MODE-S RADAR INTERROGATION ALGORITHM DESIGN, SIMULATION ENVIRONMENT SETUP AND ERROR CORRECTION

by

Ahmet Günhan Aydın

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2021

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2023

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Assoc. Prof. Ahmet Oncu, for allowing me to conduct research, for spending long office hours with me to improve my academic work, and for providing invaluable guidance throughout this thesis.

I am also grateful for the support of my colleagues Yunus Emre Erdoğan and Artun Akdoğan. Without their support, most things would be more difficult.

I also would like to express my special thanks to Professor Ali Emre Pusane and Assoc. Prof. Özgür Özdemir for their invaluable support and being a member of my thesis jury.

I would also like to express my deepest gratitude to my family, who stood by me in every decision I made and provided full support.

This work was partially supported by Boğaziçi University Research Fund Grant Number 19002, Boğaziçi University Research Fund Grant Number 14320, and the research work under the research contract between the Bogazici University Technology Transfer Office and the Scientific and Technological Research Institution of Turkey, Informatics and Information Security Research Center.

ABSTRACT

MODE-S RADAR INTERROGATION ALGORITHM DESIGN, SIMULATION ENVIRONMENT SETUP AND ERROR CORRECTION

The air traffic density was relatively low back in the day compared to today's dense air traffic environment. This increasing trend in air traffic density will continue in the near future with the addition of different aerial vehicles. Before the Mode-S protocol, Mode A and Mode C were in use; however, the Mode A/C configuration was only usable in sparsely dense air traffic. One of the useful features of Mode-S is the ability of probabilistic interrogation. However, there has not yet been a sophisticated algorithm for closely located large numbers of aircraft. Considering a futuristic air environment with a swarm of drones and airbuses equipped with transponders, the probabilistic interrogation feature of Mode-S is utilized, and an algorithm is designed. The proposed algorithm is able to collect close aircraft information in a relatively short time. There has also been created a high-level Mode-S uplink and downlink communication simulator in order to exchange all-call communication and record the algorithm's performance in terms of time and number of interrogations sent. Furthermore, error correction of erroneous messages, usually from garbling and noise, is investigated. An error correction method is developed and tested with real-time data coming from commercial aircraft.

ÖZET

MODE-S RADARLARI İÇİN SORGULAMA ALGORİTMASI TASARIMI, SİMULASYON ORTAMI KURULUMU VE HATA DOĞRULTMASI

Hava trafik yoğunluğu, günümüzün yoğun hava trafiği ortamına kıyasla, eskiden çok daha düşüktü. Bu hava trafiğindeki artış eğilimi yakın gelecekte farklı hava araçlarının da eklenmesiyle devam edecektir. Mode-S protokolünden önce, Mode A ve Mode C kullanımdaydı. Ancak, Mod A/C konfigürasyonu yalnızca seyrek yoğun hava trafiğinde kullanılabilirdi. Mode-S'nin kullanışlı özelliklerinden biri, olasılıksal sorgulama yeteneğidir. Ancak, yakında çok sayıda uçak bulunduğu durum için henüz sofistike bir algoritma bulunmamaktadır. Aktarıcılar ile donatılmış insansız hava araçları ve have otobüslerinden oluşan fütüristik bir hava ortamını göz önünde bulundurarak, Mode-S'nin olasılıksal sorgulama özelliği kullanıldı ve bir algoritma tasarlandı. Önerilen algoritma, yakın uçak bilgilerini nispeten kısa bir sürede toplayabilmektedir. Ayrıca, tüm sorgu iletişimini simüle etmek, algoritmanın performansını zaman ve gönderilen sorgulama sayısı açısından kaydetmek için bir Mode-S yer-uydu bağı ve uydu-yer bağı iletişim simülatörü oluşturulmuştur. Ayrıca, genellikle çarpışma ve gürültüden kaynaklanan hatalı mesajların doğrultulması incelenmiştir. Ticari uçaklardan gelen gerçek verilerle bir hata düzeltme yöntemi geliştirilmiş ve test edilmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES viii
LIST OF TABLES x
LIST OF SYMBOLS xii
LIST OF ACRONYMS/ABBREVIATIONS xiii
1. INTRODUCTION 1
1.1. Mode-S Interrogation Signal Format 4
1.2. Mode-S ADS-B General Message Structure 6
1.3. Mode-S UF/DF and BDS Types 8
2. MODELLING THE SIMULATION SYSTEM 10
2.1. Aircraft Scenarios in a Beam-width 13
2.1.1. One Aircraft in a Beam 13
2.1.2. Multiple Close Aircraft in The Same Beam 15
2.1.3. Multiple Distant Aircraft in The Same Beam 15
2.1.4. A Sample Scenario Simulation 16
2.2. Interrogation Sequences 18
2.2.1. UF 11 20
2.2.2. DF 11 21
2.2.3. UF 4 22
2.2.4. DF 4 23
2.2.5. UF 5 23
2.2.6. DF 5 24
2.2.7. UF 20 25
2.2.8. DF 20 26
2.2.9. UF 21 27
2.2.10. DF 21 27

2.2.11. Overall Summary of Interrogation Sequences 27
2.3. Results of Radar Simulation 28
2.4. Real-Time System Model 31
2.4.1. RF Front-End 33
2.4.2. CRC Signal Generation 34
2.4.3. CRC Signal Check and Test 36
2.4.4. Testing the CRC Process with Collected Data 37
2.4.5. Related Works on Error Correction Using CRC 39
2.4.6. Developed Error Correction Method Using CRC 42
3. MODE-S RADAR INTERROGATION ALGORITHM DESIGN 47
3.1. Selectivity and Garbling 47
3.2. Stochastic Interrogation Algorithm 48
3.3. Performance of the Algorithm 50
3.4. Effects of Basic Radar Specifications on Performance 53
3.5. Limited Lockout 55
3.6. Experimental Results 57
4. CONCLUSION 58
REFERENCES 60
APPENDIX A: LIST OF RELATED PUBLICATIONS 65

LIST OF FIGURES

Figure 1.1. Mode-S and Mode A/C interrogations and replies 2

Figure 1.2. Mode-S uplink pulse sequence 5

Figure 1.3. Mode A/C/S all-call interrogation format 6

Figure 1.4. Comm-B message types 9

Figure 2.1. Communication scheme between interrogator and transponder. . . 10

Figure 2.2. Azimuth space divided by antenna into M slices. 11

Figure 2.3. Dividing time into periods in an antenna 3-dB beam-width duration. 12

Figure 2.4. All-call and roll-call interrogation illustration. 13

Figure 2.5. Time-flow chart for a single transponder case. 14

Figure 2.6. Time-flow chart for two transponders flying close to each other in succession. 15

Figure 2.7. Time-flow chart for two transponders flying distant to each other in succession. 16

Figure 2.8. Illustration of a sample scenario. 17

Figure 2.9. Radar simulator GUI. 19

Figure 2.10. Radar simulator GUI with outputs. 29

Figure 2.11. Radar simulator output variables. 30

Figure 2.12. An example of output structure array. 31

Figure 2.13. Connection scheme. 33

Figure 2.14. Component images. 33

Figure 2.15. Real-time collection of 56-bit short data. 38

Figure 2.16. Real-time collection of 112-bit long data. 38

Figure 2.17. The flow diagram of the second method for error correction. . . . 45

Figure 3.1. Adaptive probabilistic interrogation algorithm. 51

Figure 3.2. Overall performance of adaptive algorithm vs static interrogations. 52

Figure 3.3. Effect of beam-width on the performance of the Adaptive Algorithm. 53

Figure 3.4. Effect of PRF on the performance of the Adaptive Algorithm. . . 53

Figure 3.5. Effect of RPM on the performance of the Adaptive Algorithm. . . 54

Figure 3.6. The performance of the algorithm and static methods with selected parameters, i.e., PRF: 150 Hz, BW: 2.4°, and RPM: 6 min^{-1}. . . 55

Figure 3.7. The performance of the algorithm and static methods for limited lockout case with PRF: 150 Hz, BW: 2.4°, and RPM: 6 min^{-1}. . . 56

LIST OF TABLES

Table 1.1. Mode-S ADS-B message structure 7

Table 1.2. Type codes and their corresponding contents 7

Table 1.3. UF/DF types of Mode-S 8

Table 2.1. Preset aircraft parameters. 18

Table 2.2. 56-bit UF 11 short message structure. 20

Table 2.3. Probability of reply with corresponding codes 21

Table 2.4. 56-bit DF 11 short message structure. 21

Table 2.5. CA information 22

Table 2.6. 56-bit UF 4 short message structure. 22

Table 2.7. 56-bit DF 4 short message structure. 23

Table 2.8. 56-bit UF 5 short message structure. 24

Table 2.9. 56-bit DF 5 short message structure. 24

Table 2.10. 112-bit UF 20 long message structure. 25

Table 2.11. Default selections for UF 20 long message example. 25

Table 2.12. 112-bit DF 20 long message structure. 26

Table 2.13. Default selections for DF 20 long message example. 26

Table 2.14. 112-bit UF 21 long message structure. 27

Table 2.15. 112-bit DF 21 long message structure. 27

Table 2.16. Produced formats of the sample aircraft. 28

Table 2.17. Sample message acquisitions of the sample aircraft. 28

Table 2.18. Component list. 33

Table 2.19. 56-bit (14-hex) ADS-B short message. 35

Table 2.20. 112-bit (28-Hex) ADS-B long message. 35

Table 2.21. Raw message and CRC separation. 39

Table 2.22. The results of the second method for error correction. 46

Table 3.1. Basic Mode-S SSR specifications that are used in simulations. . . . 50

Table 3.2. Number of interrogations to detect aircraft for the constant 0.5, 0.25, 0.125, and 0.0625 probability methods and the proposed algorithm. 57

LIST OF SYMBOLS

$G(x)$	Generator polynomial
$M(x)$	ADS-B message polynomial
n_{ant}	Antenna rotation speed in rpm
$P(x)$	The remainder of polynomial division
R_1	Distance between an interrogator and a transponder
t_h	3-dB beam-width duration in seconds
T_{ci}	Total time spent for complete interrogation
T_p	Time spent per pulse
x^i	Term representing the message bit
φ_{3dB}	3-dB beam-width in degrees
τ_a	Signal reception and processing time at transponder
τ_{cp}	Reply message duration
τ_d	Time the message spends in the air in one direction
τ_s	Signal reception and processing time at interrogator
τ_{sp}	Interrogation message duration

LIST OF ACRONYMS/ABBREVIATIONS

ADS-B	Automatic Dependent Surveillance Broadcast
ATC	Air Traffic Control
ATCRBS	Air Traffic Control Radar Beacon System
BDS	Comm-B Data Selector
BW	Beam-width
CA	Capability
CRC	Cyclic Redundancy Check
DF	Downlink Format
DPSK	Differential Phase Shift Keying
EHS	Enhanced Surveillance
ELS	Elementary Surveillance
GS	Ground Station
IC	Interrogator Code
ICAO	International Civil Aviation Organization
LNA	Low Noise Amplifier
MHR	Meteorological Hazard Report
MRAR	Meteorological Routine Air Report
NM	Nautical Mile
PPM	Pulse Position Modulation
PR	Probability of Reply
PRF	Pulse Repetition Frequency
RPM	Revolutions per Minute
SDR	Software Defined Radio
SI	Surveillance Identifier
SSR	Secondary Surveillance Radar
UF	Uplink Format

1. INTRODUCTION

Radar (Radio Detection and Ranging) is a tool that uses radio waves to detect surrounding objects. For example, radars can detect objects in the maritime environment, such as ships and buoys. The direction of detection and distance from the source can be determined with high precision using short-wavelength microwaves [1]. Radar operates on a similar concept to how the sound-wave reflection works. Radio frequency (RF) energy is transmitted and it is reflected from a reflecting object. The radar set receives a small part of the reflected energy. This returned energy is called echo, just as in audio terminology. Radar sets use echo to determine the direction and distance of the reflected object [2].

With the growing air traffic, the demand for radar surveillance systems is increasing. These systems are used for controlling air traffic. Some people argue that current systems have already reached their limits. Mode-S (Select) ADS-B (Automatic Dependent Surveillance-Broadcast) is a significant technological improvement and is considered a more efficient replacement for radar in the aviation industry. Mode-S is the next-generation SSR (Secondary Surveillance Radar) system. Secondary surveillance radar is a radar system based on targets equipped with a radar transponder, unlike primary radar systems used in air traffic control (ATC), which measure the direction and distance of targets using the detected reflections of radio signals. These targets respond to each interrogation signal by transmitting encoded data such as the identification code, the aircraft's altitude, and other information depending on the mode selected [3].

Mode-S is designed to help avoid excessive interrogation of the transponder due to the large number of radars in busy areas and to provide automatic collision avoidance. And the key to the need for compatibility with legacy modes was the decision to use the same frequencies for Mode-S as legacy modes of the SSR, namely 1030 MHz for interrogation and 1090 MHz for the response as shown in Figure 1.1.

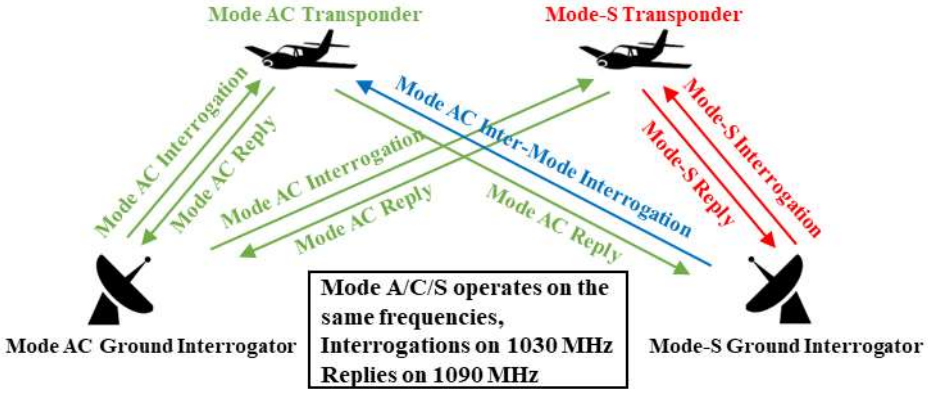


Figure 1.1. Mode-S and Mode A/C interrogations and replies [4].

Prior to the Mode-S protocol, Mode A and Mode C were in use; however, this configuration was only usable in sparsely dense air traffic [5]. The reason is that all SSR replies stack in the same frequency band; hence all aircraft signals being detected in the same antenna beam overlap with each other, and this phenomenon is known as garbling in general. When the 1090 MHz band is shared with both the Mode-S, ADS-B, and Mode A/C systems, there occurs FRUIT (false replies unsynchronized in time) which is co-channel interference due to dense environments [6].

Mode A and Mode C give information about interrogated aircraft's squawk code (4 octal digits) and altitude, respectively [7]. Besides mentioned traditional SSR abilities, Mode-S provides enhanced surveillance and datalink to ground control stations to be able to monitor aircraft with more capability [8]. ADS-B technology allows an aircraft to broadcast its information such as airspeed, altitude, flight ID, and coordinates.

Even operations with Mode-S systems cannot avoid garbling problems in the dense air traffic regions in terms of traffic since there occur close encounters between different aircraft within a close range, which is five nautical miles usual standard for horizontal separation, and around 2000 feet ($\sim$ 0.38 NM) for vertical separation [9]. These close scenarios can be caused by different sources such as lack of capacity of the coverage, weather conditions, military activities, airline decisions, etc. [10]. Another

reason for having a highly dense air environment with close encounters is that there is an immense increase in air traffic demand and the number of aircraft registered. According to an annual report by EUROCONTROL, total flights will increase by an average of 1.9% per year over the next years, reaching a total of 11.6 million flights in 2023 [11]. Furthermore, forecasted IFR (instrument flight rules) movements per traffic zone tell that there is an upward trend in all states of the eurozone states in terms of air traffic density through 2024 [12]. Another reason for seeing many more aircraft on the skyline is that the cost of air traffic management is relatively disconnected from the air traffic, and this allows for the introduction of more registered aircraft [13]. Speaking of registered aircraft, the EU continues to develop the controlled inclusion of unmanned aerial vehicles in the air environment with its U-space project. This project offers UAS (unmanned aircraft systems) traffic management (UTM) to integrate UAS into air traffic management (ATM) [14].

There have been numerous research parallel to the increasing number of transponder-equipped drones for the last decade to provide air space safety [15–17]. An increase in registered aircraft, including unmanned aerial vehicles, forces authorities to revise regulations such as RPAS (remotely piloted aircraft systems) [18]. The number of drones in urban and rural airspace also will amplify considering passenger transportation [19], package delivery [20], agricultural surveillance and air cargo [21]. On the military aspect, border monitoring, landmine detection, logistic security, and military mission delivery require more drone injection into the airspace [22, 23].

Therefore, the future of air traffic will be crowded with UAS inclusion, and this requires new techniques and utilities in communication systems in order to detect aircraft in the dense air traffic regions rapidly. At this stage, in order to reduce the time taken to detect aircraft in garbled replies, a probabilistic interrogation algorithm using already existing Mode-S features is proposed. In this approach, all-call interrogations are used with different probabilities in the uplink and interrogated with 1, 0.5, 0.25, 0.125, and 0.0625 probabilities according to the proposed adaptive algorithm. By adaptively interrogating, a ground station (GS) operator can gather ICAO information

of aircraft relatively fast in the dense garbled region from all-call replies. A different method can be interleaving scheduling for Mode-S transactions [24]. Inserting additional waiting times can diminish the heavy load of the dense air traffic in the all-call interrogation process. However, additional time and energy trade-offs appear as well.

At the same time, a system simulation model has been developed to test the algorithm in different air traffic environments. This simulation model has been a comprehensive study of SSR radars. While developing the model, a real-time data collection system was established simultaneously, allowing collected data to be used in the simulation. Since the amount of error in the collected data is high due to the collision, garbling, and noise, the error correction method has also been developed.

Before proceeding into the system simulation model and proposed stochastic interrogation algorithm for air traffic control radar systems, the basic structures and fundamental formats of Mode-S systems are presented in the following sections.

1.1. Mode-S Interrogation Signal Format

Lincoln Labs researchers used a side lobe suppression technique, which makes ATCRBS (Air traffic control radar beacon system) transponders inoperative when Mode-S signals are sent, enabling Mode-S signals to be transparent [25]. A Mode-S interrogation includes the side lobe suppression preamble, in which P_2 is equal to or stronger than P_1, followed by a data block containing either 56 or 112 bits of data. For interrogation, prevention of interference with previous equipment is achieved by transmitting SSR side lobe suppression pulses P_1 and P_2 as the start of interrogation. The first two pulses, P_1 and P_2, are 0.8 μs in duration and are 2.0 μs apart to simulate side lobe suppression pulses of Mode A and C interrogation [7]. For non-Mode-S transponders, the P_2 pulse serves as side lobe suppression, so, they do not attempt to decode the rest of the interrogation. The synchronizing phase reversal is used as a time reference to transmit the response in the same way as using the P_3 pulse in Modes A and C. The pulse sequence of uplink Mode-S interrogation can be seen in Figure 1.2.

Data is transmitted by differential phase shift keying (DPSK) during interrogation, where binary 1 is indicated with a phase change of 180 and binary 0 is without phase change. The gap between the phase reversal positions is 0.25 μs to give a data bitrate of 4 MHz. The total number of data bits transmitted is 56 or 112 depending on the requirements [26].

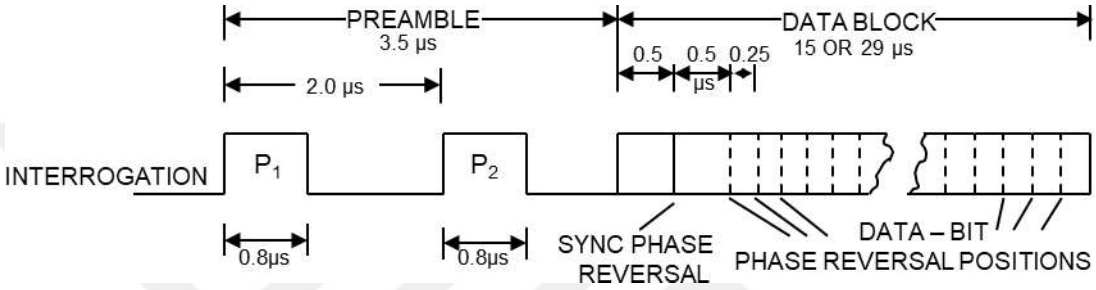


Figure 1.2. Mode-S uplink pulse sequence [27].

An ATCRBS consisting of a rotating antenna and transponders typically sends an all-call interrogation around its environment. Replies to these all-call interrogations include the aircraft identity and the altitude, as usual. In traditional Mode A/C protocols, there are two pulses, namely P_1 and P_3, separated by 8 or 21 μs according to their length. When designing Mode-S, the researchers used a similar format for the signal sent by Mode-S sensors that only request a response from Mode-S equipped aircraft. A P_4 pulse is added to distinguish Mode-S all-call interrogations from Mode A/C ones [27]. The duration of the P_4 pulse is 0.8μs or 1.6μs. When the duration of the P_4 pulse is 0.8μs, only aircraft with a Mode A/C transponder respond, and aircraft with a Mode-S transponder remain silent. Thus, aircraft without Mode-S equipment can be easily detected.

An aircraft reply to the interrogation is modulated via pulse position modulation at 1090 MHz frequency, and the pulse sequence contains an 8-μs preamble followed by 56 (short) or 112 (long) μs data block. A 16-bit preamble is fixed and always "1010000101000000" [7].

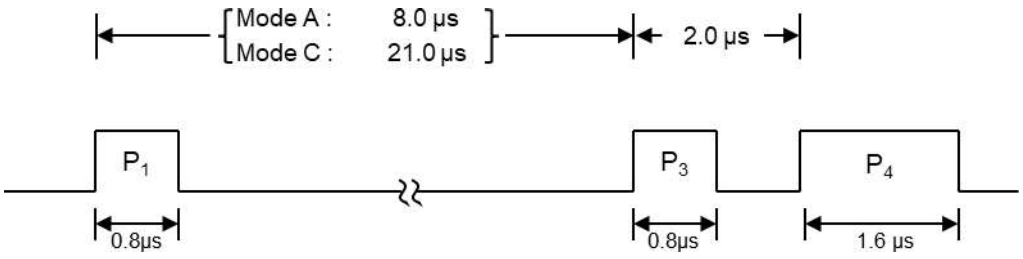


Figure 1.3. Mode A/C/S all-call interrogation format [27].

Aircraft with ATCRBS transponders will respond with a normal ATCRBS all-call reply, ignoring the last blow. Mode-S transponders see the additional pulse and recognize the signal as a Mode-S challenge and respond with a Mode-S reply. Mode-S equipped aircraft can not differentiate ATCRBS and Mode-S interrogation without this approach. They do not know which answer to transmit and so can not function with different sensors on the ground [25].

1.2. Mode-S ADS-B General Message Structure

In fact, the starting point of the whole thesis and the study is Mode-S ADS-B messages, and the collected data of these messages was reached at first from aircraft. In this section, the Mode-S ADS-B message structure is briefly explained. In Chapter 2, message structures for interrogation and reply signals will be examined in more detail.

The uplink or downlink format is defined in the first five bits of the message. Information will be given about downlink formatted messages, especially 112-bit extended squitter messages. This extended squitter type corresponds to DF number 17, which is checked before decoding. The general structure of the 112-bit ADS-B message is shown in Table 1.1. The total number of bits per block and their positions are written in the first column. Capability is the second field that contains three bits, which indicate the transponder level. These levels occur on the ground, in the air, and in the air or on the ground. It will be explained in detail in Chapter 2.

Table 1.1. Mode-S ADS-B message structure [7].

DF, 5 bits, 1^{st} to 5^{th} bit	Downlink Format
CA, 3 bits, 6^{th} to 8^{th} bit	Transponder Capability
ICAO, 24 bits, 9^{th} to 32^{nd} bit	ICAO, Aircraft Address
ME, 56 bits, 33^{rd} to 88^{th} bit	Message, extended squitter
PI, 24 bits, 89^{th} to 112^{th} bit	Parity/ Interrogator ID

The ICAO address is used in each ADS-B message to uniquely identify the sender (source aircraft) since each Mode-S transponder on an airplane has a unique ICAO address. In the binary message, these address bits are situated between bits 9 and 32. For instance, the address "4BAA0C" was found using aircraft data. There are roughly 16 million of these aircraft with ICAO addresses. The ICAO address of an aircraft should not alter over the course of its existence.

Table 1.2. Type codes and their corresponding contents [7].

Type Code	**Information Content**
1-4	Aircraft Identification
5-8	Surface Position
9-18	ICAO, Airborne Position
19	Airborne Velocities
20-22	Airborne Position
23-27	-
28	Aircraft Status
29	Target State
31	Operation Status

The type code forms the first five bits of the Message field and defines what information is included in the ADS-B message. The type code and related contents are shown in Table 1.2. Type codes ranging from 1 to 4 are used to express the aircraft identity or flight ID. Contrary to ICAO, this call sign is not specific to an aircraft, and several aircraft on the same route may use the same call sign from time to time.

1.3. Mode-S UF/DF and BDS Types

Mode-S Beacon System is an aircraft surveillance protocol that was first designed at MIT Lincoln Laboratory in the 1970s. The main characteristic of Mode-S is that it allows selective interrogation and allows SSR radars to separately ask for different information from each aircraft. Selective interrogation largely solves the garbling problem that occurs in Mode A/C, thus improving the channel capacity. In addition, Mode-S transponders have special 24-bit codes, which correspond to $2^{24} = 16777216$ different private addresses, so that each aircraft has its own address. In addition, many different message types can be used for communication via the Mode-S protocol. The Mode-S uplink signal contains the type of information the ground station wants to query. On both the uplink and downlink sides, there are two types of data blocks, 56-bit short messages, and 112-bit long messages.

With the Mode-S protocol, different message types can be sent in various formats. The first 5 bits of the messages always refer to this format type. These format types can be seen in Table 1.3.

Table 1.3. UF/DF types of Mode-S [7].

UF/DF	**Bits**	**Uplink Type**	**Downlink Type**
0	56	Short air to air surveillance (ACAS)	Short air to air surveillance (ACAS)
4	56	Altitude Request	Altitude Reply
5	56	Identity Request	Identity Reply
11	56	Mode-S All-Call	Mode-S All-Call reply
16	112	Long air to air surveillance (ACAS)	Long air to air surveillance (ACAS)
17	112	-	Extended Squitter
18	112	-	Extended Squitter/non-transponder
19	112	-	Military extended squitter
20	112	Altitude request via Comm-A	Altitude reply via Comm-B
21	112	Identity request via Comm-A	Identity reply via Comm-B
24	112	Comm-C (ELM)	Comm-D (ELM)

As will be seen in the next section, UF/DF 11 is used for all-call signals and UF/DF 4, 5, 20, and 21 are used for roll-call signals in Mode-S radar simulation. 4, 5, 20, and 21 are the most commonly used Mode-S formats. UF/DF 20 and 21 contain Comm-B messages other than the payload, and there are different types of messages in Comm-B selection. The options can be depicted in Figure 1.4.

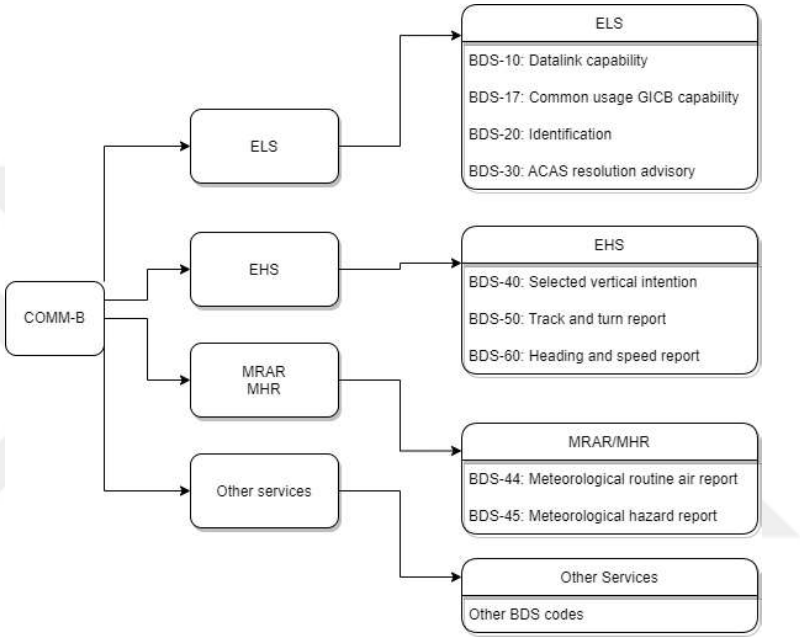


Figure 1.4. Comm-B message types [7].

Comm-B messages also allow the ground station to gather specialized information other than position. What data is contained in message fields is decided by the Comm-B Data Selector (BDS), an 8-bit code. The common BDS formats are BDS-10, 17, and 20, as they correspond to elementary surveillance. The availability and number of received Comm-B messages depend heavily on the density of air traffic and the rate of interrogations [28]. Since the BDS selection is specified in the first 8 bits of the message content of the long signal, 255 information types can be sent in theory, but not all of them are used in practice. Therefore, the most commonly used BDS formats are summarized in Figure 1.4. While ELS is the elementary surveillance service; EHS is an enhanced surveillance service. MRAR/MHR services are used to transmit meteorological reports. While these services are available for UF/DF 20 and 21, there is no long message information in UF/DF 4 and 5 formats.

2. MODELLING THE SIMULATION SYSTEM

Mode-S SSR system model and simulation were performed in a high-level MATLAB environment. While creating the graphical user interface in the simulation, MATLAB GUI Appdesigner was used. The Mode-S air traffic radar system model consists of two main parts. The first of these is the interrogator unit. The interrogator is the unit at the airport that sends interrogation signals to aircraft. The other part is the transponder unit. The transponder unit is the unit found in airplane and similar aircraft in flight. These two units, seen in Figure 2.1, ensure safe air traffic communication through two RF channels.

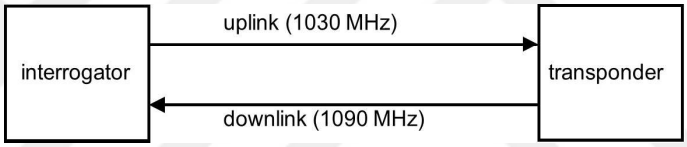


Figure 2.1. Communication scheme between interrogator and transponder.

The interrogator unit sends SSR interrogation signals through 1030 MHz frequency channel via a rotating antenna and the unit receives reply signals from aircraft inside antenna coverage through 1090 MHz frequency downlink channel. SSR interrogation antennas form an antenna array and their collective beam-width ranges from 2.4 to 12.5 degrees angle. The default beam-width is selected as 2.4 degrees in this simulation, however, the simulator is designed so that the beam-width can be changed by the user. Additionally, a common antenna radiating power is selected as 1500 W by default. This RF power is used for interrogations of aircraft at the maximum range. If the aircraft to be interrogated is close to the radar, the RF power is reduced according to the distance of the aircraft to the radar in order not to deteriorate the signal quality. RF power level management strategies are not a topic in this thesis, therefore the power level is taken maximum and constant.

One of the most important parameters determining the interrogation is the rotation speed of the interrogation antenna. In the literature, antenna rotation speed

varies between 6 rpm and 15 rpm [27, 29]. In this study, it was determined as 15 rpm. This value can be changed when the properties of the antenna are finalized by the user for simulation. Since the interrogator radar rotates on the azimuth axis, the time taken for the 3 dB beam-width is calculated as

$$\mathbf{t}_h = \frac{\varphi_{3dB}}{6n_{ant}}, \tag{2.1}$$

which is derived and used within the framework of this simulation. Here, the antenna rotation speed is specified in n_{ant} rpm (revolutions per minute), the beam radiation time is t_h seconds, and the antenna 3 dB beam-width φ_{3dB} is in degrees. During the radar operation, the space it scans in the azimuth axis while rotating is divided into M slices as shown in Figure 2.2, and each slice has t_h processing time. During this time, an interrogation should be performed in the beam slice where the antenna radiation is made. When default parameters, discussed above, are put into Equation (2.1), the time elapsed through a 3-dB antenna beam-width is obtained as 26.67 ms. This duration is much longer than air traffic radar's coverage. This means that more than one interrogation can be performed within a beam-width time. Time planning is of great importance while designing an interrogation strategy.

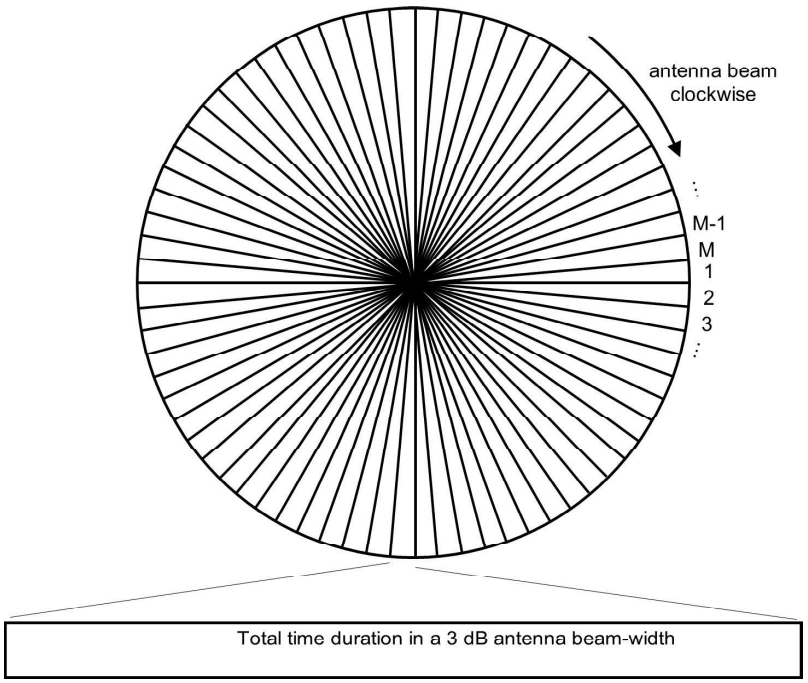


Figure 2.2. Azimuth space divided by antenna into M slices.

In order to develop the SSR interrogation algorithm, the processes that need to be done in SSR systems, the time required for them, and the relationships between these processes should be examined. One of these is the 3-dB beam-width duration due to the rotation of the SSR antenna, which has a narrow beam angle. Since this time is much longer than the time required for an SSR interrogation, it can also be divided into interrogation periods as shown in Figure 2.3. Thus, N interrogations can be performed within a 3-dB beam-width duration.

In modern SSRs, the old system ATCRBS and the new system Mode-S must work simultaneously. Accordingly, the Mode-A/C and Mode-S interrogation must be in the same system. There are 2 types of interrogations in Mode-S systems. The first of these is all-call interrogation and the other is roll-call or selective-call interrogation. As shown in Figure 2.3, a large number of all-call and roll-coll interrogations can be performed in a beam-width time. All-call interrogations should constitute at least approximately 1/3 of the time in the allotted time period so that there is no section or sector without an all-call interrogation in the entire airspace.

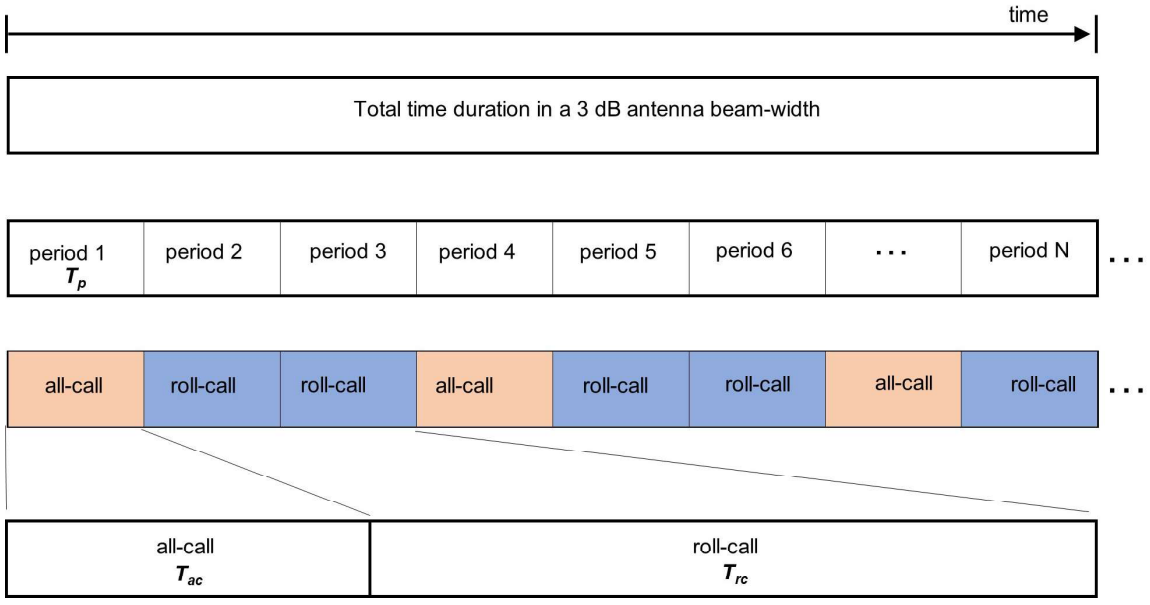


Figure 2.3. Dividing time into periods in an antenna 3-dB beam-width duration.

Another important factor is the pulse repetition frequency of Mode-S signals, which ranges between 150 and 450 Hz. This also affects the maximum coverage range and can be adjusted via user interface as well.

Figure 2.4 is an illustration that visually explains how the interrogation is performed. First, the interrogator sends an all-call (UF 11) interrogation package of τ_{sp} length on the 1030 MHz frequency channel. This signal reaches the transponder by traveling at the speed of light in the air. When the transponder receives this package, it responds with an all-call downlink message lasting τ_{cp} through the 1090 MHz channel after the τ_a signal processing time has passed. This all-call reply package contains the ICAO address of the transponder. After the interrogator learns the ICAO address of the transponder, it sends the selective-call/roll-call interrogations directly to the transponder by specifying the ICAO address. Roll-call interrogations are interrogations that ask for the desired information about the flight of the aircraft. The transponder, which receives the roll-call interrogations coming to its own ICAO address, sends the reply packages to the interrogator on the 1090 MHz frequency channel. Most commonly used roll-calls are done by UF 4/5/20/21 formats.

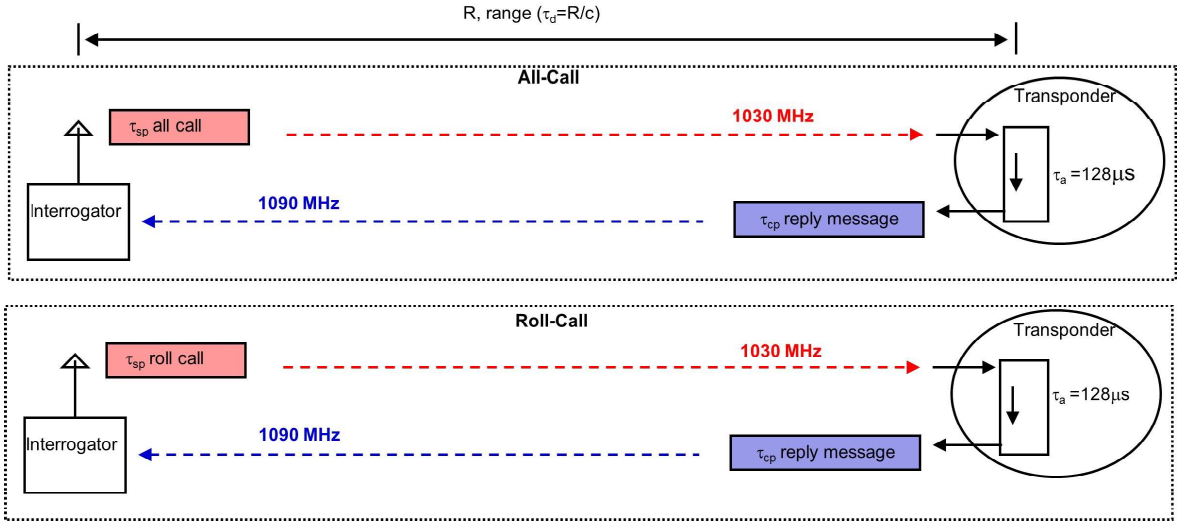


Figure 2.4. All-call and roll-call interrogation illustration.

2.1. Aircraft Scenarios in a Beam-width

2.1.1. One Aircraft in a Beam

If there is one transponder in the direction of the interrogated beam at distance R_1, first, the ICAO address of the transponder is gathered with an all-call interrogation.

Then, with roll-call interrogations, the flight information of the aircraft is learned. This process takes place according to the time-flow chart in Figure 2.5.

R_1 is the distance between an interrogator and a transponder, therefore the time spent in the air which is τ_d in one direction can be expressed as

$$\tau_d = \frac{R_1}{c}. \tag{2.2}$$

τ_{sp} and τ_{cp} stand for message durations, τ_a and τ_s stand for processing time during signal transmission and reception. The total time spent for one all-call can be calculated as

$$\tau_{total} = 2\tau_d + \tau_{sp} + \tau_a + \tau_{cp} + \tau_s. \tag{2.3}$$

At the end of this duration and after obtaining the ICAO of the transponder, the roll-call interrogations are in process. Since interrogations and replies are sent on different channels, if the interrogator knows the ICAO address of the transponder to be interrogated or will send consecutive interrogations that do not require ICAO information, τ_s time can be ignored.

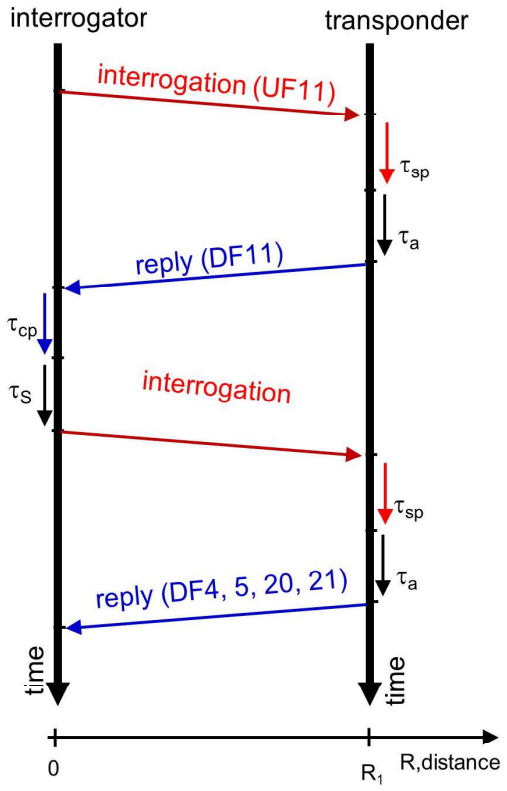


Figure 2.5. Time-flow chart for a single transponder case.

2.1.2. Multiple Close Aircraft in The Same Beam

If there are many closely spaced aircraft in a beam, all-call signals are captured by the closest aircraft, followed by the second closest aircraft, and so on. All aircraft reply in the same 1090 MHz downlink channel; hence, the downlink signals can overlap in the channel according to their closeness. If two aircraft are close to each other, the reply signal belonging to the second aircraft arrives before the first reply signal sent from the same channel is completed and these two packets collide. This causes the interrogator to not obtain an ICAO address, and roll-calls cannot be processed. In a dense air traffic situation, the Mode-S system cannot be as effective as desired. A close scenario example is shown in Figure 2.6.

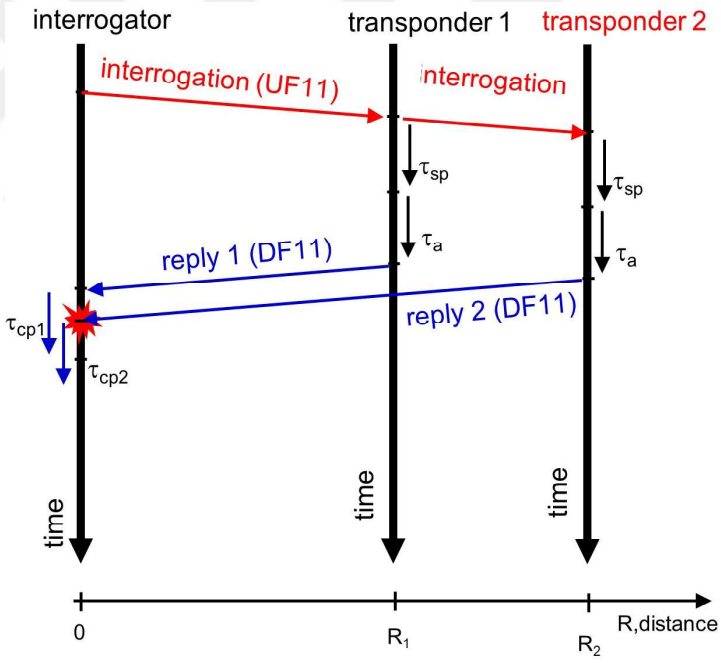


Figure 2.6. Time-flow chart for two transponders flying close to each other in succession.

2.1.3. Multiple Distant Aircraft in The Same Beam

The distance between aircraft is larger than ΔR_{min} in Figure 2.7. In this case, collusion does not happen, and the interrogator can receive all the replies.

Mode-S all-call downlink reply (DF 11) is a 56-bit answer signal on the 1090 MHz channel. This package corresponds to the short package in the Mode-S system and the all-call reply package time is 64 (56+8) μs.

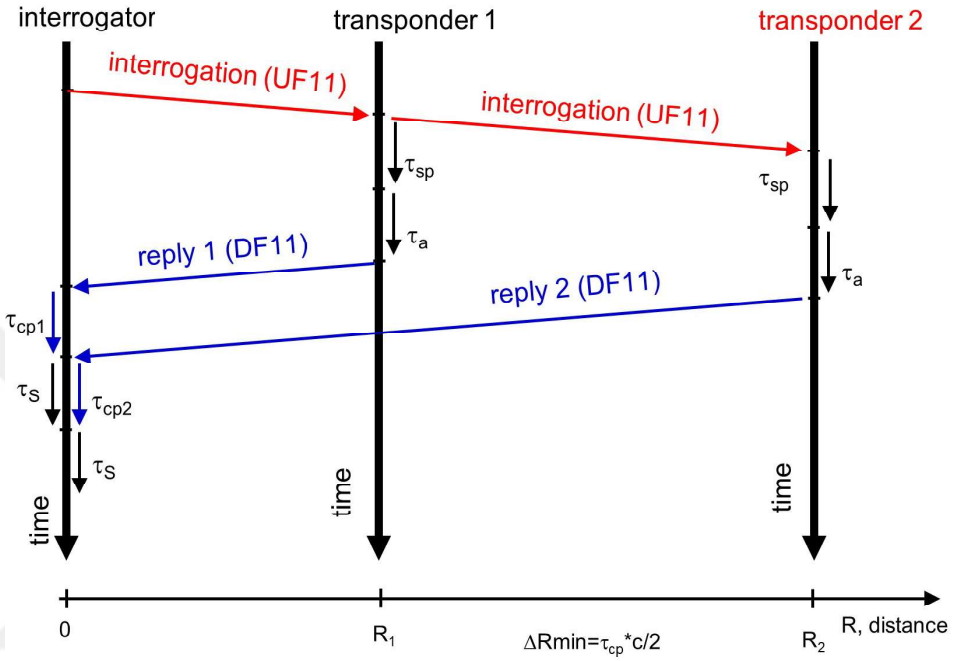


Figure 2.7. Time-flow chart for two transponders flying distant to each other in succession.

Using the all-call reply package time, ΔR_{min} is calculated as 9.6 Km (5.2NM). The 24-bit CRC parity at the end of the DF 11 package, which is used to check the correctness of the data package, does not carry information. If we subtract that time, ΔR_{min} becomes 6 Km (3.2 NM). In this simulation study, while the scenario was created, the distance between the planes was determined to be more than 10 km. In the next sections, probabilistic interrogation method and data error correction studies will be carried out for situations such as two planes being closer to each other than ΔR_{min}.

2.1.4. A Sample Scenario Simulation

A sample scenario is created to test the simulation algorithm and to see the results. The illustration of the scenario is shown in Figure 2.8.

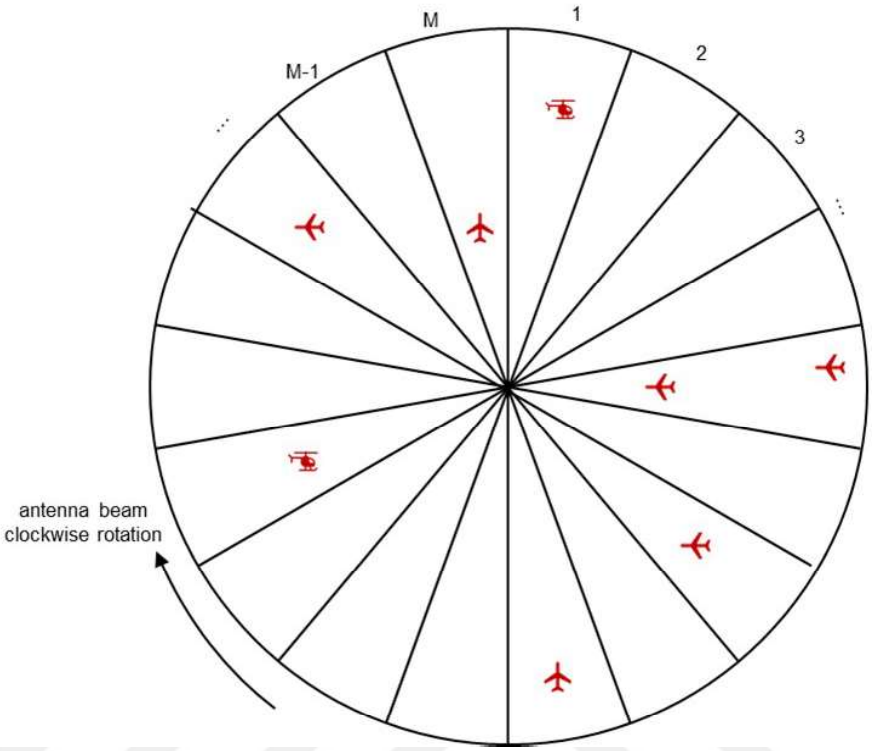


Figure 2.8. Illustration of a sample scenario.

A polar coordinate axis is used to represent the positions of aircraft, where there is a transponder at the center. The locations of aircraft are given by the user as inputs and the required interrogations are performed by the simulator. For the simulation, the user must enter the positions of the aircraft relative to the interrogator, the ICAO address, SQUAWK information, and the altitude of the scenario. Each line is for one transponder. The sample scenario in Table 2.1 contains information of 17 aircraft in the airspace determined by the user.

The simulation of the Mode-S system is performed as follows. First, the simulation time is determined. Then, antenna rotation speed (rpm), beam-width, and PRF values are entered. Then, sectors in the airspace are calculated in the polar coordinate space. Then, Mode-S interrogation processes are started.

Table 2.1. Preset aircraft parameters.

ANGLE(Degree)	RANGE(M)	ICAO 24-BIT ADDRESS	SQUAWK	CALIBRATED ALTITUDE (FT)
10	50000	4BA9E4	5617	8725
45	100000	4BA9C9	5156	3850
135	300000	4BAA0B	2366	16875
215	200000	4B8E0D	5114	14000
335	400000	4B8E1F	2204	15000
0	300000	4BA9A0	5617	8725
30	400000	4BA9A1	5156	3850
60	350000	4BA9A2	2366	16875
90	300000	4BA9A3	5114	14000
120	450000	4BA9E4	2204	15000
150	300000	4BA9A5	5617	8725
180	250000	4BA9A6	5156	3850
210	320000	4BA9A7	2366	16875
240	260000	4BA9A8	5114	14000
270	300000	4BA9E9	2204	15000
300	320000	4BA9AA	2366	16875
330	380000	4BA9AB	5114	14000

While the antenna is rotating in the time flow, the interrogation is performed as UF 11, UF 4, UF 5, UF 11, UF 20, and UF 21, respectively, hence the 1/3 all-call to beam-width duration ratio is kept constant. When this interrogation pattern is completed, the interrogation is repeated over and over again in the same interrogation pattern order until the simulation period ends. The hexadecimal DF 11, DF 4, DF 5, DF 20, and DF 21 Mode-S reply signals from these interrogations are recorded by the interrogator and the necessary information is obtained by dividing them into parts. According to this information, the location of the transmitters, ICAO address, identity (identification), altitude, and the information sent by the aircraft are received.

2.2. Interrogation Sequences

The first interrogation sent is UF 11. UF 11 is a Mode-S all-call interrogation. All aircraft that receive the Mode-S interrogation and are not locked up previously must reply to it. The lockout concept will be studied in Chapter 3. If there is an aircraft in the beam of the radar, this aircraft receives an interrogation signal with a

delay due to the constant propagation speed of electromagnetic signals according to its distance from the radar, produces the DF 11 signal in hexadecimal and sends it as a reply to the interrogator. The interrogator decodes the DF 11 signal from the reply it receives with a delay, determines the ICAO address of the aircraft in the air and the position of the aircraft using the delay, and displays it on the simulation monitor. After learning the ICAO address, the interrogator can send a roll-call interrogation to the aircraft with this ICAO address.

During the simulation, after obtaining ICAO, the roll-call process starts with UF 4/5, and then the pattern follows UF 11/20/21 in order. UF 4/5 interrogates with 56 bit format for altitude and squawk information, respectively; whereas UF 20/21 interrogates with 112 bit format for altitude plus BDS message or squawk plus BDS message, respectively. DF 4/5 are 56-bit reply messages corresponding to UF 4/5, whereas DF 20/21 are 112-bit reply messages corresponding to UF 20/21. In this study, hexadecimal forms of all these formats are produced, and the CRC parity generator algorithms required for these data packets in a real Mode-S system are also designed.

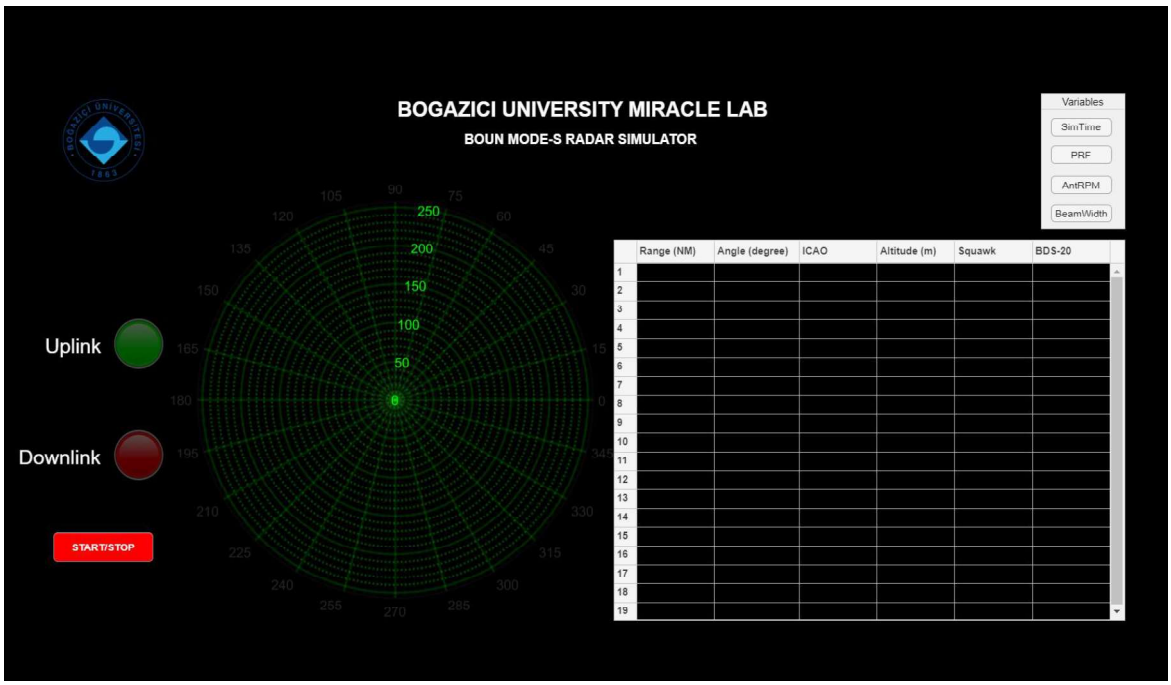


Figure 2.9. Radar simulator GUI.

In this simulation, it is aimed to perform some Mode-S interrogations, namely UF 4, 5, 20, and 21, for each aircraft within the radar's coverage area and receive the replies, and then extract the messages to display on the screen as in Figure 2.9. If the time is not enough to perform all the intended UF 4, 5, 20, and 21 interrogations during a beam-width duration, the interrogator notes which interrogation it made last and when the radar beam arrives at the location of that aircraft in the next tour, the interrogator continues from where it left off and completes all the interrogations.

In order to follow this flow and make the interrogations in order, a code scheme has been created. In general, it is ensured that the aircraft is detected with the help of UF signals. Then the necessary information of the aircraft whose ICAO was obtained by all-call is gathered by roll-call interrogations. The data contents are accessed by obtaining messages from the information embedded in the DF. In order to obtain this data, it is known in advance what information is contained in which bits of the entire message. Accordingly, the message is read. The most common Mode-S formats are used such as UF/DF 4, 5, 11, 20, and 21.

2.2.1. UF 11

As can be seen in Table 2.2, the Mode-S only all-call interrogation (UF 11) structure has six information sections. Their contents are as follows. UF is the uplink format number in binary. For example, for 11 it is [0 1 0 1 1]. PR is the probability of the aircraft to reply the interrogation. Default is selected as 8 without lockout, [1 0 0 0]. Each PR code and its corresponding meaning are given in Table 2.3.

Table 2.2. 56-bit UF 11 short message structure.

UF (5 Bits)	**PR (4 Bits)**	**IC (4 Bits)**	**CL (3 BIT)**	**Spare (16 Bits)**	**AP (24 Bits)**

IC (Interrogator Code) contains the interrogator ID (IC) or surveillance identifier (SI). Default is selected as [0 0 0 1]. CL is the code label. If CL equals 0, then the former 4 bits mean IC. When CL is between 1-4, then the former 4 bits are constructed

according to the SI table [7]. Default is selected as CL being 0, hence former 4 bits are defined as IC bits. Spare is spared bits. These bits can be empty; hence, the default selection is 0 for spare. AP is address parity. 24 bits are designed for error detection/correction purposes. For UF 11, it is always $[111...1]_{24}$ [4]. A sample hexadecimal UF 11 is generated under default conditions: '5C080000FFFFFF'.

Table 2.3. Probability of reply with corresponding codes [26].

PR Code	**Meaning**
0	Reply with probability 1
1	Reply with probability 1/2
2	Reply with probability 1/4
3	Reply with probability 1/8
4	Reply with probability 1/16
8	Ignore Lockout, reply with probability 1
9	Ignore Lockout, reply with probability 1/2
10	Ignore Lockout, reply with probability 1/4
11	Ignore Lockout, reply with probability 1/8
12	Ignore Lockout, reply with probability 1/16
5,6,7,13,14,15	Do not Reply

2.2.2. DF 11

A Mode-S only all-call reply message (DF 11) consists of 4 main parts. DF is the binary downlink format. For example, it is [0 1 0 1 1] for 11. CA is the capability. It corresponds to the transponder's communication capability, which is between 0 and 7. It varies depending on level (1-2) and whether the plane is on the ground or in the air. Default is selected as 5, [1 0 1]. Each CA code with its meaning is shown in Table 2.5.

Table 2.4. 56-bit DF 11 short message structure.

DF (5 Bits)	**CA (3 Bits)**	**ICAO (24 Bits)**	**Parity (24 Bits)**

Table 2.5. CA information [26].

CA	Meaning
0	Level 1 Transponder
1-3	Reserved
4	Level 2+ Transponder, with the ability to set CA to 7, on-ground
5	Level 2+ Transponder, with the ability to set CA to 7, airborne
6	Level 2+ Transponder, with the ability to set CA to 7, either on-ground or airborne
7	Specifies that the downlink request value is 0. or the Flight Status 2,3,4 or 5, either on-ground or airborne

A CRC remainder, which will be explained in Section 2.4.2, and interrogator identifier are overlaid with each other, and the combination is called parity bits. The interrogator identifier code can be zero and the message is generated in this case as a result of a spontaneous acquisition squitter [7]. This acquisition squitter message, in which the parity is only the CRC remainder, is called ADS-B short message in Section 2.4.2. A sample hexadecimal DF 11 generated under default conditions is '5D4BA9E42A8800'.

2.2.3. UF 4

The 56-bit short message interrogation (UF 4) consists of six main parts. UF (uplink format) is [0 0 1 0 0] for 4. PC is the protocol. If it is 1 then a non-selective all-call lockout is applied. Default is selected as 1 since UF 5 is interrogated after UF 4; however, this order of interrogations can be changed.

Table 2.6. 56-bit UF 4 short message structure.

UF (5 Bits)	PC (3 Bits)	RR (5 Bits)	DI (3 Bits)	SD (IC(4 Bits)+12 Bits)	AP (24 Bits)

RR is the reply request. This tells DF to send short or long messages. The default is selected as 0 which corresponds to a short reply length. DI is the designator identification. DI determines the SD field. The default is 1. SD is the special designator. It contains codes special to the SSR. Default is selected as [IC zeros(1,12)]. AP is the parity bits formed by overlaying ICAO and CRC remainder. A sample hexadecimal UF 4 generated under default conditions is '210110000D3EA9'.

2.2.4. DF 4

DF 4 (56-bit short message reply) consists of six main parts. DF is [0 0 1 0 0] for 4. FS is the flight status. It contains information about the aircraft's status, and also alarm and emergency situations. The default is selected as 0. DR (downlink request) is the datalink warning bits. It contains the type of prepared bits to send, e.g., DR Code 4 means 'Comm-B broadcast message 1 available' [26]. UM (Utility Message) is datalink interrogator link addressing. It contains information about the connected interrogator. The default is taken as [IC 0 0] for the example message.

Table 2.7. 56-bit DF 4 short message structure.

DF (5 Bits)	FS (3 Bits)	DR (5 Bits)	UM (IC(4 Bits)+2 Bits)	Altitude (13 Bits)	AP (24 Bits)

The altitude section contains altitude information in feet or meters. In this simulation, feet is selected as the unit. The altitude can be coded with 25 or 100 feet increments. 25 feet increments are selected in this work. The decimal conversion of real altitude is shown as

$$Altitude_{converted} = \frac{Altitude_{real} + 1000}{25} \tag{2.4}$$

This decimal number calculated from the decimal conversion is then converted to an 11-bit binary. To specify that the calculation done is in 25 feet increments, 0 is put at the 7^{th} bit location and 1 is put at the 9^{th} bit location. In the end, the 13-bit $(11 + 2)$ long altitude info is encoded. AP is again the parity part. A sample hexadecimal DF 4 generated under default conditions is '20008615E9DE78'.

2.2.5. UF 5

As can be seen in Table 2.8, UF 5 is almost the same as UF 4 except for the binary uplink format, which is [0 0 1 0 1] for UF 5. Therefore, there is no need to repeat the same definitions as in UF 4.

Table 2.8. 56-bit UF 5 short message structure.

UF (5 Bits)	PC (3 Bits)	RR (5 Bits)	DI (3 Bits)	SD (IC(4 Bits)+12 Bits)	AP (24 Bits)

As expected, the parity bits should also be different from UF 4's for the same sample aircraft. The reason is that the CRC remainder overlaid with ICAO is different, hence overall hexadecimal is not identical. A sample hexadecimal UF 5 generated under default conditions is '29011000AD2038'.

2.2.6. DF 5

The DF 5 message format is the same as DF 4 except for the first 5 bits and bits between positions 20 and 32 that contain the squawk. The downlink format bits are [0 0 1 0 1]. While FS, DR, and UM are the same as in DF 4, AP bits differ because the payload content is different.

Table 2.9. 56-bit DF 5 short message structure.

DF (5 Bits)	FS (3 Bits)	DR (5 Bits)	UM (IC(4 Bits)+2 Bits)	Squawk (13 Bits)	AP (24 Bits)

Squawk codes are 4-digit codes that provide effective communication between the ground station and aircraft. For example, 7500, 7600, and 7700 squawk codes are emergency codes. Squawk codes are between 0000 and 7777, and every decimal is encoded with 3 bits, making a total of 12 bits. This bit string, which is converted from decimal to binary, is sent scrambled. If we define the 4-digit code from the left side, from A to D, each digit is divided into 3 sub-bits as

$$A4\ A2\ A1 \mid B4\ B2\ B1 \mid C4\ C2\ C1 \mid D4\ D2\ D1 \tag{2.5}$$

Then this sequence is mixed, and binary 1 is added to the 7^{th} bit, thus obtaining a 13-bit code sequence. The resulting formation is shown as

$$C1\ A1\ C2\ A2\ C4\ A4\ 1\ B1\ D1\ B2\ D2\ B4\ D4 \tag{2.6}$$

An example hexadecimal DF 5 code generated under these conditions is '280098DFF9B 03E'.

2.2.7. UF 20

UF 20 can be thought of as a long version of UF 4. The content of these sections has already been mentioned in the UF 4 part, the default selections are shown in Table 2.11.

Table 2.10. 112-bit UF 20 long message structure.

UF (5 Bits)	PC (3 Bits)	RR (5 Bits)	DI (3 Bits)	SD (IC(4 Bits) +12 Bits)	MA (8 Bits BDS + 48 Bits)	AP (24 Bits)

Table 2.11. Default selections for UF 20 long message example.

Message Part	Binary Code	Meaning
UF	[1 0 1 0 0]	Uplink Format 20 in Binary
PC	[1 0 0]	Comm-B lockout protocol
RR	[1 0 0 0 0]	Comm-B long message token
DI	[1 1 1]	7 selected for BDS format
SD	[0 0 0 1 zeros(12)]	The same as UF 4

MA corresponds to Message, Comm-A. The first 8 bits define the BDS type. Default BDS type is selected as 20 in the simulation, [0 0 1 0 0 0 0 0].

BDS-20 contains aircraft identification information, also known as a call sign. Unlike ICAO, the call sign is not unique to aircraft, different aircraft flying through the same route can have the same call sign at different times. Call signs are at most 8 characters and 48 bits long, with every character having 6 bits. Call signs may contain characters such as the country of origin, manufacturer information, and abbreviations for the pilot. In sample simulations, call signs are selected as 'TURKISH' for illustration purposes. When we read 8 bits formed by 4-bit digits, we see that it is encoded as 2,0. Address parities are also obtained by overlaying ICAO to the top of the payload message. A sample hexadecimal UF 20 generated under default conditions is 'A4871000200000000000000274D7'.

2.2.8. DF 20

The DF 20 format can be thought of as a long message version of DF 4. The selections of DF 20 for these sections, whose contents have already been mentioned, are shown in Table 2.13.

Table 2.12. 112-bit DF 20 long message structure.

DF (5 Bits)	FS (3 Bits)	DR (5 Bits)	UM (IC(4 Bits) +2 Bits)	Altitude (13 Bits)	MB (BDS(8 Bits) + Message(48 Bits)	AP (24 Bits)

Table 2.13. Default selections for DF 20 long message example.

Message Part	Binary Code	Meaning
DF	[1 0 1 0 0]	Downlink Format 20 in Binary
FS	[0 0 0]	No alarm and airborne
DR	[0 0 0 0 1]	Comm-B message indicator
UM	[0 0 0 1 0 0]	IIDS is taken as zero

Altitude is encoded the same as in DF 4. MB corresponds to Message, Comm-B. The first 8 bits indicate BDS type, which is 20 in this simulation as stated in UF 20. The rest of the 56 bits are generated by a character look-up table to construct the call sign. The character look-up table for characters is

$$' ABCDEFGHIJKLMNOPQRSTUVWXYZ \qquad 0123456789 \quad ' \tag{2.7}$$

The blank spaces are placed intentionally to create the character look-up table.

The index of every character is read, and the corresponding binary conversion is done in 6 bits. For example, for the letter A, the mapped binary sequence is [0 0 0 0 1 0], since A is located at index two. All letters are encoded this way and embedded in the MB bit order. A sample hexadecimal DF 20 generated under default conditions is ‘A0088615205564CC2942618200FE’.

2.2.9. UF 21

As one can see from Table 2.14 and Table 2.10, UF 21 is almost the same as UF 20, except that the uplink format is equal to [1 0 1 0 1]. In order to avoid repetition, the same content has not been explained in this format.

Table 2.14. 112-bit UF 21 long message structure.

UF (5 Bits)	PC (3 Bits)	RR (5 Bits)	DI (3 Bits)	SD (IC(4 Bits) +12 Bits)	MA (8 Bits BDS + 48 Bits)	AP (24 Bits)

An example hexadecimal UF 21 code generated under these conditions is 'AC8710 00200000000000000C1E30C'. As expected, UF bits and AP parity bits differ from the ones in UF 20. The reason why the AP bits are different is that the payload overlaid with ICAO has changed again.

2.2.10. DF 21

DF 21 can be considered a long message version of DF 5. Here, DF is equal to [1 0 1 0 1]. FS, DR, UM, and MB contents are the same as DF 20. The squawk code is the same as used in DF 5. The address parity will be different as it is obtained by overlaying ICAO and payload. An example hexadecimal DF 21 code generated under these conditions is 'A80898DF205564CC29426109D958'.

Table 2.15. 112-bit DF 21 long message structure.

DF (5 Bits)	FS (3 Bits)	DR (5 Bits)	UM (IC(4 Bits) +2 Bits)	Squawk (13 Bits)	MB (BDS(8 Bits) + Message(48 Bits)	AP (24 Bits)

2.2.11. Overall Summary of Interrogation Sequences

Example hexadecimal sequences are created in Sections 2.2.1 to 2.2.10. These hexadecimal sequences are shown in Table 2.16 for the same aircraft.

Table 2.16. Produced formats of the sample aircraft.

Format	**Hexadecimal Sequence**
UF11	'5C080000FFFFFF'
DF11	'5D4BA9E42A8800'
UF4	'210110000D3EA9'
DF4	'20008615E9DE78'
UF5	'29011000AD2038'
DF5	'280098DFF9B03E'
UF20	'A4871000200000000000000274D7'
DF20	'A0088615205564CC2942618200FE'
UF21	'AC8710002000000000000C1E30C'
DF21	'A80898DF205564CC29426109D958'

The interrogation flow of simulation, namely creating the interrogation and reply signals, was mentioned in this section. In the other part of the simulation, message acquisition, the coded hexadecimal sequences are converted to their decoded form. The reverse is applied when generating Altitude, Squawk, and BDS-20 messages. These extracted messages are printed directly to the table in the application designer. In Table 2.17, we see the sample message acquisitions printed for the aircraft above.

Table 2.17. Sample message acquisitions of the sample aircraft.

Range	**Angle**	**ICAO**	**Altitude**	**Squawk**	**BDS-20**
26.9978	9.0000	4BA9E4	8725	5617	TURKISH

2.3. Results of Radar Simulation

One of every three interrogation times in this simulation is reserved for all-call interrogations. If there is no aircraft where the radar beam is pointing, there will be no reply after the all-call signal has been sent. In sectors where there is no all-call reply, roll-call interrogations such as UF 4, 5, 20, and 21 are not performed. In this study, if there are no aircraft in the place where the radar is looking, the radar does not interrogate in the 2-time units after the all-call interrogation. In the next fourth

time frame, it performs an all-call interrogation again. It does this until it receives an all-call reply. Thus, the Mode-S radar does not unnecessarily occupy the 1030 MHz uplink channel in the region and time where there are no aircraft.

The screen of the Mode-S radar system simulator, which was created to test the developed algorithms, is shown in Figure 2.10. While the Mode-S radar system is operating, two signal lamps are placed on the radar interface screen to track the times when the uplink signals are sent, and the downlink signals are received. These signal lamps are located on the left of the screen. The green light indicates that the uplink signal is sent, and the red light indicates that the downlink signal is received. Since the 1/3 time distribution rule is used in all-call interrogations, the green light always appears to be active over the entire airspace while the radar is operating. In the region with no aircraft, there is no questioning in the 2/3 time interval. However, because the interrogation pattern is very fast, the human eye perceives the green light as always on. The red light turns on when a downlink signal is coming. Thus, the designed interrogation pattern and the corresponding reply signaling can be followed.

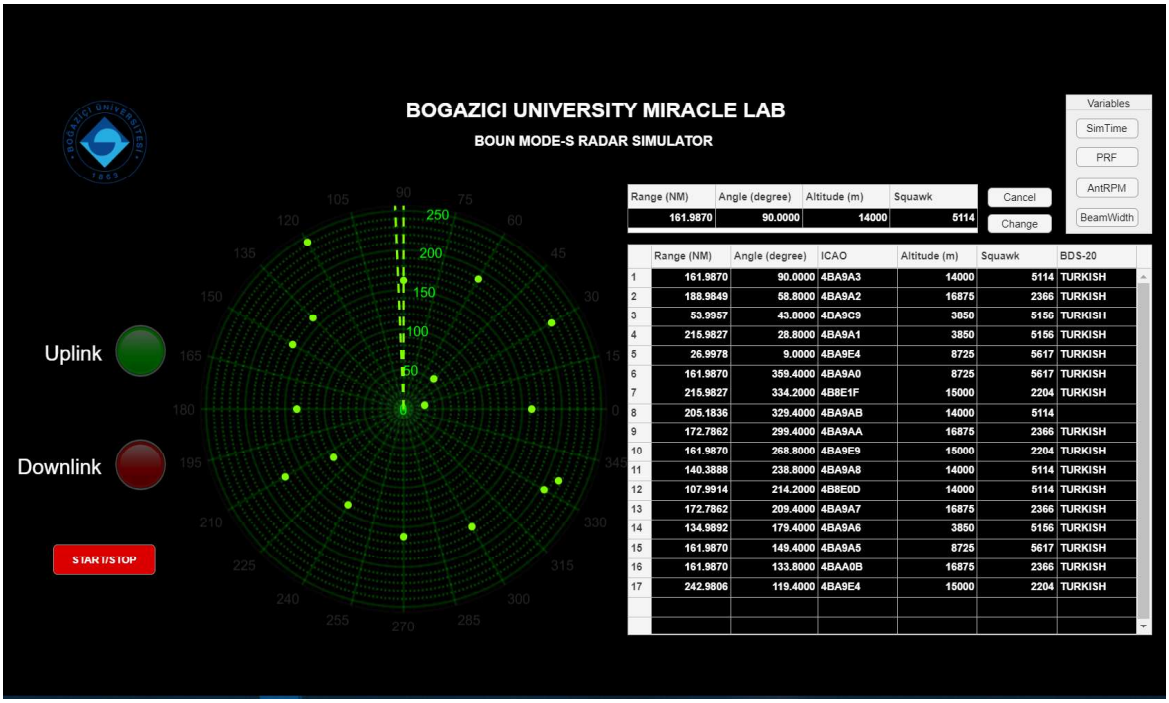


Figure 2.10. Radar simulator GUI with outputs.

In the middle of the screen, the radar airspace appears in the polar coordinate system. There is an interrogator in the center. The rotating radar beam is indicated. As the radar rotates clockwise, it shows the locations of the planes it detects with Mode-S interrogations on this polar coordinate plane. The nautical mile is used as a distance unit. The maximum distance is 256 miles. On the right side of the screen, there is a summary of the information about the aircraft that the radar detects as it works. Each separate line represents a different detected aircraft. Each column contains information about the distance of the aircraft, its angle, ICAO, altitude, Squawk information, and BDS-20 message, respectively. As can be seen in detail in Figure 2.11, the output results of the table on the radar screen match the information entered by the user shown in Table 2.1 on a one-to-one basis. While ICAO, Altitude, and Squawk values are precisely correct, Range and Angle values are approximations at the decimal level.

	detectedaircrafts					
	RANGE	Angle	ICAO	Altitude	Squawk	BDS_20
	Number	Number	Text	Number	Number	Categorical
1	161.9870	90.0000	4BA9A3	14000	5114	TURKISH
2	188.9849	58.8000	4BA9A2	16875	2366	TURKISH
3	53.9957	43.8000	4BA9C9	3850	5156	TURKISH
4	215.9827	28.8000	4BA9A1	3850	5156	TURKISH
5	26.9978	9.0000	4BA9E4	8725	5617	TURKISH
6	161.9870	359.4000	4BA9A0	8725	5617	TURKISH
7	215.9827	334.2000	4B8E1F	15000	2204	TURKISH
8	205.1836	329.4000	4BA9AB	14000	5114	TURKISH
9	172.7862	299.4000	4BA9AA	16875	2366	
10	161.9870	268.8000	4BA9E9	15000	2204	TURKISH
11	140.3888	238.8000	4BA9A8	14000	5114	TURKISH
12	107.9914	214.2000	4B8E0D	14000	5114	TURKISH
13	172.7862	209.4000	4BA9A7	16875	2366	TURKISH
14	134.9892	179.4000	4BA9A6	3850	5156	TURKISH
15	161.9870	149.4000	4BA9A5	8725	5617	TURKISH
16	161.9870	133.8000	4BAA0B	16875	2366	TURKISH
17	242.9806	119.4000	4BA9E4	15000	2204	TURKISH

Figure 2.11. Radar simulator output variables.

In addition, a structure array has been created for all transmitted pulses, including antenna beam number, angle, signal sending and receiving time, and UF/DF types. An example of this array can be found in Figure 2.12.

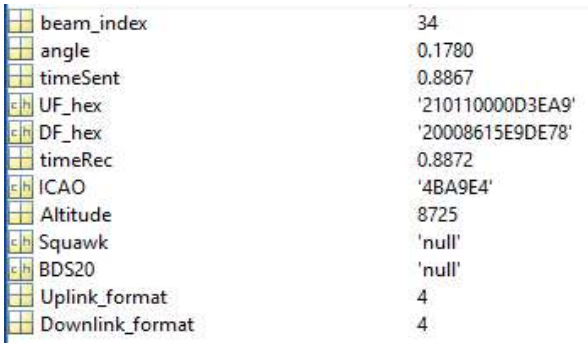


Figure 2.12. An example of output structure array.

As a result of all these outputs, it is seen that the Mode-S radar simulator works successfully. Also, the results show that the simulator performs UF/DF 4, 5, 11, 20, and 21 formats properly.

2.4. Real-Time System Model

The real-time system model is examined under two main headings: hardware and software model. The hardware model used in this thesis is built to collect short (56-bit) and long (112-bit) messages in the 1090-MHz band from aircraft transponders. Detailed information on hardware installation will be given in subsection 2.4.1. Briefly, the first part of the receiver system is the antenna that can pick up 1090 MHz signals. Then, a band-pass filter centered at 1090 MHz is placed to attenuate the unwanted signals. The filter is followed by a low noise amplifier (LNA) which is biased by an external DC power supply. After the LNA, a DC-Block is utilized to protect the receiver from unpredictable power levels. The DC block ensures that only the RF signal is allowed to pass through, preventing any DC voltage from appearing at the output. Finally, a radio equipment that will cover the transmitter/receiver bandwidth levels of 1090-MHz, and a personal computer system connected to this equipment with a USB cable complete the hardware installation.

MATLAB was chosen as the environment for software modeling. Both the data acquisition and the CRC process were implemented in this environment. The steps

of the CRC process are CRC generation from the collected data, CRC control, and bit/hex decoding of the data in the signals which have passed these tests. If we summarize the system stages, firstly, the 1090-MHz aircraft reply signals collected from the aerial environment via the antenna are recorded in a text file using the hardware parts mentioned above in hexadecimal form for ease of use. The data recording duration is determined by the user in seconds. Then, the recorded text file that contains the hexadecimal aircraft data is inserted into the CRC generation and control stages. After the statistical separation of erroneous and error-free messages, the erroneous messages are forwarded to the error correction stage. The error-free ones are decoded according to the previously determined message structure. After summarizing the simulation environment, it is necessary to mention the CRC codes at this stage.

Mode-S interrogation signals sent by the ground station in the 1030-MHz band and the reply signals sent by the aircraft in the 1090-MHz band contain Cyclic Redundancy Check (CRC) codes. Remaining part of the thesis, the terms parity code and CRC code are used interchangeably to avoid any confusion of terms, however, the address parity (AP) bits mentioned during interrogation sequences are different. For ADS-B messages, parity bits are only CRC remainders because ADS-B is not interrogation-based, the identifier is set to zero. When we consider the downlink side, these parity codes constitute the last 24 bits of the short and long reply signals. Reply signals of 56 or 112 bits can be thought of as *32+24* and *88+24* with parity bits added, respectively. CRC is used to detect data loss when interrogation or reply signals arrive with an error caused by the air environment, channel, or hardware problems. It even helps to fix some corrupted bits, but more detailed information on error correction is mentioned in Section 2.4.6. In order to determine whether the ADS-B messages are error-free, that is, to check the consistency, the bits other than the last 24 bits and the parity bits are subjected to the processes mentioned in the Sections 2.4.2 and 2.4.3. If the parity check result is consistent, the message is divided into parts according to the predetermined message structure, and decoding is applied.

2.4.1. RF Front-End

To collect data and apply further processing to the data in the software environment, an RF Front-End system is established. The parts used in the system are listed in Table 2.18. Moreover, the connection scheme is shown in Figure 2.13, and the images of the system components are shown in Figure 2.14

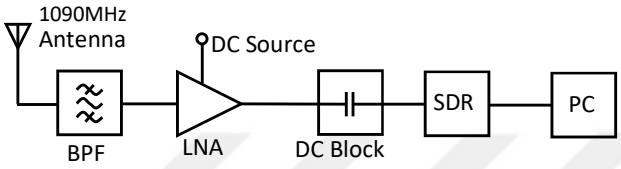
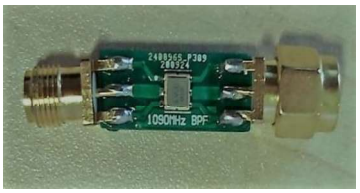


Figure 2.13. Connection scheme.

(a) Antenna.

(b) Radio transceiver (SDR).

(c) LNA.



(d) Bandpass filter.

(e) DC blocker.

Figure 2.14. Component images.

Table 2.18. Component list.

Component	**Preferred Component Type**
Antenna	A designed dipole antenna with a length of 6.875 cm
Bandpass filter	1090-MHz Saw Bandpass filter
Low noise amplifier	+5 VDC Amplifier
External Power Unit	Portable Charger
DC Blocker	50-ohm DC blocker
Radio Transceiver	Adalm-Pluto SDR
Personal Computer	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz

The signals collected on the hardware side are transferred from the software defined radio receiver to the personal computer via USB. The hardware model of the system is completed at this stage. On the software modeling side, the MATLAB environment is being used. The ADS-B raw data acquisition is realized by receiving the data in packets and applying the pulse position demodulation process. Before demodulation, synchronization and preamble detection were performed. These stages include data detection and recording, which implements the classical sequencing such as frame, package, sample, symbol, and bit layers. The demodulated data is saved in a text file as hexadecimal. The text file consists of lines containing 14 or 28 hexadecimal characters, namely short (56-bit) and long (112-bit) packages.

2.4.2. CRC Signal Generation

With Mode-S, signals from both the ground station and the aircraft started to be transmitted with CRC. This is an innovation in Mode-S. It can be used to detect data loss in Mode-S signals and even to correct some corrupted bits. An error-detecting code commonly used in digital networks and storage devices to detect undesired changes in raw data is the cyclic redundancy check. A generator polynomial is used to calculate parity bits in the CRC process. For ADS-B messages, the same generator polynomial is used to generate checksums by the encoder and validate the message by the decoder. The binary and polynomial formats of the generator polynomial are given as

$$\mathbf{G} = [\mathbf{1111111111111010000001001}], \tag{2.8}$$

$$\mathbf{G}(x) = \mathbf{x^{24}} + \mathbf{x^{23}} + \mathbf{x^{22}} + \mathbf{x^{21}} + \mathbf{x^{20}} + \mathbf{x^{19}} + \mathbf{x^{18}} + \mathbf{x^{17}} + \mathbf{x^{16}} + \mathbf{x^{15}} + \mathbf{x^{14}} + \mathbf{x^{13}} + \mathbf{x^{12}} + \mathbf{x^{10}} + \mathbf{x^{3}} + \mathbf{1}. \tag{2.9}$$

The algorithm consists of XOR gates (addition modulo 2) and shift operations. The parity is calculated at the sending side, combined with the 24-bit aircraft address, and appended to the message. The parity is recalculated at the receiving end and compared to the combined parity and address attached to the message to verify that the message was transmitted correctly and designed for that aircraft, and vice versa.

Before moving on to the CRC generation process in detail, it is useful to remember the structure of short and long messages of ADS-B. These structures are given in Tables 2.19 and 2.20.

Table 2.19. 56-bit (14-hex) ADS-B short message.

DF(Downlink Format) 5 bits	**CA(Transponder Capability)-3 bits**	**ICAO(Aircraft Address)-24 bits**	**PI(Parity ID) 24 bits**

Table 2.20. 112-bit (28-Hex) ADS-B long message.

DF(Downlink Format) - 5 bits	**CA(Transponder Capability)-3 bits**	**ICAO(Aircraft Address)-24 bits**	**ME(Message Extended Squitter)-56 bits**	**PI(Parity ID) 24 bits**

The last 24 bits of ADS-B messages always contain parity bits. Therefore, the 24 bits of the obtained error-free 56 or 112-bit messages should be extracted, and the remaining raw message should be inserted into the CRC signal generator algorithm.

The algorithm works as follows:

- An 8 or 22-character hexadecimal sequence is given as input, so, the last 6 hex characters or the last 24 bits are reserved from the whole message.
- Then the generator polynomial $G(x)$ mentioned above is created as a boolean vector.
- The hexadecimal input is converted to binary and appended with a 24-bit zero vector. This completes the message to 56 or 112 bits.
- Then, the CRC parity value in the tail of the package is produced by shifting the generator polynomial over the message bit by bit and using XOR operations simultaneously.
- The last 24 bits of the 112-bit message obtained as a result of this operation are taken. This 24-bit vector is the remainder of the input vector that undergoes division by the generator polynomial. This remainder is given as the final output.

The remainder of the bits from the CRC division operation will be our parity bits. It is expected that these bits to be the same as the last 24 bits in the message sent by the aircraft via the downlink at the end of an error-free process. The summary of the operations in the polynomial format can be shown as

$$M(x) = \sum_{i=0}^{31 \; or \; 87} a_i x^i, a_i \in (0,1), \tag{2.10}$$

$$P(x) = x^{24} M(x) \; \% \; G(x), \tag{2.11}$$

where x^i corresponds to the term representing the message bit [7,30]. The coefficients of the remainder of polynomial division are the artificial CRC parity bits that we produce from the raw message that we extracted the last 24 bits of.

2.4.3. CRC Signal Check and Test

Again, the 24^{th}-degree generator polynomial is used for error detection in the resulting 56 or 112-bit raw message. The CRC error-checking algorithm will give a total error value as a result, we expect this value to be 0 in an error-free message. This control method detects whether there is an error in the short or long message, but does not give a result as to which bits and how many bits contain the error. If we itemize the algorithm,

- 14 or 28 hexadecimal character inputs are received, this time the entire message is input, with the last 24 parity bits.
- The hexadecimal input is converted to binary, resulting in a message of 56 or 112 bits.
- This binary input is checked by the generator polynomial as in CRC generation.
- The last 24 bits obtained as a result of CRC operation are saved, and the numerical sum of these 24 bits is taken.
- If the total value is 0, the specific long or short message collected from the receiver and inserted into the algorithm has been received without error. If the sum is an integer value other than 0, the message contains the erroneous bit or bits.

The summary of the operation in the algorithm can be expressed as

$$M(x) = \sum_{i=0}^{55\ or\ 111} a_i x^i, a_i \in (0,1), \tag{2.12}$$

$$P(x) = M(x)\ \%\ G(x), \tag{2.13}$$

where the error is found by summing the coefficients of the remainder of the polynomial division because the remainder should be zero for error-free messages. These coefficients are the last 24 bits of the output vector of the CRC operation.

During the CRC signal test phase, the parity bits in the messages that are found to be error-free are compared with the artificially produced parity bits and checked to see whether they are the same. This stage is the accuracy test of the CRC generation and control algorithms. Briefly, with the CRC process performed, Mode-S messages containing hexadecimal 14 or 28 characters in a text file can be received, and the total number of ADS-B messages with zero error and the ratio of ADS-B messages with zero error to all Mode-S messages can be calculated. At the same time, the percentage match rate of the original CRC parities and artificially produced CRC parities in error-free messages can be calculated, and displayed in the command window. The CRC generation and check algorithms tested in the software simulation environment have created a kind of algorithm testing environment by giving the results mentioned above.

2.4.4. Testing the CRC Process with Collected Data

CRC parts of both 56 and 112-bit package data are taken, and their correctness is checked with artificially produced CRCs. In Figures 2.15 and 2.16, 56 and 112-bit message samples are shown in microseconds on the X-axis with an 8-microsecond preamble added, while the Y-axis gives the magnitude of the pulses collected in the form of I-Q.

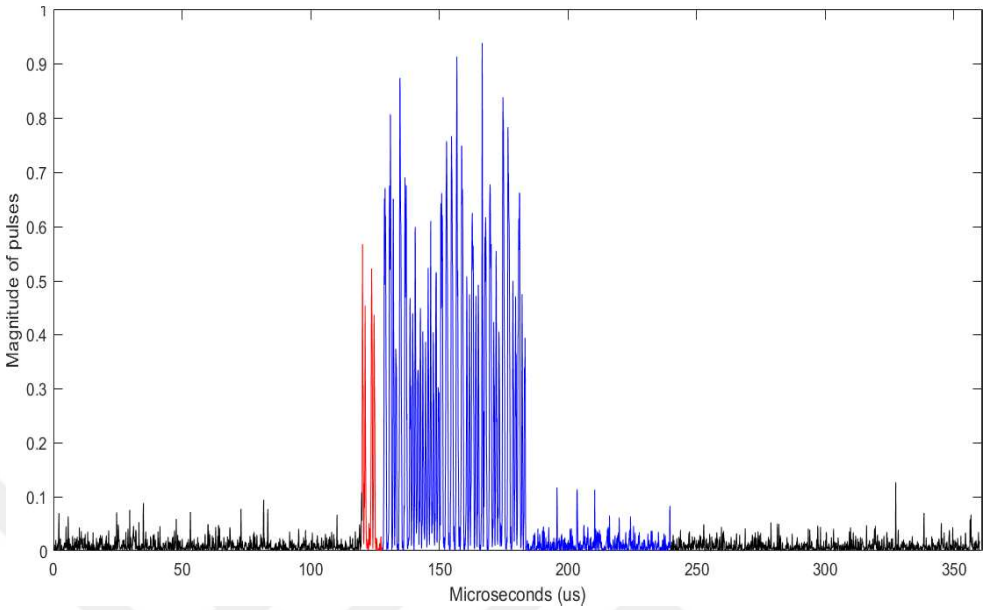


Figure 2.15. Real-time collection of 56-bit short data.

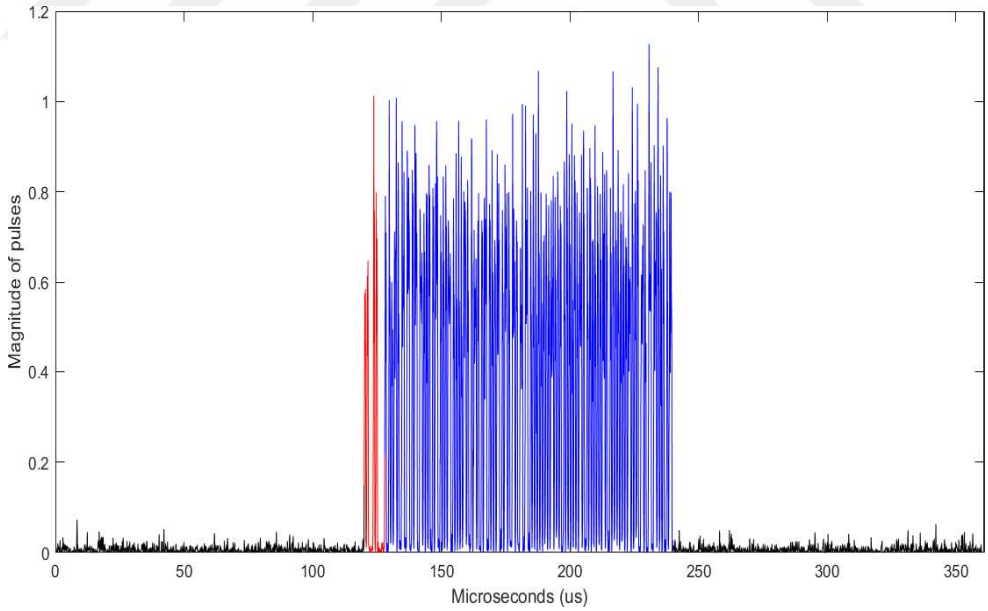


Figure 2.16. Real-time collection of 112-bit long data.

While the 8-bit preamble samples are represented in red, the samples shown in blue contain the 112-bit collected message. Black signals outside of 8+112 bit data are noise. In the short message example in Figure 2.15, it is seen that half of this 112-bit blue window is filled with noise as expected. Since the sampling frequency was

determined as 12 MHz, $120 \times 12 = 1440$ samples were collected for an $8+112 = 120-bit$ message. In the figures, there are $3 \times 1440 = 4320$ samples in total, the purpose of which is to display the preamble-detected signal together with the samples that come before and after it.

The hexadecimal values of the sample data shown above were recorded during the real-time data collection phase, and these values can be seen in Table 2.21.

Table 2.21. Raw message and CRC separation.

Hexadecimal full message	**Message with CRC removed**	**CRC**
5D4001551CBD47	5D400155	1CBD47
8D4BC8419909B500780C0601ED79	8D4BC8419909B500780C06	01ED79

A summary of the control scheme is as follows. Short and long messages with CRC removed are inserted into the CRC-generator function and CRC is generated artificially. Then the generated CRCs and the actual CRCs are inserted into the CRC comparison function. If the inserted CRCs are the same, there is no error.

When both the short and long messages in the example are inserted into the CRC generator, we get the results '1CBD47' and '01ED79', respectively. Therefore, the comparison function says successful for both examples. Multiple samples can be tested as well as a single hexadecimal data entry. At this point, 10100-odd data collected in real-time were put into the flow outlined above and tested, and a hundred percent error-free result was obtained. This +10100 data is from aircraft with 53 different ICAO addresses.

2.4.5. Related Works on Error Correction Using CRC

A CRC can be employed as a checksum to find unintentional changes to data during transmission or storage. An important benefit of parity encoding in Mode-S communications is that it prevents the sensor retransmission penalty in the event of

an error. In order to safeguard against potential interference, Mode-S designers also had to incorporate some type of redundancy in message encoding.

Binary polynomial arithmetic is the foundation of the CRC algorithm. The binary polynomial division is used to calculate the remainder by employing a predetermined divisor, commonly referred to as a "generator". The message includes the CRC in the final portion of it. The receiving party can do the same operation on the complete message to determine whether the remainder is zero. A considerably more effective method for error detection is CRC. It also offers the ability to correct errors [31].

Burst/group errors are the most prevalent error type on Mode-S signals when the error categories and how frequently they occur are considered. Instead of being a single or two-bit single error sequence, a burst error is a sequential or clustered multi-bit error. The main causes of burst failure are signal fading and interference. Interference in ATCRBS responses is the main cause of burst errors in Mode-S communications. Any cyclic code with a generator polynomial can recognize burst errors up to the polynomial's highest order [32]. The CRC-24 polynomial, for instance, which was utilized in the preceding sections, can identify burst errors up to 24 bits. Lincoln Labs researchers made the assumption that Distance Measuring Equipment and Tactical Air Navigation pulses might result in error groups as long as 19 bits, and ATCRBS pulses could result in 5-bit error groups while developing Mode-S [33]. As a result, the interference environment is a good fit for this parity check approach.

The simplest method that can be used for error correction is to change those bits for each error combination and CRC test. This method can be slow and inefficient since no prior knowledge is used but considering that the messages are pulled from the preamble's locations, not many errors are expected in the message. However, when the message overlaps with another signal after the preamble, that is, when interference occurs, the message may be very wrong even if the preamble is found to be correct. Interference can cause errors in data transmission. In such cases, the interference location can be estimated by observing the magnitude of the energy throughout the

message. This interference location information can be effective in error correction.

The bit-confidence declaration is generally used by bit-flipping algorithms to conduct bit-flips on low-confidence bits by testing all or part of low-relying bit patterns [34]. Reliability calculation can be done by calculating the score during PPM demodulation. A low-confidence bit pattern is created from a low-confidence vector. CRC syndrome is calculated from the received code word. The CRC generator polynomial is moved over the syndrome, and a new vector is created with the two modulo sums (XOR). With the newly created vector, the low confidence bit pattern is compared with a certain principle, and it is decided whether the error can be corrected [35].

Generator polynomials can be used for error detection or both error detection and error correction codes according to their properties. In the article [31], it is explained whether different CRC polynomials have these properties and how to correct them by creating an error table. For example, for a single-bit correction, every bit of a correct message consisting entirely of 0s is changed in sequence and the CRC operation is applied. A table is created from the errors for each bit. If the result is in the table after CRC is applied to a new message, the bit corresponding to that error is corrected, and the erroneous message is corrected. In fact, for this method to be valid, the CRC result must be different when each bit is changed for the 0 messages. Otherwise, that CRC polynomial is not used in error correction because if it gives the same result for two different errors, it is unclear which bit should be corrected.

In the project report from Lincoln Laboratory, the downlink encoding and the error correction method using both the error syndrome table and confidence bits were examined [30]. To summarize the method, the parity/address space AP is expressed as $AP = R(x)+A(x)$. $R(x)$ is the remainder of the division with the generator polynomial used on the transmit/receive side. $A(x)$ is the original error-free aircraft address. Then the downlink message to be generated would be: $D = x^{24} \times M(x) + R(x) + A(x)$. Three operations are applied to the message of this nature, respectively: the resolved remaining $R(x)$ and the expected address $A(x)$ must be compared to detect whether

there is an error or not. This process is the same as putting the whole message signal into CRC check in section 2.4.3, only this time the CRC error check is done for the aircraft address. Then, the location of the errors is determined with the help of receiver reliability calculation. Error correction is done as the last operation. If an error is detected, the remaining address $R(x)$,$(D(x) - A(x))$ is called the error syndrome and is put into the next two operations. The fact that the error syndrome is nonzero is proof of the existence of the error. There are low confidence bits in the 24-bit data sequence of the message if the syndrome identifies a burst failure. As a result, the erroneous bits' location is identified, and those bits are chosen to be input into the correction process. Multiple burst error intersections in the same 24-bit short sequence or faults in high-confidence bits will make location identification challenging, resulting in inaccurate correction.

2.4.6. Developed Error Correction Method Using CRC

Each aircraft has its own ICAO address, which is included in every ADS-B message. As a result, the ICAO addresses can be used to count the number of different aircraft as well as compare and correct erroneous ADS-B messages. Because of the garbling and collision of different aircraft Mode-S signals, every message could not be entirely correct or the CRC check holds. Most ADS-B message detection methods begin by attempting to find the preamble part using match filters or correlation with known synchronization sequences. Then, the packages located at the maximum correlation points are extracted. Finally, the PPM demodulation and CRC check are done. If the CRC check holds, i.e., the last 24 bits are equal to zero after the sliding two-modulo sum operation, the decoding part begins. However, if one examines the data coming from aircraft, the error-free message ratio is very low. The error correction method was developed after seeing the large number of erroneous messages we received during data collection. To increase the number of correct messages, two new methods are suggested. These methods were used on collected data, allowing the correct messages to be used in the simulation environment and to move the simulation environment further. The first method with brute force bit traveling and flipping is described below.

- Step 1: As previously stated, ICAO numbers are unique to each aircraft, and they are included in every ADS-B message. These properties can be used to correct erroneous messages. Firstly, there should be an ICAO list kept from real-time error-free ADS-B messages. If the CRC check holds and the current ICAO is not on the list, then this ICAO is appended to the list. When the erroneous message comes, its ICAO part should be compared with all the list members. The most similar ICAO in the list should be appointed to the ICAO part of the erroneous message. For the second method, there is a minimum threshold for the similarity between two ICAOs because new aircraft entering the airspace should also be considered. The list must be kept in a binary matrix to speed up the algorithm, and the maximum similarity can be found by the XNOR-equivalence operation between the current binary ICAO vector and the binary matrix rows.
- Step 2: Besides the ICAO part, all ADS-B Extended Squitter messages that have 112 bits come with a downlink format as DF 17, which is 10001 in binary. In addition, the capability of the messages detected during data collection is usually 5, 101 in binary, corresponding to Level 2+ transponder and airborne. Then, one can simply set the first two hexadecimal digits of a long ADS-B message to "8D", which is the concatenation of the above downlink format and capability. The derivation of the "8D" can be shown as

$$17 = 10001 \quad \rightarrow \quad Downlink\ Format, \tag{2.14}$$

$$5 = 101 \quad \rightarrow \quad Capability, \tag{2.15}$$

$$1000\ 1101 = 8D \quad \rightarrow \quad First\ Two\ Hexadecimal\ Digits. \tag{2.16}$$

For the short messages, the downlink format is 11, 01011 in binary, and one can set the first two hexadecimal digits of short messages to "5D" too. The derivation of the "5D" can be represented as

$$11 = 01011 \quad \rightarrow \quad Downlink\ Format, \tag{2.17}$$

$$5 = 101 \quad \rightarrow \quad Capability, \tag{2.18}$$

$$0101\ 1101 = 5D \quad \rightarrow \quad First\ Two\ Hexadecimal\ Digits. \tag{2.19}$$

Although these two steps correct a very small number of messages, these corrections are required to enable further processing of the remaining 80 bits. Downlink format, capability, and ICAO have 5, 3, and 24 bits, respectively.

- Step 3: Further processing of 80 bits can be described as follows. Firstly, the inverse of the first bit of 80 bits is taken. If the CRC check holds after this operation, the correction of this message is done. Otherwise, the inversion and CRC check operations are done for the remaining 79 bits with the first bit kept as inverse. Then, these operations are done for the second bit and the remaining 78 bits, for the third bit and the remaining 77 bits like this. This means that if the correction could not be done, all the loop sums up to

$$80 + 79 + 78 + \cdots + 1 = 80 \times \frac{(80+1)}{2} = 3240. \tag{2.20}$$

However, without correcting the DF, CA, and ICAO part, the loop sums up to

$$112 + 111 + 110 + \cdots + 1 = 112 \times \frac{(112+1)}{2} = 6328. \tag{2.21}$$

This means, in that case, there are $6328 - 3240 = 3088$ number of redundant loop elements. The data processed contains 8200 messages with 1606 of them correct ADS-B messages without any operations. In the worst case assuming all the messages are long and errors could not be corrected, $(8200 - 1606) \times 3088 = 20362272$ loop elements are avoided. For every erroneous message, correction of the DF and capability part is just one operation. The similarity check with the ICAO list is also a low-cost process because the list has a very small number of elements even with the dense air traffic in Istanbul. Also, the matrix operations speed up the ICAO correction part.

These three steps are applied in combined form, and for 8200 Mode-S message data, 430 more correct ADS-B messages are found. This means our correct ADS-B message ratio has risen from $1606/8200 = 0.1959$ to $(1606 + 430)/8200 = 0.2483$.

For the second method developed, the first two steps of the first method are applied again. For the remaining part of the method, firstly, the table construction

is explained. The zero message, containing 112 zeros for long and 56 zeros for short, is an error-free message. This means that the remaining checksum is all zero after the CRC process. Each bit is flipped for the one-bit error, and the CRC checksum is recorded in the error correction table. In the two-bit error case, bit flipping and CRC checksum recording are performed again for each binary combination of 112 bits. However, only burst (group of consecutive bits) errors are considered for three-bit errors, because the checksums of different messages with three-bit errors may be the same. The same checksum breaks the error correction procedure due to the random decisions of false bit positions. For each checksum or syndrome, the false bit positions are also recorded for error correction. The calculation for the overall table elements for a long message can be shown as $112 + comb(112, 2) + 110 = 6438$, where the left side represents one-bit, two-bit, and burst three-bit error cases, respectively. The third step of the first method is changed to finding the checksum by applying the CRC process to the erroneous message in which the first two steps were applied and comparing the found checksum with the constructed table. In this method, if the ICAO part does not pass the similarity threshold, the checksum is compared with the table part where errors in ICAO bits also exist. But if the ICAO has passed the similarity threshold and been changed, the checksum will only be compared to the error table part with errors after the 32^{nd} bit. Then, if the matching occurs, corresponding bits in the error locations are flipped. Otherwise, the method ends with the decision that this message cannot be corrected. Finally, the CRC check is done again for the sake of completeness.

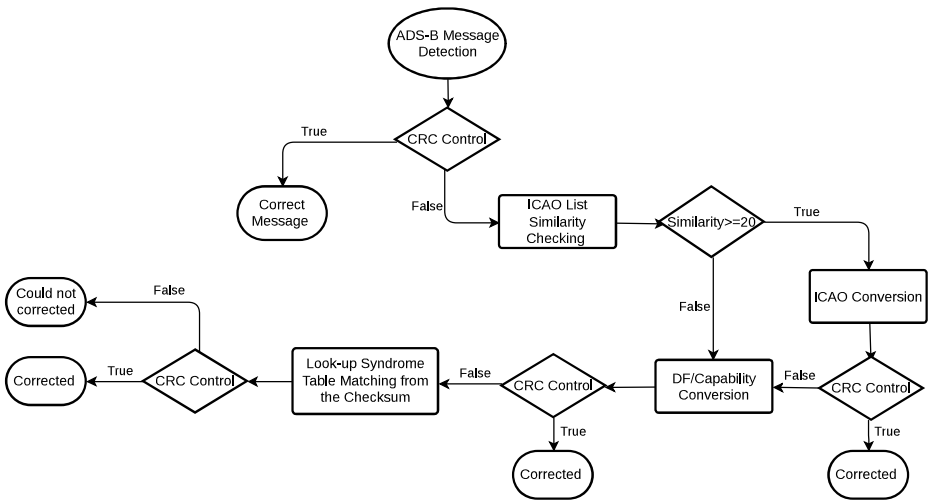


Figure 2.17. The flow diagram of the second method for error correction.

The flow diagram of the second method proposed is shown in Figure 2.17. This method was examined in 7 different records and a total of 59175 Mode-S messages. During the tests, different limits for the minimum ICAO similarity were tried. Overall, the best-performing limit is 20. This means that ICAOs have at least 20 bits in common. The results of the second method are given in Table 2.22, with the overall results shown in the last row.

Table 2.22. The results of the second method for error correction.

Experiment No	**Number of Correct Messages w/o Correction**	**Number of Correct Messages after Correction**	**% Increase in Ratio**
1	1606	2082	29.64
2	737	964	30.80
3	865	1127	30.29
4	927	1131	22
5	591	750	26.90
6	1200	1618	34.83
7	2925	3967	35.62
Overall	**8852**	**11639**	**31.48**

Although the second method provided a slight performance increase compared to the first method, it made a significant contribution in terms of execution time. The second method is a much more time-efficient algorithm. For instance, for the first experiment, data containing 8200-message, the first method lasts more than 5 minutes, whereas, the second method gives the result approximately in 5.2 seconds. During the second method, single, two-bit, and consecutive three-bit errors are corrected using the CRC syndrome with a look-up table. Furthermore, DF/Capability and ICAO conversions can provide correction on more than three-bit errors. These two methods are correcting only ADS-B messages from the Mode-S signals.

3. MODE-S RADAR INTERROGATION ALGORITHM DESIGN

3.1. Selectivity and Garbling

Mode-S transponders have 255 different 56-bit data storage caches called BDS registers, and the contents of these BDS registers are updated periodically by a flight management system [36]. Mode-S interrogators have two main tasks, one being to detect the replying aircraft and another being to extract more information from a detected aircraft. Using a unique ICAO and getting all necessary information from the minimum number of replies reduce the interrogation density per aircraft. When the ground station is able to detect an aircraft with at most two interrogations, this results in avoiding Mode A/C overlaps, and hence selective interrogations could be further enabled.

Every region has a lockout map in an SSR Mode-S network. Lockout command is given by a ground station to a selected aircraft. The selected aircraft stops responding to other calls when it is in a lockout state [4]. The lockout map determines the area where the ground station locked up an aircraft. Generating a lockout map and managing these maps with multiple cities are challenging tasks since the number of locked-up aircraft is constantly changing [37].

All-call interrogations are used for the initial communication when an aircraft enters the radar coverage. All-call interrogation signals do not have a specific destination; hence all non-locked-up aircraft must respond to this call. A ground station periodically radiates all-call interrogations with a maximum rate of 250 Hz [37]. After receiving a reply from an aircraft, the ground station extracts the specific aircraft's rough position and ICAO address. The next step is to follow the aircraft and interrogate it with a roll-call. Roll-call interrogations are aircraft-specific interrogations once the aircraft is in a lockout state. This methodology helps ground stations to prevent

unnecessary all-calls and reduce garbling effects. The proposed Mode-S interrogation algorithm comes in handy at this stage since collecting all the all-call replies from aircraft in the dense region is an issue due to interrogating with a hundred percent probability.

3.2. Stochastic Interrogation Algorithm

The algorithm always works better in the dense air traffic regions than the current Mode-S radar uplink interrogation scheme, as the perception is that aircraft will not be closer than some limited distances. If any close encounter happens, this situation will be short, and the garbling effects will be temporary. However, the developed algorithm takes into account the fact that air traffic density will rise in the future and that there can be close encounters in the air with the addition of new aerial vehicles like drones designed for specialized applications, airbuses, etc. By this logic, the current mechanism cannot deal with highly dense air environments as the number of aircraft increases. One possible solution to this futuristic dense air traffic problem in civil aviation is to interrogate with constant probabilities such as 0.5, 0.25, 0.125, and 0.0625 as Mode-S uplink communication allows [38]. In this approach, constantly interrogating with 0.5 uplink requests would require too many interrogations while trying to detect all aircraft if the number of aircraft is too large. On the other hand, interrogating with 0.0625 is inefficient if there are not many aircraft in the target region. However, in dense environments, after detecting a number of aircraft by interrogating with a lower probability, one can fasten the detection process by increasing the probability contained in the interrogation. When a lot of garbling is received, the probability of intercept can be decreased. An algorithm is proposed with these adaptive changes in such situations that will work much more efficiently than the static interrogation method. In the proposed method, the density, low or high, of the remaining aircraft can be estimated by their overall responses for one or more, depending on the control length and interrogations. For instance, if we are interrogating with 0.0625 and getting no replies, we should interrogate with a higher probability. In this algorithm, the feature of Mode-S is also used; when the aircraft is locked-out, it will not reply to

interrogations until a particular time passes. Furthermore, the proposed algorithm will decrease the need for operator intervention.

The stochastic interrogation working principle is like this when the probability of intercept contained in the interrogation is p.

- An aircraft replies with probability p,
- An aircraft does not reply with probability $1 - p$.

Theoretical calculations of static interrogations to detect all aircraft in the dense region can be seen by using binomial distributions with the below definitions.

$P_{DETECTION}$ = Probability of only 1 aircraft replying
$P_{NO-RESPONSE}$ = Probability of getting zero responses
$P_{GARBLING}$ = Probability of collision due to multiple replies

The formulas of these definitions can be expressed as

$$P_{DETECTION} = (1-p)^{(N-1)}pN, \tag{3.1}$$

$$P_{NO-RESPONSE} = (1-p)^{N}, \tag{3.2}$$

$$P_{GARBLING} = 1 - P_{DETECTION} - P_{NO-RESPONSE}, \tag{3.3}$$

$$P_{GARBLING} = 1 - (1-p)^{(N-1)}(pN + (1-p)). \tag{3.4}$$

Here, N is the number of aircraft in the dense region. The number of all-call interrogations with a static p probability of collecting all all-call replies from aircraft in the dense area becomes $M_{DETECTION}(p)$ which can be expressed as

$$M_{DETECTION}(p) = \frac{1-p}{p}\sum_{k=1}^{N}\frac{1}{(1-p)^k k}. \tag{3.5}$$

As seen from Equation (3.5), the total number of interrogations steeply increases as N grows very large. When N is small, the minimum number of required interrogations is best with 0.5 static probability. Considering both small and large N cases, an adaptive approach is needed to utilize the best performances of all static probabilities ranging from 0.5 to 0.0625. Therefore, an adaptive algorithm is required to solve this futuristic

dense air traffic interrogation problem without requiring any hardware changes. The pseudocode of the developed algorithm can be seen in Figure 3.1.

In the pseudocode, the control length is 1. This means that when there is no reply for only one interrogation, the probability of interrogation (POI) is increased, or when garbling happens, the POI is decreased. When the control length is n, after n consecutive interrogations with no reply, the POI is increased. For probabilities 1 and 0.0625, some exceptions exist. Also, for probability 0.5, when detection occurs, the POI is increased again. Typically, there is no probability change in detection cases.

3.3. Performance of the Algorithm

The algorithm runs with three different parameters for each Mode-S SSR specification, namely pulse repetition frequency, RPM, and beam-width. The selected parameters are given in Table 3.1. These parameters are selected according to the values used or studied for industrial airport radar antennas and ATCRBS antennas [27,29,39].

Table 3.1. Basic Mode-S SSR specifications that are used in simulations.

Parameter	**Selected Values**
PRF (Hz)	[150, 225, 300]
RPM (1/min)	[6, 10, 15]
Beam-width (degree)	[1.2, 1.8, 2.4]

Monte Carlo simulation with 1000 iterations is run for each number of aircraft ranging from 2 to 20; therefore, a total of $3 \times 3 \times 3 \times 19 = 513$ combinations is investigated in the results. Both the adaptive probabilistic interrogation algorithm and theoretical static interrogations are simulated to compare the performance in terms of time spent detecting all aircraft in the dense air environment. The most important metrics to consider are the time spent collecting all ICAO addresses during all-call interrogations and the mean number of interrogations to receive aircraft replies one by one without garbling.

start procedure
Step 1 Interrogate with $P = 1$
    if number of aircraft replied == 0 **then**
        return
    else if number of aircraft replied == 1 **then**
        Repeat Step 1
    else if number of aircraft replied > 1 **then**
        Go Step 2
    end if
Step 2 Interrogate with $P = 1/2$
    if number of aircraft replied == 0 or 1 **then**
        Go Step 1
    else if number of aircraft replied > 1 **then**
        Go Step 3
    end if
Step 3 Interrogate with $P = 1/4$
    if number of aircraft replied == 0 **then**
        Go Step 2
    else if number of aircraft replied == 1 **then**
        Repeat Step 3
    else if number of aircraft replied > 1 **then**
        Go Step 4
    end if
Step 4 Interrogate with $P = 1/8$
    if number of aircraft replied == 0 **then**
        Go Step 3
    else if number of aircraft replied == 1 **then**
        Repeat Step 4
    else if number of aircraft replied > 1 **then**
        Go Step 5
    end if
Step 5 Interrogate with $P = 1/16$
    if number of aircraft replied == 0 **then**
        Go Step 4
    else if number of aircraft replied >= 1 **then**
        Repeat Step 5
    end if
end procedure

Figure 3.1. Adaptive probabilistic interrogation algorithm.

Using the proposed algorithm and static interrogations in Monte Carlo simulation in order to compare the performance of these approaches in terms of the number of all-call interrogations, the results can be seen in Figure 3.2.

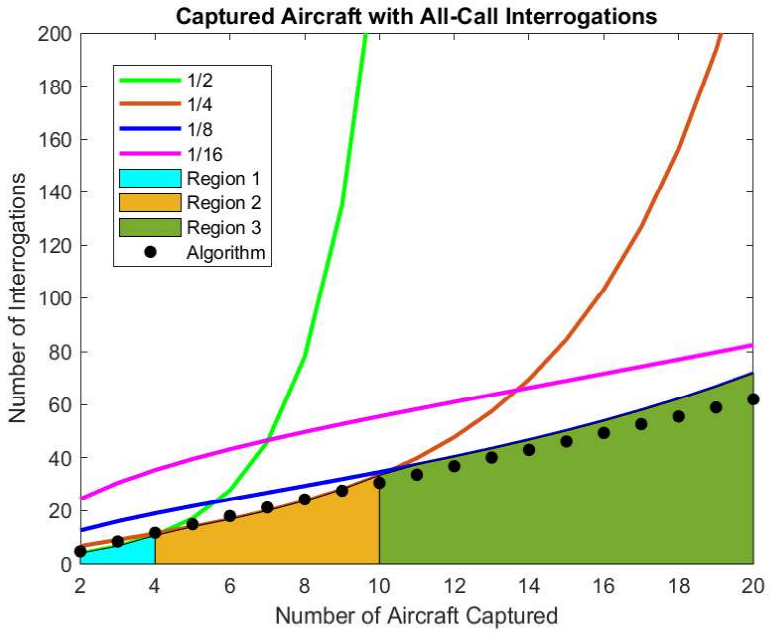


Figure 3.2. Overall performance of adaptive algorithm vs static interrogations.

The graph is divided into three parts according to the density of the aircraft in the target area. Interrogating with static 0.5 probability all-calls gives the best performance in the less dense blue part since the garbling effects are diminished due to the small numbers of aircraft. Using low probabilities such as 0.125 or 0.0625 to detect a few aircraft in the blue region is inefficient. Interrogating with static 0.25 probability gives the best result in the dense, orange part. However, interrogation with static 0.125 or 0.0625 performs the best in the densest, green, region since the number of aircraft is too many and garbling effects are immense in low probabilistic all-calls. All the static interrogation approaches tend to perform well within defined colored regions. However, the proposed algorithm provides the best results when there are more than 10 aircraft in a dense environment, and moreover, it comes second best when there are few aircraft.

3.4. Effects of Basic Radar Specifications on Performance

Simulation results of the adaptive interrogation algorithm in terms of different RPM, PRF, and BW parameters are shown in Figures 3.3 to 3.5.

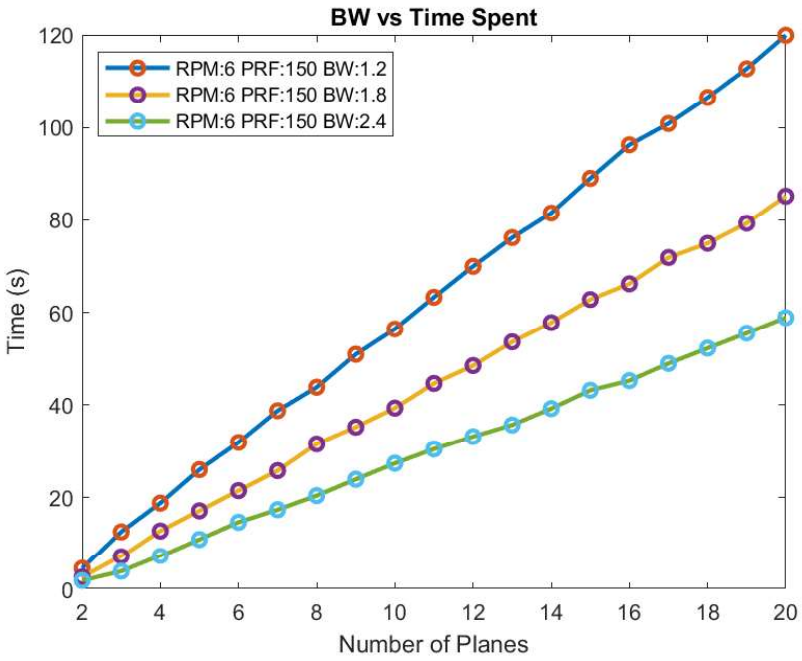


Figure 3.3. Effect of beam-width on the performance of the Adaptive Algorithm.

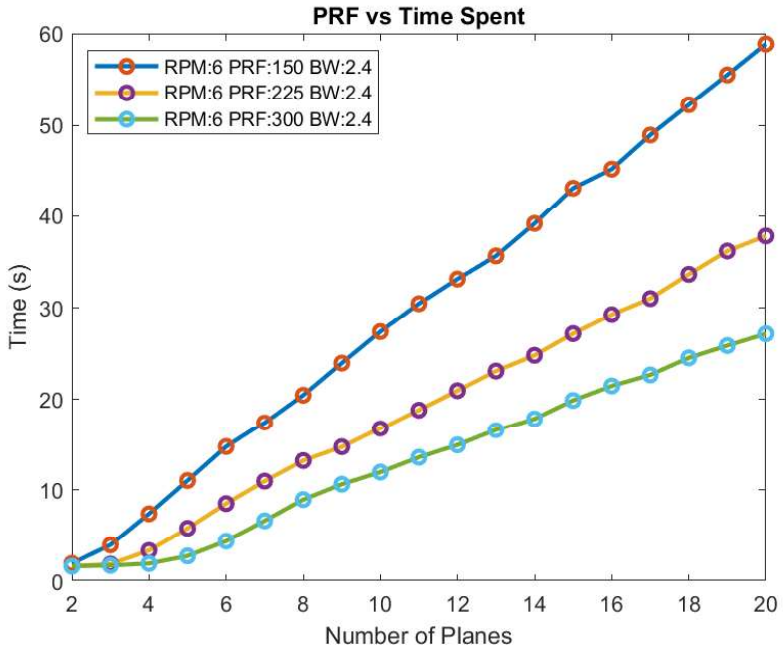


Figure 3.4. Effect of PRF on the performance of the Adaptive Algorithm.

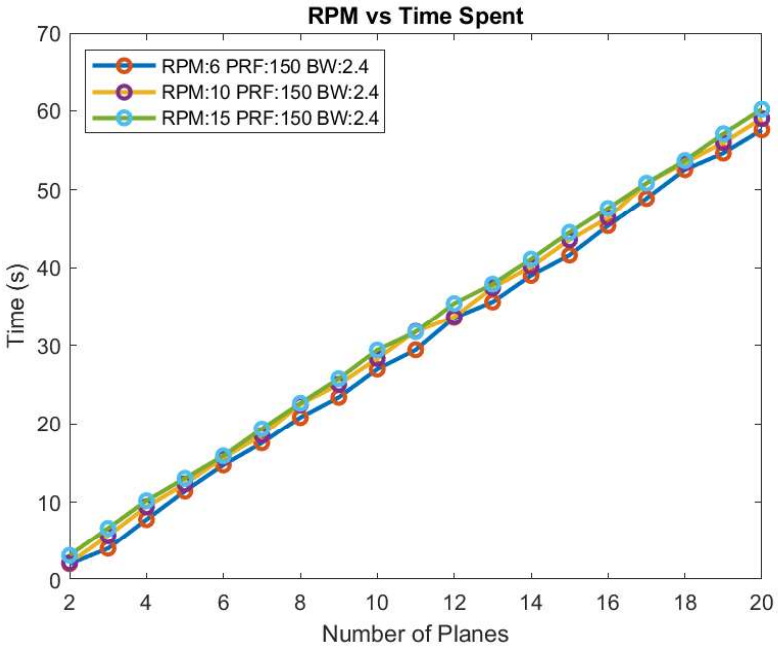


Figure 3.5. Effect of RPM on the performance of the Adaptive Algorithm.

In all three figures, the time spent to detect all aircraft in the dense garbled region increases as the number of aircraft increases. Here, the performance metric is the time rather than the number of interrogations, and the time spent per pulse, T_p, and total time spent for a complete interrogation, T_{ci}, are calculated as

$$T_p = \frac{1}{PRF} \tag{3.6}$$

$$T_{ci} = Number\ of\ completed\ interrogations \times T_p \tag{3.7}$$

As observed in Monte Carlo simulations with 500 iterations, parameter changes do not affect the number of interrogations required to detect all aircraft. For this reason, since the increase in PRF will shorten the time taken for interrogation, the total time spent will also decrease. The situation is similar in the beam-width case: As the increase in beam-width will increase the number of interrogations per aircraft in one tour, the number of tours taken will be shortened, which will shorten the total elapsed time. The RPM indicates how many seconds it takes to complete a lap. For this reason, PRF is divided by RPS (rounds per second) to find the total number of pulses sent in each round. To find the number of pulses in each beam, the total number of pulses in

one round is divided by the total number of beams. Although the increase in RPM causes the antenna to rotate faster, we cannot say that there is a dominant correlation between the total interrogation time and RPM. The reason for that is the increase in rpm reduces the number of pulses in a beam. For selected parameters, the total time spent is shown in Figure 3.6. The proposed algorithm performs the best after the number of aircraft exceeds 10. It is observed that it gives very close to the best results in other places.

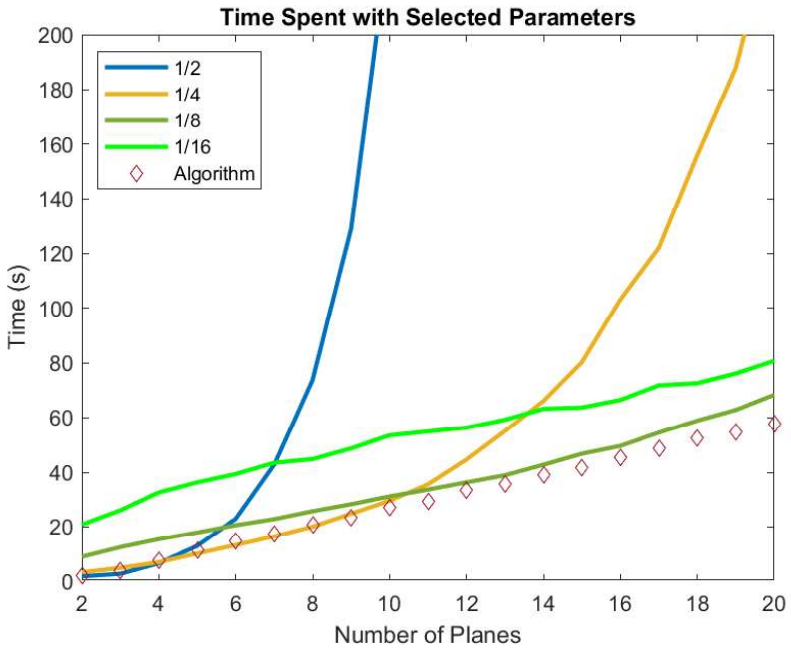


Figure 3.6. The performance of the algorithm and static methods with selected parameters, i.e., PRF: 150 Hz, BW: 2.4°, and RPM: 6 min^{-1}.

3.5. Limited Lockout

The lockout condition is that when an aircraft replies to an all-call message, the ground station locks up that aircraft in order to interrogate with roll-call messages, and this lockout duration takes approximately 18 seconds [38]. During this lockout duration, the detected aircraft is not interrogated once again with all-call uplink requests; therefore, it is deduced that an aircraft is found in the dense air traffic region, and it shall not be interrogated with all-call messages. Once the eighteen-second lockout duration ends, that specific aircraft is interrogated again if it reenters the radar

coverage area. By doing this, re-entries to the radar coverage are taken into account in the vicinity of sections where two different ground stations intersect with each other's coverages.

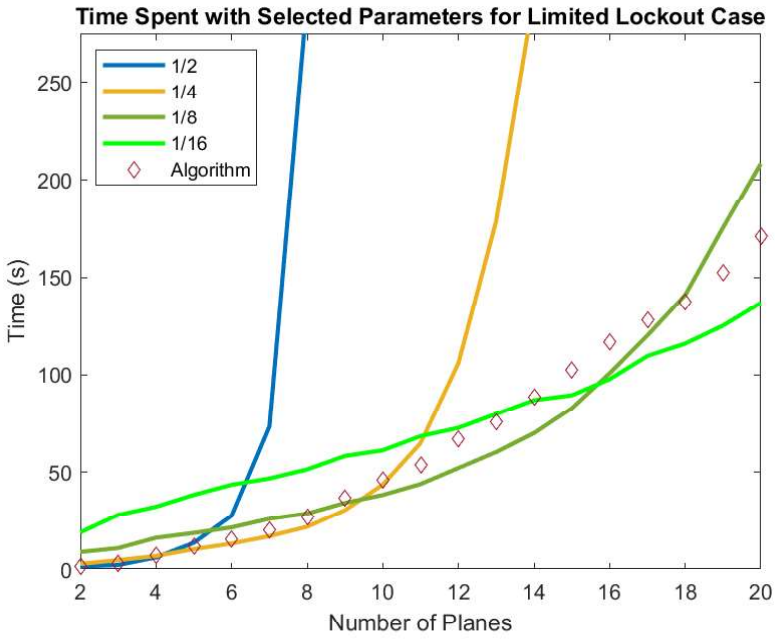


Figure 3.7. The performance of the algorithm and static methods for limited lockout case with PRF: 150 Hz, BW: 2.4°, and RPM: 6 min^{-1}.

In Figure 3.7, the performances are given in terms of total time spent using selected parameters. The adaptive algorithm also works as the second best most of the time in different density regions. Nevertheless, the adaptive algorithm comes on top since the information about the number of aircraft in the region is not required, and this is handy for the operator in the GS. An additional improvement can be made by taking care of entering and exiting time instances of an aircraft in the dense region rather than just interrogation probability switching. There is a lockout override scenario in addition to the lockout situation, where the interrogator forces a transponder to respond to all all-calls. Stochastic interrogation needs to be used to avoid garbling in lockout override operations. This method is called stochastic lockout override acquisition (SLO) and used when radar coverage clusters of different GSs with the same ICs overlap with each other [40]. While implementing SLO, the GS uses the different PR fields to avoid RF pollution since GS will receive an immense amount of replies due to overriding.

3.6. Experimental Results

The time spent or the number of interrogations sent while gathering ICAO addresses of all aircraft in the dense region varies for different numbers of aircraft. The statistical performance results of the 0.5, 0.25, 0.125, and 0.0625 probability strategies, including the proposed algorithm, in terms of the number of all-call interrogations, can be seen in Table 3.2. For regions 1 (2-4 aircraft) and 2 (5-10 aircraft), the proposed algorithm gets better as the number of aircraft increases in the region, and for region 3 (11-20 aircraft), the proposed algorithm clearly outperforms other static approaches.

Table 3.2. Number of interrogations to detect aircraft for the constant 0.5, 0.25, 0.125, and 0.0625 probability methods and the proposed algorithm.

Number of aircraft	**0.5 method**	**0.25 method**	**0.125 method**	**0.0625 method**	**Algorithm**
2	4.00	6.67	12.6	24.5	4.7
3	6.67	9.03	16.0	30.6	8.3
4	10.7	11.4	19.0	35.4	11.6
5	17.1	13.9	21.8	39.6	14.9
6	27.7	16.7	24.3	43.2	18.0
7	46.0	20.0	26.9	46.6	21.2
8	78.0	23.7	29.4	49.8	24.3
9	135.0	28.1	32.0	52.8	27.6
10	237.3	33.5	34.7	55.6	30.6
11	423.5	40.0	37.4	58.4	33.7
12	764.8	47.8	40.3	61.1	37.0
13	1395	57.5	43.4	63.8	40.0
14	2565	69.6	46.7	66.4	43.0
15	4749	84.5	50.1	69.0	46.2
16	8845	103.2	53.8	71.7	49.4
17	16556	126.7	57.8	74.3	52.7
18	31119	156.2	62.1	77.0	55.6
19	58714	193.6	66.8	79.7	59.0
20	111140	240.9	71.8	82.4	62.1

4. CONCLUSION

In this thesis, a comprehensive simulation environment was realized to model air traffic. Uplink and downlink formats of Mode-S messages are created according to the information of each aircraft and known message structures. The message timings with different scenarios of aircraft locations are examined for the garbling cases. The results in Section 2.3 show that the simulation is working successfully, and it is a good example for modeling air traffic and testing on future scenarios. A real-time data collection system, radar receiver setup, was established to use in this simulation and also to improve the simulation by utilizing a software defined radio operating at 1090-MHz. For this system, both hardware and software sides are built. A novel error correction method, described in Section 2.4.6, has also been developed to correct erroneous ADS-B messages encountered in this data collection system. A look-up syndrome table and the unique features of ADS-B messages, such as the ICAO address, DF, and capability for matching are utilized in the proposed method. The measurement results show that the second algorithm proposed improved the correct ADS-B messages by 31 percent. Additionally, the algorithm operates more than 50 times faster than the bit-flipping algorithm, the first method introduced, in a standard processor. For further work, instead of keeping the ICAO list online, we can have a very large ICAO list beforehand to reduce computational load during work.

Furthermore, the results from Section 3.6 show that the proposed adaptive algorithm can be quite useful when considering future needs and ATM (air traffic management) or UTM (UAS traffic management) control strategies. Such air traffic control strategies are required as the air traffic environment becomes denser year after year. As some UAVs start to use Mode-S communication protocols as well, e.g., ads-95 ranger drones of the Swiss Airforce, the need for ICAO registration of this type of vehicle increases as they enter airspaces [41]. Therefore, the increase in air traffic reveals the importance of probabilistic interrogations in Mode-S protocol. In this thesis, the theoretical and computational results of probabilistic interrogations are obtained. At the

same time, an interrogation algorithm is proposed that both reduces the operator requirement and works well without many static interrogations. This algorithm is tested with the simulation created by placing all aircraft in one antenna beam to create futuristic dense environment. The proposed algorithm works adaptively according to the density of the aircraft. This adaptive operating principle will facilitate the adaptation to future dense air traffic environments and reduce the detection time of the aircraft in air control and monitoring applications.

REFERENCES

1. Furuno Technology, "Radar Basics", 2014, https://www.furuno.com/en/technology/radar/basic/, accessed on September 2, 2022.

2. Wolff, C., "Radar Principle - Radartutorial", https://www.radartutorial.eu/01.basics/Radar%20Principle.en.html, accessed on September 3, 2022.

3. Wikipedia, "Secondary surveillance radar", https://en.wikipedia.org/wiki/Secondary surveillance radar, accessed on October 3, 2022.

4. Bodart, J., "Mode S Surveillance Principle", *ICAO Surveillance/MICA Workshop*, Cairo, Egypt, 2019.

5. Baker J.L., V.A. Orlando, W.B. Link and W.G. Collins, "Mode S System Design and Architecture", *Proceedings of the IEEE*, Vol. 77, No. 11, pp. 1684–1694, 1989.

6. "Reception of automatic dependent surveillance broadcast via satellite and compatibility studies with incumbent systems in the frequency band 1 087.7-1 092.3 MHz", International Telecommunication Union, 2017.

7. Sun, J., *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*, Delft: TU Delft OPEN Publishing, 2021.

8. Wen, M. and C. Zhang, "Algorithm Design of Mode S Downlink Signal Processing Based on 20MHZ Sampling Rate", *Proceedings of IEEE CIE International Conference on Radar*, pp. 1749-1751, Chengdu, China, 2011.

9. David, H., "Conflict-Free Direct Routings in European Airspace", *EEC Report*, No. 308, 1997.

10. Standfuß, T. and M. Schultz, "Performance Assessment of European Air Naviga-

tion Service Providers", *IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pp. 1-10, London, UK, 2018.

11. "Annual Report 2016 Driving Excellence in ATM Performance", European Organisation for the Safety of Air Navigation, Brussels, Belgium, 2017.

12. "Five-Year Forecast 2020-2024 European Flight Movements and Service Units Three Scenarios for Recovery from COVID-19", European Organisation for the Safety of Air Navigation, Brussels, Belgium, 2020.

13. "Data Snapshot #8 on the costs of air traffic management in Europe", European Organisation for the Safety of Air Navigation, Brussels, Belgium, 2021.

14. Barrado, C., M. Boyero, L. Brucculeri, G. Ferrara, A. Hately, P. Hullah, D. Martin-Marrero, E. Pastor, A. P. Rushton and A. Volkert, "U-Space Concept of Operations: A Key Enabler for Opening Airspace to Emerging Low-Altitude Operations", *Aerospace*, Vol. 7, No. 3:24, pp. 1-18, 2020.

15. Haessig, D. A., R. T. Ogan and M. Olive, ""Sense and Avoid" - What's required for aircraft safety", *IEEE SoutheastCon*, pp. 1-8, Norfolk, VA, USA, 2016.

16. Minucci, F., E. Vinogradov and S. Pollin, "Avoiding Collisions at Any (Low) Cost: ADS-B Like Position Broadcast for UAVs", *IEEE Access*, vol. 8, pp. 121843-121857, 2020.

17. Jonáš, P., M. Jančík, Š. Holoda and J. Bodart, "Impact of SUAS Equipped with ADS-B on 1090MHz Environment", *New Trends in Civil Aviation (NTCA)*, pp. 63-67, 2020.

18. Batuwangala, E., T. Kistan, A. Gardi and R. Sabatini, "Certification challenges for next-generation avionics and air traffic management systems", *IEEE Aerospace and Electronic Systems Magazine*, Vol. 33, No. 9, pp. 44-53, 2018.

19. Doole, M., J. Ellerbroek and J.M. Hoekstra, "Investigation of Merge Assist Policies to Improve Safety of Drone Traffic in a Constrained UrbanAirspace", *Aerospace*, Vol. 9, No. 3:120, 2022.

20. Kellermann, R., T. Biehle, and L. Fischer, "Drones for parcel and passenger transportation: A literature review", *Transportation Research Interdisciplinary Perspectives*, Vol. 4, pp. 100088, 2020.

21. Shrestha, R., R. Bajracharya and S. Kim, "6G Enabled Unmanned Aerial Vehicle Traffic Management: A Perspective", *IEEE Access*, Vol. 9, pp. 91119-91136, 2021.

22. Yoo, L.-S., J. H. Lee, S. H. Ko, S. K. Jung, S. H. Lee and Y. K. Lee, "A Drone Fitted With a Magnetometer Detects Landmines", *IEEE Geoscience and Remote Sensing Letters*, Vol. 17, No. 12, pp. 2035-2039, 2020.

23. Jan, S.-U. and H. U. Khan, "Identity and Aggregate Signature-Based Authentication Protocol for IoD Deployment Military Drone", *IEEE Access*, Vol. 9, pp. 130247-130263, 2021.

24. Bria, O., J. Giacomantone and H. V. Wanza, "Interleaving Scheduling Algorithm for SLM Transactions in Mode S Surveillance Radar", *XXIV Congreso Argentino de Ciencias de la Computación*, pp. 948-956, 2018.

25. Chang, E., R. Hu, D. Lai, R. Li, Q. Scott and T. Tyan, *The Story of Mode S: An Air Traffic Control Data Link Technology*, M.I.T, 2000.

26. Stevens, M. C., *Secondary surveillance radar*, Artech House Radar Library, Boston, 1988.

27. Orlando, V.A., "The Mode S Beacon Radar System", *The Lincoln Laboratory Journal*, Vol. 2, No. 3, pp. 345-362, 1989.

28. Sun, J., H. Vû, J. Ellerbroek and J. M. Hoekstra, "pyModeS: Decoding Mode-S

Surveillance Data for Open Air Transportation Research", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21, No. 7, pp. 2777-2786, July 2020.

29. da Silva, B.A.C., P.P. Porto and R. Eller, "Aeronautical Surveillance systems: Historical and future perspectives", *Journal of the Brazilian Air Transportation Research Society*, Vol. 9, No. 1, pp. 9-10, 2013.

30. Gertz, J.-L., *Fundamentals of Mode S Parity Coding*, Lincoln Laboratory, M.I.T, Lexington, 1984.

31. Mandel, T. and J. Mache, "Investigating CRC Polynomials that Correct Burst Errors", *Proceedings of the International Conference on Wireless Networks* , pp. 632-637, 2009.

32. Wikipedia, "Burst error-correcting code", https://en.wikipedia.org/wiki/Burst_error-correcting_code, accessed on June 25, 2022.

33. M.I.T, 2000, "Error Correction Design", http://web.mit.edu/6.933/www/Fall2000/mode-s/ecc.html, accessed on January 4, 2023.

34. Neindre, F.-L., D. Dallet, G. Ferré, E. Boulanger and F. Letellier, "An Enhanced Random Error CRC Correction for ADS-B Squitters Receivers", *27th Telecommunications Forum (TELFOR)*, pp. 1-4, 2019.

35. Shi-yi, C. and L. Yu-bai, "Error Correcting Cyclic Redundancy Checks based on Confidence Declaration", *6th International Conference on ITS Telecommunications*, pp. 511-514, Chengdu, China, 2006.

36. Koga, T., "Classification of Mode S transponders by datalink capability", *Integrated Communications, Navigation and Surveillance Conference (ICNS) Conference Proceedings*, pp. O3-1-O3-7, 2014.

37. Koga, T., and K. Mori, "Autonomous Lockout Map Construction Technique for

Secondary Surveillance Radar Mode S Network", *IEEE Radar Conference*, pp. 1439-1443, Arlington, VA, USA, 2010.

38. "Principles of Mode S Operation and Interrogator Codes Ver:2.3", European Organisation for the Safety of Air Navigation, Brussels, Belgium, 2003.

39. Jiang G., Y. Fan and H. Yuan, "Assessing the Capacity of Air Traffic Control Secondary Surveillance Radar System", *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC)*, pp. 1-3, Taiyuan, China, 2019.

40. Mazúch, P. and R. Klecun, "Identification of an Aircraft with the Mode-S", *Acta Avionica*, Vol. 15, No. 27, pp. 1-6, 2013.

41. Schafer, M., M. Strohmeier, M. Smith, M. Fuchs, R. Pinheiro, V. Lenders and I. Martinovic, "OpenSky report 2016: Facts and figures on SSR mode S and ADS-B usage", *IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pp. 1-9, Sacramento, CA, USA, 2016.

APPENDIX A: LIST OF RELATED PUBLICATIONS

During the thesis study, a paper named "Mode-S Radar Interrogation Algorithm Design for Dense Air Traffic Environment" was published in the Radioengineering Journal (2022, vol. 31, no. 4, p. 460-467) and the authors were Ahmet Öncü, Ahmet Gunhan Aydin, Yunus Emre Erdoğan, and Artun Akdoğan. The images produced within the scope of this thesis work and whose copyrights were transferred to the publisher were used in the thesis in accordance with the publishing policy applicable to the reuse of the text and graphics produced by the author on the website of the publisher.