A REPORT

ON

GUI for spatio-temporal mapping of low frequency ambient noise in the IOR

BY

Ayush Kumar

AT

Maritime Research Centre
Deccan College, Yerawada, Pune

(July, 2019)

Acknowledgement

I would like to express my very great appreciation to Dr (Cdr) Arnab Das, Director, for providing me the opportunity to do this project at Maritime Research Centre. I would also like to thank him for his valuable and constructive suggestions during the planning and development of this research work.

I am really thankful to Mr. Sridhar Prabhuraman for his willingness to give his time so generously. He helped me a lot in finalizing this project within the limited time frame.

Abstract

The Indian Ocean Region (IOR) today is arguably the most strategically important sea area globally. Being the focal point of maritime trade and a Geo-political hot-spot, the IOR is the transit point of maximum number of military and merchant fleets. The high volume of maritime traffic also translates to high levels of low frequency anthropogenic noise that are primary cause of acoustic habitat degradation for the big whales. The frequent stranding of such species in the recent past is a testimony of high ambient noise in the region. The quantitative and qualitative assessment of the acoustic habitat degradation is a major step in any conservation initiative. In this work, my aim is to perform mapping of low frequency ambient noise over a selected Indian Ocean Region.

Table of contents

Introduction

The graphical user interface or GUI is a form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard.

A major advantage of GUIs is that they make computer operation more intuitive, and thus easier to learn and use. GUIs generally provide users with immediate, visual feedback about the effect of each action. GUI allows multiple programs and/or instances to be displayed simultaneously. This feature can used to design a user-friendly GUI for spatio-temporal mapping of low frequency ambient noise in the Indian oceanic region.

The interaction of the wind with the ocean surface has long been recognized as a major source of acoustic noise. Measurements of the omnidirectional noise at the higher frequencies have been found to exhibit wind-dependent characteristics; and, when not dominated by shipping noise, the most likely mechanisms are related to bubbles, spray, and splashes associated with white caps, as well as capillary wave/wave interactions.

Since the Indian oceanic region has become one of the most important strategically important seas globally mapping the ambient noise in the region becomes necessary both for strategic and marine conservation purpose.

Main Text

1. History: A visual language has evolved as GUI has become commonplace in both operating systems (OS) and software applications. Even those with few computer skills can now, through the use of GUI, learn how to use computer applications for word processing, finances, inventory, design, artwork or hobbies.

1.1 Xerox Alto: Alan Kay was born on 17th May 1940 and is now known as the man who developed object oriented programming. At the beginning of his career he predicted that one day soon we would have inexpensive personal computers with a phenomenal amount of power and high quality graphics. After Kay finished his PhD in Graphical Object Orientation, he did some work on Programming Language Design at the Stanford Artificial Intelligence Laboratory. He then joined Xerox PARC as a researcher in 1971 and with the help of Dan Ingalls designed a programming language called Smalltalk where individual cells communicate with each other. Kay also came up with the idea of the Dynabook, a laptop computer for children. Together with Chuck Thacker they developed a prototype Dynabook which they named the Alto, the first personal networked computer. The Xerox Alto is the first computer designed from its inception to support an operating system based on a graphical user interface(GUI), later using the desktop metaphor. The first machines were introduced on 1 March 1973, a decade before mass market GUI machines became available.

1.2 Lisa : One of the things Xerox showed Jobs was the Alto, which sported a GUI and a three-button mouse. When Jobs saw this prototype, he had an epiphany and set out to bring the GUI to the public. Apple engineers developed Lisa, the first GUI-based computer available to the public. It was too expensive; no one bought it. But the seed germinated into a flower that would change the world.

1.3 Macintosh : Released in 1984 and billed as "insanely great," the Macintosh caught the public eye with one of the most famous commercials ever. This immortal television advertisement depicted users of IBM's PC as Orwellian drones trapped in the maw of a monochromatic, brutally mechanical, command-line interface, and dramatized their symbolic liberation by a woman bearing a new tool for home computations

2.Working of GUI

2.1 AIS : The **Automatic Identification system** or AIS has been developed with the primary aim of collision avoidance for water transport. It uses both land and satellite based sensors to receive and further transmit the unique identification, position, course and speed of the vehicle thus supplementing marine radar for collision avoidance. AIS since its development has gathered many uses including collision avoidance, fishing fleet management and control, maritime security, aids to navigation, search and rescue, accident investigation, ocean current estimates, infrastructure protection and fleet and cargo tracking. With such widespread use, AIS has become the most important tool for maritime domain awareness.

Although sound emanates in the ocean from a myriad of sources, shipping has become a dominant source of low frequency ambient noise in the ocean and thus mapping the noise created by the ships becomes important both for defence and marine conservation purposes. **Don Ross**, a civilian submariner and acoustics expert is credited with discovering that low-frequency ocean ambient noise is largely determined by shipping, and that ocean ambient noise has been steadily increasing due to human activities. He also made the first detailed measurements of the noise characteristics of nuclear submarine.

Over the years, the shipping noise estimation techniques as well as the applications have evolved quite a bit with advancement in technology and now have relevance to multiple military and non-military applications across multiple stakeholders.

2.2 QGIS: Q- Geographical Information System is open source software which will be used to map the ambient noise from shipping using the AIS data provided. This software can be called up by the GUI and real time mapping will be reflected in the GUI.

2.3 GUI: The graphic user interface will be designed in python using Py-QT. The GUI will call up the QGIS and will provide real time data of the AIS for mapping. This GUI will have a standard and user-friendly interface.

The major benefit of a GUI is that systems using one are accessible to people of all levels of knowledge, from an absolute beginner to an advanced developer or other tech-savvy individuals

3. Flow of the work

3.1 AIS Decoding: AIS data received and transmitted from the ships is in decoded form, so the first task is to decode AIS data.


```
!BSVDM,2,1,0,A,544hWO0256J9D<U`001aHETpTDe<00000000000N1h`745S`0;0PDSClQKP0,0
*2D,04/17/2018 10:52:24 AM
!BSVDM,2,2,0,A,00000000002,0*3D,04/17/2018 10:52:24 AM
!AIVDM,1,1,,,144hWO0P0TSB;1>`fTJLEgwV2d03,0*05,04/17/2018 9:14:54 AM
!BSVDM,2,1,0,A,545=>iP20R6mDLiP000dU<PTpEHd40000000000N1@b275B<09l1E4Sl0ES3,0
*30,04/17/2018 10:52:24 AM
!BSVDM,2,2,0,A,mTjh0000002,0*06,04/17/2018 10:52:24 AM
!AIVDM,1,1,,,345=>iP009;5NGfOqsuf4:oV01u0,0*1A,04/17/2018 9:14:54 AM
!AIVDM,1,1,,,15S<?B01B9<:bi`NMqtaR7e`00S6,0*04,04/17/2018 9:14:55 AM
!AIVDM,1,1,,,344iBn0P@@;M89pNfDB8Q6sd1Pu@,0*44,04/17/2018 9:14:57 AM
!AIVDM,1,1,,,145:G90P@j;:l@TQL2r;Gpad0`Pw,0*03,04/17/2018 9:14:59 AM
!BSVDM,2,1,0,A,544cme02?N2MD9DGT01@t8uT000000000000001S4@h8:4kaNG5PDPCQ1F
@0,0*77,04/17/2018 10:52:24 AM
!BSVDM,2,2,0,A,00000000002,0*3D,04/17/2018 10:52:24 AM
!AIVDM,1,1,,,144cme00014:8UfWPi1MW0Uj0KvR,0*2C,04/17/2018 9:14:59 AM
!AIVDM,1,1,,,145=>iPOh9;5NGlOqsw>4:ol0<00,0*1D,04/17/2018 9:15:01 AM
```


Fig 1: Sample AIS data

The encoded data contains information related to Ship static and voyage related data, AIS Version, Call Sign, Vessel Name, Ship Type, Dimension to Bow, Stern, Port & Starboard (all in m), ETA at destination (MMDDHHMM), Destination. The decoding is done based on type of AIS data (class 1 and class 5 data were decoded by me). Sometimes data is so long that data of a single ship is sent through 2-3 packets which are identified using numbers just after the class identifier.

The main decoded part after these are converted to ASCII characters and message is decoded using the given sets of rules. A python script (given in Appendix B) was created to decode the huge amount of AIS data and subsequently saving it into an excel file.

MMSI number, ship's name, latitude, longitude, speed, bow, stern, port, starboard and draught were obtained.

	A	B	C	D	E	F	G	H	I	J	K
1	MMSI	NAME	LAT	LON	SPEED	BOW	STERN	PORT	STARBOARD	DRAUGHT	
2	273426300	ZVEYNIEKS@@@@@@@@@@@	71.17671	45.91281	36	14	40	7	2	44	
3	273895110	KISHINEVKA@@@@@@@@@@				10	42	2	3	39	
4	372445000		53.24542	170.1027	137						
5	273348020	TOBOY@@@@@@@@@@@@@@@	69.05217	58.13791	1	34	48	8	5	92	
6	273895110		55.7585	154.9871	9						
7	209443000		48.25505	158.1131	79						
8	273390940	NORD" "WEST@@@@@@@@@@@				17	30	6	3	64	
9	273314360	DUDINKA@@@@@@@@@@@@@				23	61	11	5	66	
10	273841220	STERLYAD@@@@@@@@@@@@				15	39	3	3	43	
11	273841220		57.19157	154.5564	100						
12	273316240		71.38045	72.41757	0						
13	273381830		59.56335	150.7354	1023						
14	369283000		60.54381	268.8669	19						
15	273312530		68.50069	73.66964	1						
16	273363870		62.64928	135.4753	0						

Fig 2: Decoded AIS Data

3.2 Mapping of AIS data: The decoded AIS data was mapped in the Indian Oceanic Region using Q- Geographical Information System, an open-source software. This will help us get a rough idea of low frequency ambient noise created due to shipping. Using the channel modelling, precise information can be generated. Overlapping the AIS data with the satellite image we got a good overlap of the marine traffic and thus the ambient noise.

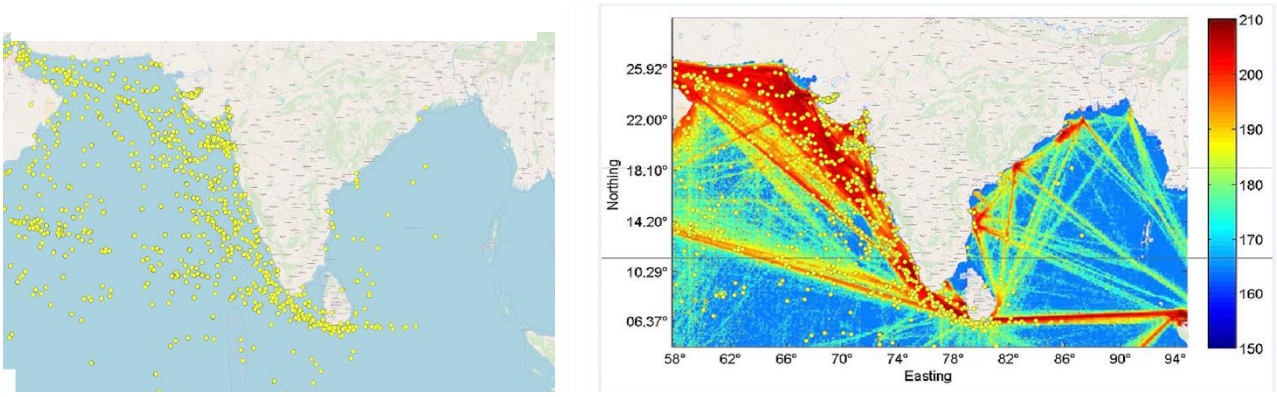


Fig 3. AIS data plotted in the IOR and then overlapped with actual noise map

3.3 The Graphic User Interface: Finally the graphic user interface was designed using PyQt in QT designer.

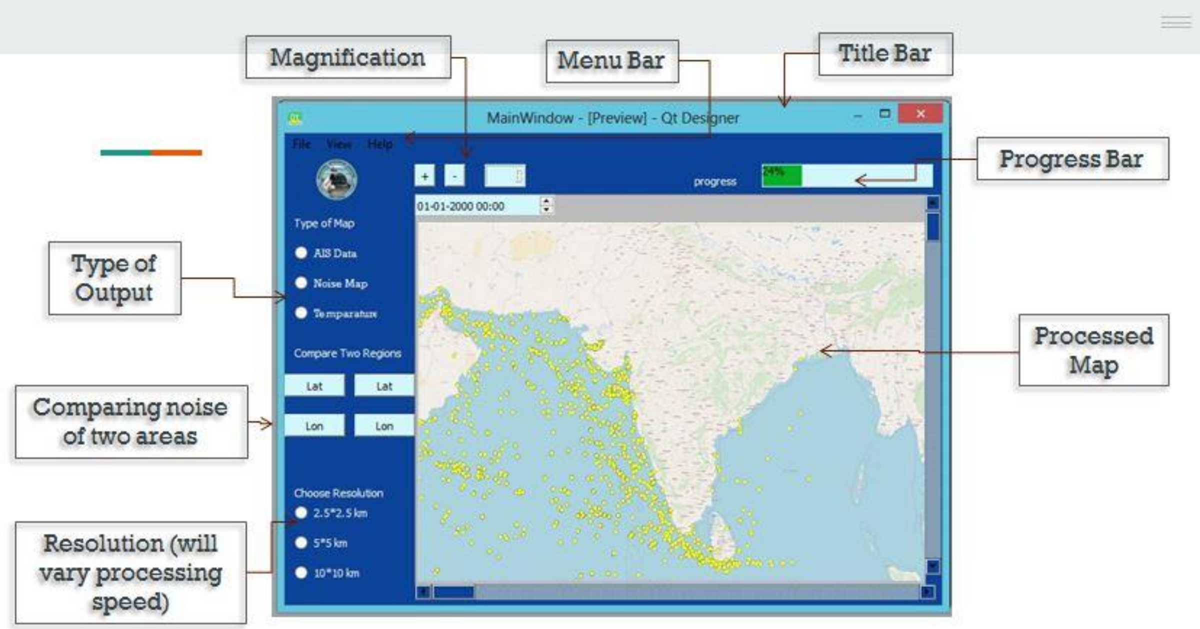


Fig 4 Designed GUI with its Features

The GUI contains Menu bar, Title bar, radio buttons to choose type of output and resolution and feature of comparing two strategically important areas. Progress bar for loading of map and last update time is also available. Magnification can also be adjusted with dedicated buttons.

Choice of resolution is there so as processing time could be managed. If you choose to view a small area then you can view at a higher resolution but if you see a larger area then choose the smaller resolution so as to manage the processing time.

4. Future Scope of GUI

Some of the special features that the GUI can contain are:

• The software will be able to map both manually entered data as well as data taken directly from the internet.

• It will also be able to show 3D profile of noise in a selected area

• This software will also have option for showing temperature gradient, ocean floor contour plot and type of ocean bottom.

Conclusion

The developed GUI can be used for:

Navy-submarine Deployment:

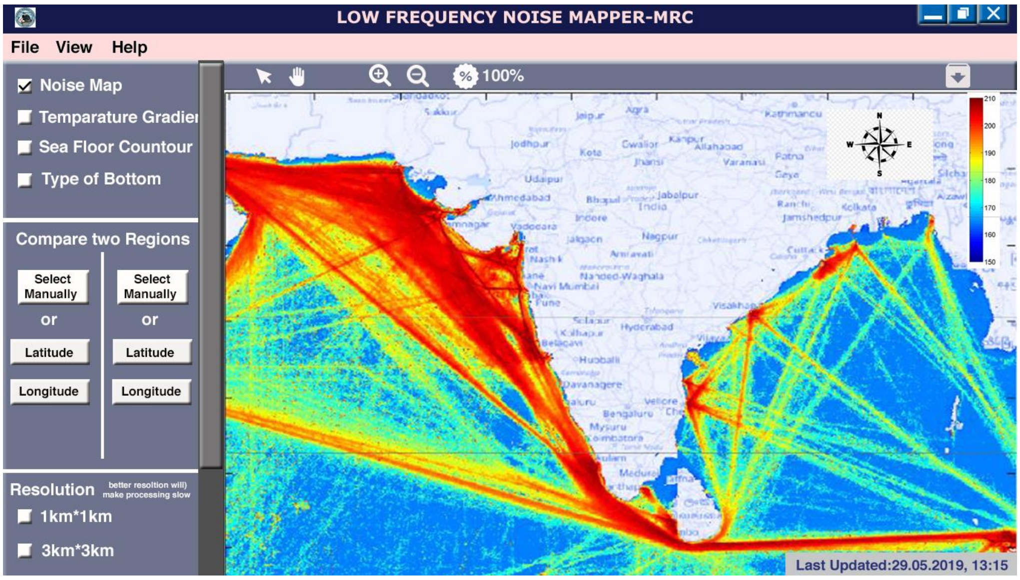
Sonar (originally an acronym for sound navigation ranging) is a technique that uses sound propagation (usually underwater, as in submarine navigation) to navigate, communicate with or detect objects on or under the surface of the water, such as other vessels. Submarine should be deployed as such that there is low anthropogenic noise in the surrounding so as the use of SONARs should be judicious.

Maritime Conservation:

Anthropogenic noise is considered an acoustic pollutant, with an anticipated increase through the expansion of shipping, resource extraction and offshore development. Impacts of acoustic pollution on marine biota range from death due to physical injury and auditory damage, to behavioural and habitat-use changes. Marine animals rely on sound for navigation, feeding and communication and are known to be particularly sensitive to anthropogenic noise, with effects being detected over tens of kilometres from sources. Effects depend on various factors, including overlap in space and time with the organism and sound source, duration. Thus study of such mapping will help to decide marine animal's conservation sites.

Appendix

A - Sample GUI



B- AIS Decoder Python Script

Main.py

```
# Information

# xN[0] = NMEA Identifier of Sentence N
# xN[1] = Fragment Count of Sentence N
# xN[2] = Current Fragment Number Sentence N
# xN[3] = Message ID of Sentence N
# xN[5] = Data Payload of Sentence N
#
====================================================================================
================================= #
# imports
import sys
import os
import payloadDecoder
import time
import openpyxl

#
====================================================================================
================================= #
# Global Values

# Making sure the output file is empty everytime before the module is run.
# copyfile('documents/format_ais_compiled.xlsx', 'documents/ais_compiled.xlsx')
#
====================================================================================
================================= #
# Main Functions

def cleanAISData():
    line_no = 1

    f = open("documents/ais_processed.txt", encoding='utf-8')  # opened in read
mode
    f_length = file_len('documents/ais_processed.txt')  # returns length of the
```

```
file
    f1 = open("documents/ais_processed_cleaned.txt", mode="w", encoding='utf-8')  #
opened in write mode
    f1_length = file_len('documents/ais_processed.txt')

    # remove any sentences having missing identifiers
    while (line_no <= f_length):
        sentence = f.readline()
        x = sentence.split(',')  # splitting the sentence
        if (x[0] == "!AIVDM" or x[0] == "!BSVDM"):
            print(x[0])
            print("reached line : ", line_no)
            f1.write(sentence)
        line_no = line_no + 1

    line_no = 1
    f.close()
    f1.close() # close f1 which was opened in write mode

#
===================================================================================
================================= #
# Extra Functions
def file_len(fname):
    with open(fname) as f:
        for i, l in enumerate(f):
            pass
    return i + 1

def generateError(error):
    print("Some Error Occured : ", error)
    sys.exit()
#
===================================================================================
================================= #
#
===================================================================================
================================= #

def run():

    # clean the ais file. Remove/Rectify incomplete lines
    cleanAISData() #todo: uncomment post testing

    # obtain output
    sr_no = 1 # serial number of the sentence. Used to store data in excel sheet
setially

    f = open("documents/ais_processed_cleaned.txt", encoding='utf-8')  # opening a
file
    f_length = file_len('documents/ais_processed_cleaned.txt')  # length of the
file viz total statements count
    print("file length : ", f_length)

    for j in range(1, 325800):# todo: change to f_length+1 post testing
        sentence = f.readline()
        sentence_split = sentence.split(',')
        fragment_count = int(sentence_split[1])  # total number of fragments in the
current sentence
        # sentence_identifier = sentence_split[0]
        sub_sentence = [0] * fragment_count
        sub_sentence_split = [0] * fragment_count

        # take note : eg when fragment count = 2, sub_sentence & ..split will be a
list with 0 & 1 positions
        print("\n")
        print(sr_no,"}","Message :", sentence_split[3], "Sentence Number",
sentence_split[2], "/", sentence_split[1])
```

```
        if(fragment_count>1):
            for i in range(1, fragment_count):
                sub_sentence[i] = f.readline()  # Reading a sentence
                j+=1 #incrementing J so that we do not unnecessarily re-read the
sentence
                sub_sentence_split[i] = sub_sentence[i].split(',')  # Splitting the
sentence
                # check if msg id matches
                if sub_sentence_split[i][3] == sentence_split[3]:
                # if identifiers match, join the payload
                    if sub_sentence_split[i][0] == sentence_split[0]:
                        sentence_split[5] = sentence_split[5] +
sub_sentence_split[i][5]
                        payload = sentence_split[5]
                    else:
                        payload = sub_sentence_split[i][5]
                else:
                    print("Error! Message ID does not match")

            print("payload :", payload)
            sr_no+=1 # sr+no ++
            payloadDecoder.run(payload,sr_no)

        else:
            # fragment count <=1
            # check current fragment number. It should be 1
            if(int(sentence_split[2]) == 1):
                payload = sentence_split[5]
                print("payload :", payload)
                sr_no += 1   # sr+no ++
                payloadDecoder.run(payload,sr_no)

            else:
                print("Fragment Count Error")

    f.close()
    payloadDecoder.wb.save('documents/ais_compiled.xlsx')


if __name__ == '__main__':
    start_time = time.time()
    run()
    # lastly club the data of multiple redundant lines
    wb = openpyxl.load_workbook('documents/ais_compiled.xlsx')
    sheet = wb['Sheet1']

    maxRow = sheet.max_row
    maxCol = sheet.max_column

    for i in range(2,10):
        this_MMSI = sheet.cell(row=i,column=1).value
        print(this_MMSI)
        next_MMSI = sheet.cell(row=i+1,column=1).value
        print(next_MMSI)
        if(this_MMSI == next_MMSI):
            print("in row :",i)
            # They both are parts of same ship's AIS.We need to club them.
            # Acquire values of the next row
            next_lat = sheet.cell(row=i+1,column=3).value
            next_long = sheet.cell(row=i + 1, column=4).value
            next_speed = sheet.cell(row=i + 1, column=5).value
            # Insert them into the current row
            sheet.cell(row=i, column=3).value = next_lat
            sheet.cell(row=i, column=4).value = next_long
            sheet.cell(row=i, column=5).value = next_speed
            # delete next row
```

```
            sheet.delete_rows(i+1,1)
            row_deleted = True
        else:
            # Do nothing and skip to next row
            row_deleted = False
    wb.save('documents/ais_compiled.xlsx')
    print("--- %s seconds ---" % (time.time() - start_time))
```

PayloadDecoder.py

```
import openpyxl
from shutil import copyfile

ais_ascii = ['@','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O'
            ,'P','Q','R','S','T','U','V','W','X','Y','Z','[','\\',']','^'
            ,'_','" "','!','"','#','$','%','&',"'",'(',')','*','+',',','-'
            ,'.','/','0','1','2','3','4','5','6','7','8','9',':',';','<','='
            ,'>','?']


# creating ais_complied.xlsx
copyfile('documents/format_ais_compiled.xlsx', 'documents/ais_compiled.xlsx')
# opening ais_compiled.xlsx
wb = openpyxl.load_workbook('documents/ais_compiled.xlsx')
sheet = wb['Sheet1']



# Payload armoring
def removePayloadArmoring(payload):

    total_value = "" # empty string

    for each_character in payload:
        value = ord(each_character)             # value = ASCII
        value = value - 48                      # subtract 48
        if value> 40:                           # after subtraction , if > 40
            value = value - 8                   # subtract 8
        value = str(decimalToBinary(value,6))   # converting (decimal to 6 bit
binary of value) to string
        total_value = total_value + value       # concatenation of 6 bit bin values
of every character in payload

    print("Provided Payload :",payload)
    print("Decrypted Value :",total_value,"\n")
    return total_value

# Interpreting the payload after armoring is removed
def interpretPayload(clean_payload,sr_no):

    # First check the type of message
    message_type = clean_payload[0:6]                               # 0-5 field in
payload is Message Type
    message_type = binaryToDecimal(message_type)                    # Binary to Decimal
Conversion
    print("Message Type :", message_type)

    # Here we are considering only two types of messages : 1 & 5
    if message_type == 1:

        repeat_indicator = clean_payload[6:8]
        repeat_indicator = binaryToDecimal(repeat_indicator)
        print("Repeat Indicator Value :",repeat_indicator)
        if (repeat_indicator == 0 or repeat_indicator == 7):
            print("      >  Vessel is performing some activity. Noise will be
considered")
```

```
        else:
            print("       >  Vessel is not active. Complete noise will not be
considered")

        mmsi = clean_payload[8:38]
        mmsi = binaryToDecimal(mmsi)
        print("MMSI :",mmsi)
        sheet.cell(row = sr_no,column = 1).value = mmsi

        speed = clean_payload[50:60]
        speed = binaryToDecimal(speed)
        print("Speed :",speed, "Kts")
        sheet.cell(row=sr_no, column=5).value = speed

        longitude =  clean_payload[61:89]
        longitude = binaryToDecimal(longitude)
        longitude = longitude/600000.0
        print("longitude :",longitude)
        sheet.cell(row=sr_no, column=4).value = longitude

        latitude = clean_payload[89:116]
        latitude = binaryToDecimal(latitude)
        latitude = latitude / 600000.0
        print("latitude :", latitude)
        sheet.cell(row=sr_no, column=3).value = latitude

    elif message_type == 5:

        mmsi = clean_payload[8:38]  # 8-37 field is MMSI
        mmsi = binaryToDecimal(mmsi)
        print("MMSI :", mmsi)
        sheet.cell(row=sr_no, column=1).value = mmsi

        vessel_name = clean_payload[112:232]
        full_name = ""
        for i in range(0, len(vessel_name), 6):
            name = vessel_name[i:i + 6]
            name = binaryToDecimal(name)
            name = decimalToAisAscii(name)
            full_name = full_name + name
        vessel_name = full_name
        print("Vessel Name :",full_name)
        sheet.cell(row=sr_no, column=2).value = full_name

        bow = clean_payload[240:249]
        bow = binaryToDecimal(bow)
        print("Bow :", bow)
        sheet.cell(row=sr_no, column=6).value = bow

        stern = clean_payload[249:258]
        stern = binaryToDecimal(stern)
        print("stern :", stern)
        sheet.cell(row=sr_no, column=7).value = stern

        port = clean_payload[258:264]
        port = binaryToDecimal(port)
        print("Port :",port)
        sheet.cell(row=sr_no, column=8).value = port

        starboard = clean_payload[264:269]
        starboard = binaryToDecimal(starboard)
        print("Starboard:",starboard)
        sheet.cell(row=sr_no, column=9).value = starboard

        draught = clean_payload[294:302]
        draught = binaryToDecimal(draught)
        print("Draught:", draught)
        sheet.cell(row=sr_no, column=10).value = draught
```

```
    else:
        print("Unknown Message Type. Only Type 1 and Type 5 Messages are
supported")

    # wb.save('documents/ais_compiled.xlsx')


def decimalToBinary(n,fill):
    return bin(n).replace("0b","").zfill(fill)

def binaryToDecimal(n):
    return int(n,2)

def decimalToAisAscii(name):
    return (ais_ascii[name])

def run(payload,sr_no):
    clean_payload = removePayloadArmoring(payload)
    interpretPayload(clean_payload,sr_no)

# run("144bb50000VaKvpP3<Wf47EL2@K>",1)
```

References

[1] (n.d.). Retrieved June 03, 2019, from https://en.wikipedia.org/wiki/Graphical_user_interface

[2] What is a Graphical User Interface (GUI)? - Definition from Techopedia. (n.d.). Retrieved June 3, 2019, from https://www.techopedia.com/definition/5435/graphical-user-interface-gui

[3] MP3 Audio of Ron Cude talking about the 1979 Boca Raton Alto Event. The DigiBarn Computer Museum, 2003.

[4] Xerox Alto. (n.d.). Retrieved June 3, 2019, from https://history-computer.com/ModernComputer/Personal/Alto.html

[5] Powell, A. (2017, December 15). Web 101: A History of the GUI. Retrieved June 3, 2019, from https://www.wired.com/1997/12/web-101-a-history-of-the-gui/

[6] Booth, C. (2011, July 13). Alan Kay and the Graphical User Interface. Retrieved June 3, 2019, from https://web.archive.org/web/20110713232927/http:/www.lottiebooth.com/pdf/essay.pdf

[7] J.Smith.JJJ, J.Smith.JJJJ.Smith.JJJ 172129, Jørn E. AngeltveitJørn E. Angeltveit 11.7k33578, Benny SkogbergBenny Skogberg 46.2k20124231, Bchettybchetty 20913, PeterisPeteris 1, . . . User1444692user1444692 91. (n.d.). Why are graphical user interfaces considered user-friendly? Retrieved June 3, 2019, from https://ux.stackexchange.com/questions/52372/why-are-graphical-user-interfaces-considered-user-friendly/52376

[8] What is a graphical user interface? (1970, January 28). Retrieved from https://www.itpro.co.uk/operating-systems/30248/what-is-a-graphical-user-interface

[9] Automatic identification system. (2019, May 30). Retrieved June 3, 2019, from https://en.wikipedia.org/wiki/Automatic_identification_system

[10] Putland, R. L., Constantine, R., & Radford, C. A. (2017, July 18). Exploring spatial and temporal trends in the soundscape of an ecologically significant embayment. Retrieved June 3, 2019, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5516011/

[11] Don Ross (acoustician). (2018, December 12). Retrieved June 3, 2019, from https://en.wikipedia.org/wiki/Don_Ross_(acoustician)

[12] D. Ross, "Ship sources of ambient noise," in *IEEE Journal of Oceanic Engineering*, vol. 30, no. 2, pp. 257-261, April 2005. doi: 10.1109/JOE.2005.850879

[13] MAPPING AMBIENT NOISE FOR BIAS - BIAS | Baltic Sea ... (n.d.). Retrieved June 3, 2019, from https://biasproject.files.wordpress.com/2017/01/qo-20130203-01-rap-001-01b-foi-bias-modelingreport.pdf

[14] Spatial Variation of Ambient Noise in Indian Ocean Region. (n.d.). Retrieved June 3, 2019, from https://shodhganga.inflibnet.ac.in/bitstream/10603/209097/15/15_chapter 6.pdf

[15] Sonar. (2019, June 02). Retrieved June 3, 2019, from https://en.wikipedia.org/wiki/Sonar

[16] Sutton, Gerry & Jessopp, Mark & Folegot, Thomas & Clorennec, Dominique. (2014). Mapping the Spatio-temporal Distribution of Underwater Noise in Irish Waters. 10.13140/RG.2.1.1673.1289. – marine conservation

Glossary

GUI- Graphic User Interface

AIS- Automatic Information System

QGIS- Quantum Geographical Information System