method) or scrolls (in the mouseWheelMoved(...) method) or clicks (in the mousePressed(...) method), and for every aircraft tick. The local booleans defined in this function are used by the autoPaintScaled(...) method to determine which path to draw.

updateHoverLog() stores the datalog that should be drawn due to the mouse hovering over a TTB/TTI point in the local ArrayLists drawTTBDataLog or drawTTIDataLog. Depending on the value of drawTTBTrajectory or drawTTITrajectory, this datalog is then drawn in the autoPaintScaled(...) method. The method also defines the colour of the trajectory to be drawn, depending on path feasibility checks from the Aircraft.isSeparated(...) and isILSCatchPoint() methods. If the trajectory drawn is feasible, the corresponding boolean TTBLogGreen or TTILogGreen will be set to true. If the trajectory is unfeasible due to separation problems, this method stores all conflicting aircraft in the local ArrayList highlightedAcList. The autoPaintScaled(...) method highlights all aircraft in the highlightedAcList and sets their overruleHighlighting to true, to avoid conflicting highlighting commands. When the trajectory is unfeasible due to the ILS catch point, the boolean highlight_ILS is set to true. The autoPaintScaled(...) method highlights the ILS when this boolean is true.

drawHoverTrajectory(gl) draws the trajectory in drawTTBDataLog or drawTTIDataLog, depending on the booleans drawTTBTrajectory and drawTTITrajectory, using the Graphics Library interface defined by gl. The colour of the trajectories is defined in this method as well, depending on the values of either TTBLogGreen or TTILogGreen. The trajectory is drawn using the drawTrajectory(...) method.

isILSCatchPoint(x, y, alt) checks whether the point defined by the x-coordinate x in NM, the y-coordinate y in NM and the altitude alt in ft is a valid point to catch the ILS.

mouseInTTB() and **mouseInTTI**() return true when the computer mouse location is inside the polygon defined in local variable TTBPoly or TTIPoly, respectively. The TTBPoly and TTIPoly polygons are set when drawing the TTB or TTI display in the drawDisplay(...) method.

updateGhosts(dataLog) stores all data needed for the drawing of the ghost in local variables. The state of the aircraft flying with the path provided in the dataLog variable at the moment its leading aircraft lands is stored for use in the autoPaintProjected(...) method where the ghost aircraft is drawn. If the aircraft has a leading aircraft when it would fly with the path provided in dataLog, the boolean drawTBSMarkerLead is set to true. When the aircraft has a follower aircraft, the boolean drawTBSMarkerFollow is set to true, and the location of the follower aircraft at the moment the calling aircraft lands is stored. Note that follow-ghosts are not included in the final display design.

drawTBSLeadMarker(gl) draws the TBS marker before the runway threshold for when a ghost is plotted using the Graphics Library interface defined by gl. This method is only activated in autoPaintScaled() when the local boolean drawTBSMarkerLead is true, thus only drawing the TBS marker when a ghost is visualised.

drawTBSFollowMarker(gl) draws the TBS marker behind the ghost to separate the follower aircraft, using the Graphics Library interface defined by gl. This method is only activated in autoPaintScaled() when the local boolean drawTBSMarkerFollow is set to true, thus only drawing the TBS marker when a ghost is visualised.

autoPaintProjected(drawable) is the method where the ghost aircraft gets drawn. It is activated by the autoPaintProjected(...) method in the GLAircraft class. The colour of the ghost is set using the local booleans TTBLogGreen and TTILogGreen. The actual drawing of the ghost happens in the drawGhost(...) method.

drawGhost(drawable, gl, x, y, HDG_DEG) draws a ghost aircraft at the location specified by x and y in NM with a heading given by HDG_DEG, using the Graphics Library interface defined by gl and the GLAutoDrawable provided in drawable.

drawHoverDisplayInfo(gl, GLAD) draws the turn-in marker and information at the mouse location when the mouse is hovering over a turn-in point, using the Graphics Library interface defined by gl and the GLAutoDrawable provided in GLAD. The location of the drawing is found using the findDataLogIndex(...) method. The text is drawn at a location depending on the turn-in heading to declutter the display.

drawTPDisplayInfo(gl, GLAD) draws the turn-in marker and information at the predicted turn-in location, using the Graphics Library interface defined by gl and the GLAutoDrawable provided in GLAD. The location of the drawing is found using getTTBPoint() or getTTIPoint() method. The text is drawn at a location depending on the turn-in heading to declutter the display.

findDataLogIndex(dataLog) returns the index of the point in dataLog where the computer mouse is closest to. This is used when locating the point where a turn-in should be drawn when hovering over a turn-in point.

terminateDisplay() empties all values and lists for the display. It also sets all aircraft highlighting and overruleHighlighting parameters to false. This method is called when the aircraft gets deselected.

3.5. Wind class

The wind class provides the wind at a specific location. In this version of the simulator, only uniform wind is used. However, the algorithm is built in such a way that implementation of non-uniform wind is relatively simple; all inputs in the functions are correct, only the wind model function has to be written in his class.

setWind(x, y) sets a uniform wind vector with the x part of the vector provided by variable x and the y part by variable y.

getWindVec_kts(x_nm, y_nm, alt_ft, time_s) returns the wind at the location put in by x_nm, y_nm and alt_ft, at time_s seconds in the future. In this version, for all inputs the same uniform wind vector in kts is returned. However, all methods in the aircraft class that use this method already implement all right inputs. As such, should one wish to implement non-uniform wind, only this method has to be adapted, making use of these inputs.