International Association of Maritime Universities

IAMU 2021 Research Project
(No. YAS202102)

Calibrating ship emissions with AIS data and field measurements

Theme:(Future opportunities and challenges of the sustainability of maritime industry)

By

Barcelona School of Nautical Studies -
Polytechnical University of Catalonia (FNB-UPC)

August 2022

IAMU
International Association of Maritime Universities

This report is published as part of the 2021 Research Project in the 2021 Capacity Building Project of International Association of Maritime Universities, which is fully supported by The Nippon Foundation.

The text of the paper in this volume was set by the author. Only minor corrections to the text pertaining to style and/or formatting may have been carried out by the editors.

Editorial

IAMU Academic Affairs Committee (AAC)

Head of Committee : Professor Dr. Adam Weintrit
Rector, Gdynia Maritime University (GMU)

Editorial committee : Adam Przybylowski (GMU)
Avtandil Gegenava (BSMA)
Christian Matthews (LJMU)

Contractor : Daniel Crespo Artiaga

Research Coordinator : Anna Mujal Colilles

Published by the International Association of Maritime Universities (IAMU) Secretariat
Meiwa Building 8F, 1-15-10 Toranomon, Minato-ku,
Tokyo 105-0001, JAPAN
TEL : 81-3-6257-1812 E-mail : info@iamu-edu.org URL : http://www.iamu-edu.org

ISBN978-4-907408-40-4

IAMU 2021 Research Project
(No. YAS202102)

Calibrating ship emissions with AIS data and field measurements

Theme:(Future opportunities and challenges of the sustainability of maritime industry)

By
Barcelona School of Nautical Studies -
Polytechnical University of Catalonia (FNB-UPC)

Contractor : Daniel Crespo Artiaga
Research Coordinator : Anna Mujal Colilles

Calibrating ship emissions with AIS data and field measurements

Theme:

(Future opportunities and challenges of the sustainability of maritime industry)

Facultat de Nàutica de Barcelona, UPC-BarcelonaTECH

And

Research Coordinator

Anna Mujal Colilles

Serra Húnter Lecturer, Facultat de Nàutica de Barcelona, anna.mujal@uc.edu

Abstract: The impact of the SARS-CoV pandemic has gone well beyond health concerns, reaching the maritime industry. In this paper, we compare the impact of COVID-19 on maritime traffic and related emissions in port cities. We present a case study centered in the Port of Barcelona covering a 30 nautical miles range in the period March to July 2020, during which different levels of restrictions and stringent lockdown measures were enforced. The study uses real-time Automatic Identification System (AIS) data to assess maritime traffic, and ship emissions are estimated according to the Ship Traffic Emission Assessment Model (STEAM). During lockdown, we observed a 27.9% reduction in the number of port calls compared to the pre-lockdown scenario, which suggested a low to moderate vessel traffic decrease. Our monitoring reveals that vessel traffic decreased by 8.4% in the area during the same period. Interestingly, results show that the decrease in maritime emissions had no correlation with the observed decline in maritime traffic because of changes in vessel operation. Vessels switched from Underway to At Anchor or Moored AIS status, during which auxiliary engines are used at higher revolutions, resulting in higher emissions. Hence, during home quarantine, the smaller number of ships was balanced by the vessel operation mode, and CO_2, SO_2, NOx, and PM emissions rose by 3%, 3.7%, 7.3% and 3.8%, respectively, compared to pre-lockdown levels. This can be attributed to slow-steaming orders given to tankers and cargo ships by shipping companies during the period when oil barrel prices fell to the lowest values since the 2000s. Air quality measurements indicate a significant drop on pollutants concentrations regardless of the small increase on vessel related emissions. The drop can be attributed to wheeled traffic

Keyword*: AIS, maritime pollution, air quality*

Executive summary

This proposal was accepted as part of the YAS-FY2020 and was scheduled to start May 2020 to May 2021. Due to the Covid19 pandemic, it was postponed as agreed by the International Executive Board in May 2020. However, the worldwide pandemic was seen as an opportunity to achieve the main goal and see the total contribution of the maritime traffic to the air pollution in the city of Barcelona.

The Port of Barcelona is one of the main ports in the western Mediterranean Sea, and is an example of a port growth near the city. It combines a cargo-oriented design and an in-port city concept since it is also an important port in cruise calling. According to the annual statistic reports of the Port of Barcelona, last year there was an increase between 9% to 15% of passengers and TEU's respectively, although the total calls only increased 0.7%, which means that larger and more powerful vessels are docking in it. This activity has led to an increase of city pollution which is one of the main concerns of harbor and city authorities as well as citizens since ship emissions was believed to be one of the main contributors to air pollution in Barcelona.

The main goal is to explore correlation between maritime traffic and air quality at the vicinity of a harbor, using Automatic Identification System (AIS) data and real air quality measurements. The steps followed are:

1. Set up the AIS system to automatically record messages and store them properly.
2. Create a workflow to decode and store decoded messages following the international guidelines.
3. Analyze maritime traffic.
4. Compute air pollutant emissions generated by maritime traffic at Port of Barcelona and nearby areas (~30 nm)
5. Obtain air quality measurements.

AIS data was collected through a receiving station located at the Department of Nautical Sciences and Engineering at the Barcelona School of Nautical Studies (UPC-BarcelonaTech). The antenna is located at the roof of the building with a >10 m wire connecting it to the Class B-AIS, SeaTraceR AIS Class B Transponder S.287, at the Informatics Laboratory. A second receiver was put in a different location of the roof with the receiver connected through a 3 m cable. The recording system included the addition of four last digits at the end of each message in order to track the local time stamp of the received message. Raw messages were then decoded using a python algorithm. Only static messages 5 and 24A-B for Class A and Class B AIS transmitters were decoded respectively, along with dynamic messages 1-2-3 and 18 containing trip information. As an order of magnitude, one month of messages was around ~1GB of disk space.

After recording and pre-processing the AIS data following international standards, we estimated the emissions generated by vessels in the area using the STEAM v3 algorithm. Data from engines was obtained from the HIS-Markit database. Air quality was initially compared using data from Barelona city council but an air quality and meteo monitoring system using low-cost tools has been developed.

The covid pandemic provided a very good environment to check the contribution of the maritime traffic to the overall air pollution at the city of Barcelona. Results of the research show a mild decrease of vessels present in the area during the lockdown (less than 10%), whereas there was a drop of ~30% in port calls during the same period. This indicates that the vessels were in the vicinity of the port. As a result, during the same period, fuel consumption dropped only 4% and the correspondent air-pollutant emissions did not change in PM and slightly increased in NOx (1.3%).

Surprisingly, the estimated emissions from the vessels in the vicinity of the port suggest that the level of pollutants did not decrease during the lockdown mainly due to vessels remaining longer in the area and changing their operational mode to moored or at anchor using auxiliary engines that usually produce more NOx's. Also, when traffic contribution of merchant vessel types is compared to their contribution to air pollution in , interesting outcomes suggest that passenger vessels (including regular ferry lines and cruises) are the less environmentally friendly of merchant vessels. This raises the current ongoing discussion about sustainability of passenger business other than ferry crossings

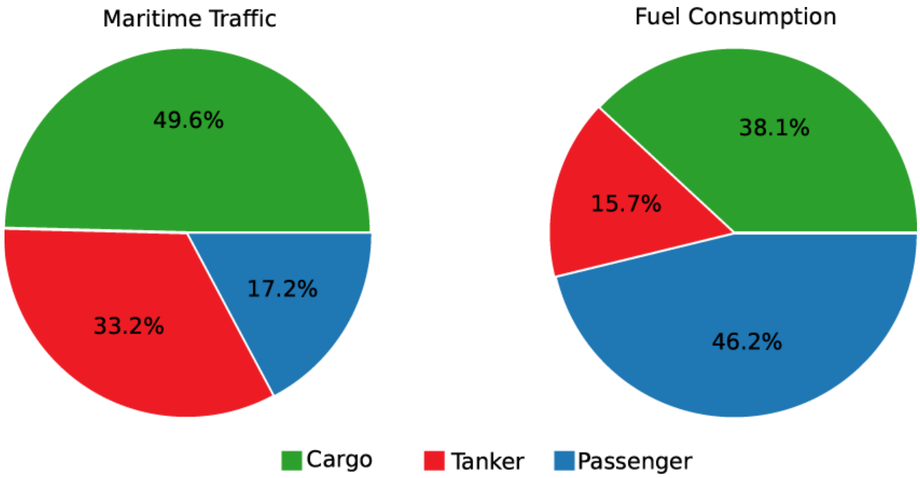


Fig. 1. Average distribution (%) of left) maritime traffic and right) fuel consumption per ship type

In contrast to vessel emissions, air pollution dropped significantly (~30%) at the city of Barcelona during the same period. This result can be linked to the prominent decrease in the wheeled traffic and industrial activity.

It is worth mentioning that the outcomes of the project have provided the students, professors and research staff of the Barcelona School of Nautical Studies and, by extension, the IAMU community, of a protocol to decode and store AIS data that can be further used for future lectures, research project and tools development. Moreover, the development of an air quality low-cost measuring system has also provided the School with a new tool to be replicated by students and record scientific data at the same time.

Acknowledgements

Author would like to acknowledge the Nipon Foundation and the International Association of Maritime Universities for the opportunity to carry out the present project. Author also acknowledges the Serra Húnter Programme of the Generalitat de Catalunya, the Facultat de Nàutica de Barcelona and its Dean Capt. Agustí Martín and sub-dean Dr. Rosa Fernandez.

Special acknowledgement to the colleagues and students that have worked in the framework of this project: Dr. Jordi Fonollosa, Dr. Marcel·la Castells, Dr. Toni Llull, Msc. Javier Nieto, Msc. Oriol Valldosera and Msc Carolina Bagés. The technical staff of the Barcelona School of Nautical Studies Santi Menendez, Diego Campanario and David Segura for their support.

Contents

Table of Figures

List of Tables

1. Introduction

This proposal was accepted as part of the YAS-FY2020 and was scheduled to start May 2020 to May 2021. However, due to the Covid19 pandemic, it was postponed as agreed by the International Executive Board in May 2020. Therefore, the initial results are focused on the data recorded during the pandemic.

The novel Coronavirus disease was first reported symptomatically somewhere between December 1 and December 8, 2019 in the landlocked Chinese city of Wuhan, Hubei. Human-to-human transmission was first confirmed by World Health Organization (WHO) authorities on January 20, 2020 [1]. The rapid spread of the disease across the world led to the global pandemic declaration on March 11, 2020, with subsequent disruptions in world industry, trade and economy. At that time, intercity wheeled and rail traffic dropped by 90% in major European Union (EU) cities. According to [2], COVID-19 transmission was associated with air pollution and environmental factors. Thus, the decrease in wheeled, rail, air and maritime traffic during the pandemic to reduce social interaction should further represent a reduction in air pollution and also in COVID-19 transmission.

Although the impact on worldwide maritime transport was initially not very significant, vessel activity slowed down its pace considerably as uncertainties kept growing [3]. For example, the 40% demand fall in the car manufacturing industry resulted, in turn, in a reduction of car carrier operations. So far, commodity trade seems the only sector able to weather the crisis with minor impact on cargo vessels. Since July 31, 2020, the industry has been on a slow recovery path although, at that time, passenger traffic was still limited mostly to ferry crossings as cruise tourism was still banned in major destinations across the globe [4].

Spain was one of the EU countries most hit by the COVID-19 crisis, with a staggering 288,522 confirmed cases and 28,455 officially declared deaths in July, 2020 [5]. National lockdown entered into force at midnight on March 16, 2020 and extended until June 22, 2020, with a nationwide home quarantine week running from April 6 to April 13. Air traffic was reduced by 90%, wheeled traffic dropped to a residual 20% and passenger traffic by sea was completely disrupted [6].

Zooming in on greater Barcelona, it is observed that it ranks among the top most polluted areas in Spain and the EU in terms of NOx and CO_2. The 2018 indicators of nitrogen oxides (NOx) emissions and particles in suspension with a diameter of less than 10 μm (PM_{10}) revealed that the Port of Barcelona's emissions (from accessing vehicles, civil works, vessels and machinery used in port operations) represent between 15% and 20% of the total emissions of the metropolitan region of Barcelona [7]. In relation to environmental impact, the Port of Barcelona has estimated that cruise vessel emissions contribute to about 1.2% of the city's air pollution, 0.23% of NOx levels and 0.23% of PM_{10} levels [8].

The confinement measures adopted in Barcelona during the early COVID-19 outbreak led to a fall in the overall emissions of major air pollutants. An early study [6] conducted during the first two lockdown weeks in Barcelona found that urban air pollution decreased, with substantial differences among pollutants. Air quality records were obtained at two stations: i) Urban Background (UB) station (located far from emission sources and representative of pollution levels of the urban background) and ii) Traffic (TR) station (located within the urban center and directly affected by traffic emissions). PM_{10} concentrations decreased by 28% and 31% at UB and TR, respectively, and NO_2 concentrations decreased by 47% and 51% at UB and TR, respectively. Reductions in NO_x can be attributed primarily to reduced road traffic activity within the city limits and greater Barcelona, [9]. Lower PM_{10} levels were strongly related to less road traffic and power generation owing to reduced industrial activity. Although a more aggressive PM_{10} reduction was expected, meteorological conditions during the early lockdown days explain the obtained values and indicate that city PM_{10} levels have a regional-background origin and are mostly related to air mass transportation. As stated by [9], during March 2020 the average hourly reduction in NO_2 emissions in the metropolitan areas of Barcelona and Madrid was 50% and 62%, respectively. These results agree with the lower levels observed by [6] . The difference between both cities may be due to the fact that, unlike Madrid, Barcelona is a coastal city and the port must be considered in its emissions pattern.

2. Objectives

The main goal is to explore the correlation between ship traffic, maritime-related pollutant emissions and air quality at the vicinity of the Port of Barcelona using Automatic Identification System (AIS) data.

This main goal can be divided into several specific goals:

1. Develop an open access system to record and decode AIS data with a single antenna.
2. Compare ship emissions with maritime traffic.
3. Compare ship emissions with air quality.

The success of these goals will help the Barcelona School of Nautical studies to increase opportunities for developments in Global MET system through scientific and practical approach, since it will provide the School with new equipment (AIS base) useful for both science and educational purposes. The IAMU community will be benefited through the establishment of a scientific methodology.

3. Methodology

Maritime traffic and port call evolution is assessed through Automatic Identification System (AIS) devices that all vessels over 300 gross tonnage are forced by law to carry. AIS maritime traffic data and real technical data from the Sea-Web database [10] are used to estimate emissions for all vessels according to the STEAM v.2 algorithm [11]. This research uses the study case of the Port of Barcelona (Spain) over a 5-month timespan, from March to July 2020.

3.1. Area of interest

The port of Barcelona is located within the Western Mediterranean Basin in the city of Barcelona. It is an industrial and fishing maritime port, extending all over the coast of Barcelona city from the Old Town to Zona Franca in the limit of the neighbor municipality of l'Hostpitalet. The port of Barcelona is one of the most important ports at the Mediterranean sea, particularly regarding cruising activity.

It is divided in two parts: the logistic area comprising the Logistic Activity Zone, the urban port (Old Port) and the commercial port with mainly ferries and cruises.

Fig. 2 shows an upper overview of the Port of Barcelona.

Fig. 2. Port of Barcelona

The Port of Barcelona has superficial extension of 1110 ha with more than 20 km of docs. It has two different entrances: South entrance, oriented 168.4º North, with a width of 370 m and a depth of 16 m and North entrance, oriented 205º North, a narrower width compared to South entrance of 145 m and also a narrower depth of 11.5 m. This also indicates the main organization of the Port activity with larger vessels entering through the South entrance and smaller vessels (mainly recreational and fishing) using the North entrance.

The Port of Barcelona grow parallel to the city of Barcelona. Initially born as a natural port within the geographical limits of the Montjuic mountain and the former Llobregat estuary. The port has grown in four main phases: one in 1860 towards Barceloneta neighborhood in the North, in 1900 towards Montjuic in the South and in 1965 when it reached the former Llobergat estuary (already visible in Fig. 2). The fourth process took place as part as the Olympic Games investment in public infrastructures. It began in 1989 with the extension of the Logistics Activity Zone (ZAL), and continued in 1994. In 2201-2004 the Llobregat river was deviated towards south to increase the ZAL and the capacity of the containership docking are. This fourth extension was the most important in the history of the Port since it doubled the extension.

Moreover, the fourth extension allowed the port to increase the maritime traffic to a more than 100 million tons and 5 million TEUs after which the train access was also modified to be able to hold the maximum maritime activity.

Fig. 3 shows the map with the different activities giving a general overview of the location and specialization of the different terminals within the port area. It is important to highlight that the zone nearby the port entrances at the outer part is also used as an anchorage area for merchant vessels. Therefore, velocity regulations also affect the outer part of the port.

Although public opinion has always held that the Port of Barcelona is one of the most important sources of air pollution in the city, the COVID pandemic and the mobility restrictions linked to it have been a good opportunity to have a real experiment of a city without road traffic. Recent studies based on the present work suggest that maritime traffic is not the main source of air pollution in the city [12], [13].

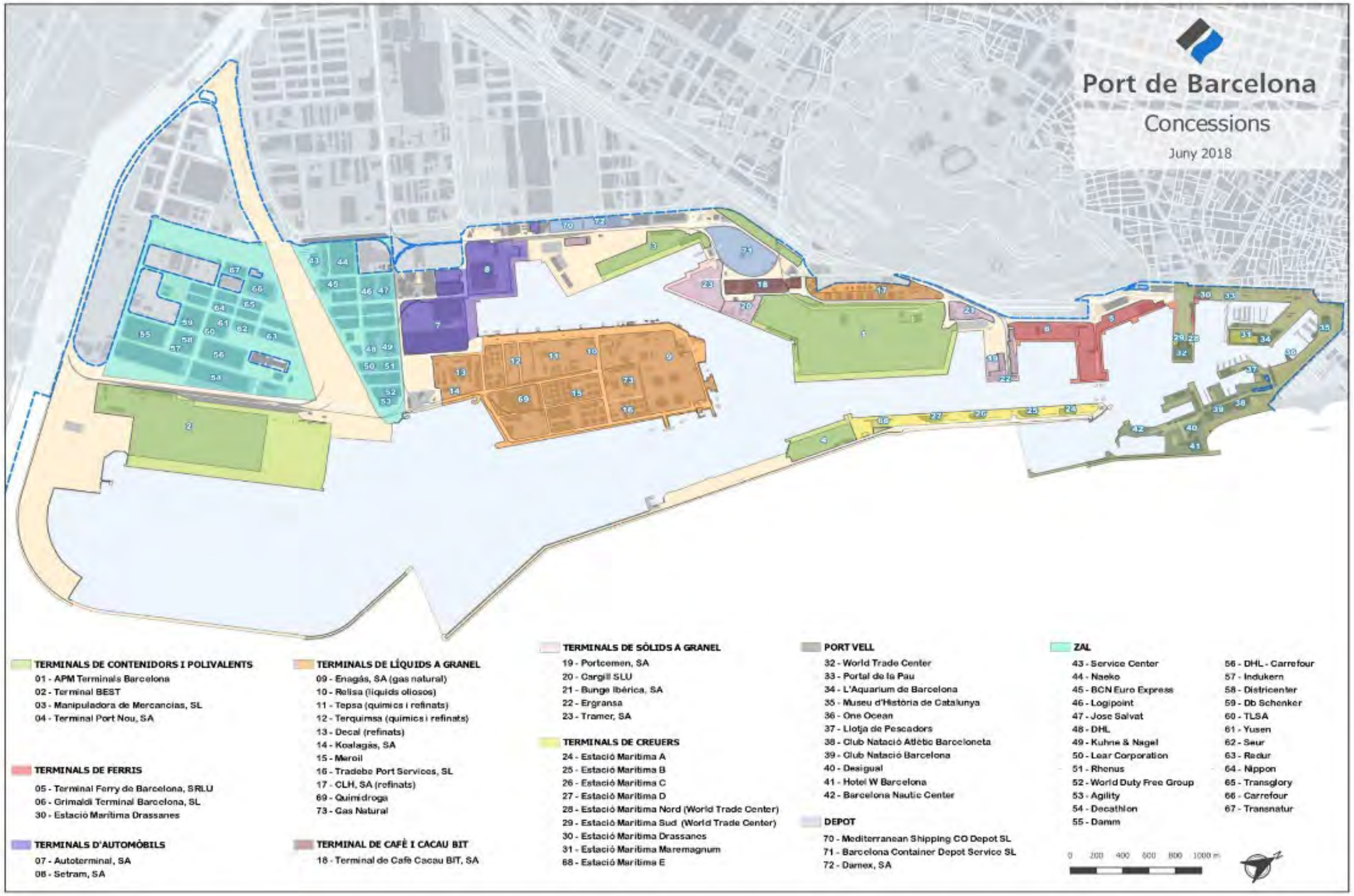


Fig. 3. Map of the Port of Barcelona. Source: APB

3.2. AIS data

The Automatic Identification System (AIS) is a system that transmits the position of a ship so that other ships are aware of it and thus avoid collisions. In addition, the system is also used by Vessel Traffic Services (VTS), which are marine traffic monitoring services established by harbor or port authorities. The aim of using AIS equipment by port authorities is to improve the safety and efficiency of vessel traffic and protect the environment.

The International Maritime Organization (IMO) requires the use of AIS on ships of more than 300 gross tons engaged in international voyages, cargo vessels of more than 500 GT engaged in domestic traffic, and passenger ships. In addition, some national governments have ordered vessels that fall outside IMO regulation to use AIS.

Although the main use of AIS is for collision avoidance, there are multiple applications:

- Collision avoidance: the technology was developed by IMO technical committees in order to prevent collisions of vessels that are not in range of shore-based systems. Since AIS is mainly used by vessels over 300 GT, it is commonly used together with radar. However, AIS advantage is that it transmits valuable information like course and speed, so other vessels can plan the maneuvering in accordance with this.
- Maritime security. AIS data can be automatically processed to create normalized activity patterns for individual vessels, which when breached, create an alert, thus highlighting potential threats for more efficient use of security assets. This can be useful when navigating areas with piracy activity in order to identify what type of vessels are approaching.
- Search and rescue. For coordinating on-scene resources of a marine search and rescue (SAR) operation, it is imperative to have data on the position and navigation status of other ships in the vicinity. AIS can provide additional information and enhance awareness of available resources, even if the AIS range is limited to VHF radio range.
- Accident investigation. AIS provides accurate historical data on time, identity, GPS-based position, compass heading, course over ground, speed (by log/SOG), and rates of turn, rather than the less accurate information provided by radar.
- Aids to navigation. AIS can provide positions and names other than vessels in its vicinity, for example marker positions and dynamic data reflecting the marker's environment (i.e. currents and climatic conditions). The aids can be located on water, like platforms or buoys, and on shore (such as a lighthouse).
- Fishing fleet monitoring and control. It is used by national authorities to track and monitor the activities of their national fishing fleets in their coastal waters.
- Infrastructure protection. AIS information can be used by owners of marine seabed infrastructure, such as cables, pipelines, or platforms in order to monitor the activities of vessels close to their assets in real time.
- Ocean current estimates. A French company called e-Odyn estimates ocean surface currents based on the analysis of AIS data
- Fleet and cargo tracking. AIS data can be viewed in several online platforms in order to check position, route or vessel speed. This info is regularly checked by several companies including ship owners, consignees, ship brokers, etc.

AIS is composed of an automatic tracking system that uses transceivers on ships. These transceivers automatically send information messages every certain period. The messages are sent through a VHF transmitter which is built into the transceiver installed at the ship. Once the messages are sent, these are received by other transmitters installed at other ships or land bases such as VTS. Consequently, the multiple information sent by the different transmitters can be seen at a screen or in a chart plotter, the same way as a radar plotter shows the different ships, but this way according to the latitude and longitude values sent by the transmitter. However, an important disadvantage compared to radars is that AIS is not able to detect objects that do not send signals.

Since 2002, the use of the AIS Class A transceivers started to become mandatory for all ships over 300 gross tons or carrying more than 12 passengers, due to the great safety benefits offered by AIS. For smaller vessels that were outside this SOLAS agreement implemented by IMO, a Class B transceiver was developed that allows fishing and recreational vessels to adapt and install an AIS transceiver at low cost, with less power, but operating on the same AIS network and that it can receive and transmit signals from AIS Class A transceivers installed on commercial vessels.

3.2.1. AIS transmitting system

An AIS transceiver incorporates a GPS receiver, and transmits digitally the GPS position on two VHF marine band frequency channels dedicated to AIS (161.975 MHz and 162.025 MHz). In order to work multiple AIS transceivers simultaneously and to prevent all devices from transmitting at the same time, causing interference and loss of data, AIS transceivers use a system called "Time Division Multiple Access" (TDMA). With the TDMA technology, each AIS transceiver claims a very short “time interval” of 26.6 milliseconds in order to transmit its information.

AIS Class A has been implemented by the International Maritime Organization (IMO) for vessels of 300 gross tonnage and upwards engaged on international voyages, cargo ships of 500 gross tonnage and upwards not engaged on international voyages, as well as passenger ships (more than 12 passengers), regardless of size.

AIS Class B provides limited functionality and is intended for non-SOLAS vessels. It is not required by the International Maritime Organization (IMO) and has been developed for vessels such as work craft and pleasure craft.

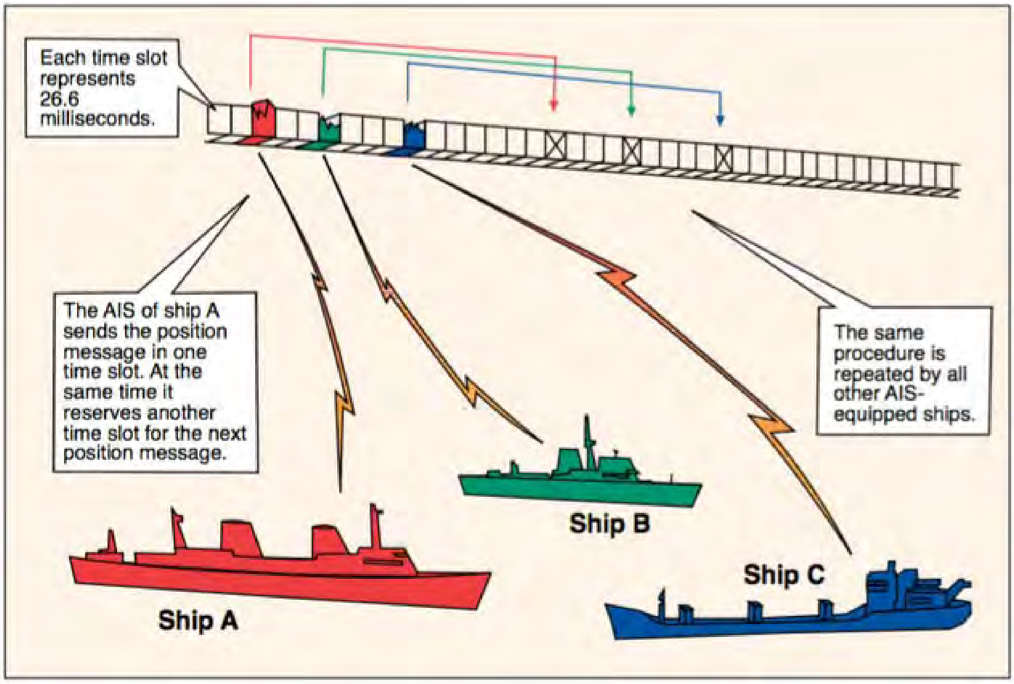


Fig. 4. Time slot distribution. Source: GMDSS Test Equipment

The first TDMA system developed by AIS and used by AIS Class A transceivers, uses a technology called Self-Organizing Time Division Multiple Access (SOTDMA), see Fig. 4. With this system, multiple transceivers are able to claim and reserve time slots automatically. In addition, the transceivers know how to proceed in case of a dispute with another transceiver trying to claim the same interval.

The system is efficient, allowing nearly 4,500 ship signals to work together, giving automatic priority to the distance, i.e. as the number of ships increases, the most distant ships do not get a time slot.

Once Class B transceivers were introduced, the system used a slightly different technology called TDMA "carrier detection", in which Class B transceivers remain stand by listening Class A transceivers, and as soon as it detects an empty time slot, books it and performs its transmission.

The number of transmissions executed by a transceiver, and the type of data sent, varies depending on the class (A or B), its speed, maneuvering status, and navigation status. A Class A AIS transceiver on a fast ferry will be able to transmit its position every two seconds, while a pleasure craft equipped with a Class B AIS transceiver will only transmit its position every 30 seconds.

3.2.2. AIS messages

AIS uses ship's speed and maneuvering status as a means of governing information update rates; in addition, speed and status are also used to ensure the appropriate levels of positional accuracy for ship tracking. Static, dynamic or voyage-related are different information types. These information are valid for different time periods; therefore, they require different update rates:

- Static information (reporting period every 6 minutes and on request):
    - IMO number (if available)
    - MMSI
    - Name and call sign
    - Type of ship
    - Length and beam
    - Location of the position - fixing antenna on the ship (aft or bow / port or starboard of centerline)
- Dynamic information (reporting period depending on speed and course alteration):
    - Position time stamp (in UTC)
    - Ship's position with accuracy indication and integrity status
    - Course over ground (COG)
    - Heading
    - Speed over ground (SOG)
    - Navigational status - manually input: at anchor, aground, underway, etc.
    - Rate of turn (if available)
- Voyage-related information (reporting period every 6 minutes, when data is amended or upon request):
    - Ship's draught
    - Hazardous cargo (type)
    - Destination and ETA (Estimated Time of Arrival) - manual input
    - Route plan (waypoints)
    - Short safety - related messages
- Free format text (sent as required)

In the present project, the AIS messages that have been used in order to develop a method able to estimate the power installed and vessels together with the GhG emissions are the dynamic messages 1, 2, 3 (AIS Class A), and the static messages 5 (AIS Class A), see Table 1 and Table 2.

Table 1. **AIS messages 1, 2 and 3** [14]

Parameter	Number of bits	Description
Message ID	6	Message identifier: 1, 2 or 3
Repeat indicator	2	Used by the repeater to indicate how many times a message has been repeated: 0 = defect 3= Do not repeat more
User ID	30	ISMM number (Maritime mobile service identify)
Navigation status	4	0 = On the way with motor 1 = Anchored 2 = Out of control 3 = Restricted maneuverability 4 = Limited by draft 5 = Berthed 6 = Aground 7 = Fishing 8 = Underway sailing 9 = Reserved for future navigational status amendments for ships carrying hazardous materials, harmful substances or marine pollutants, or IMO category C (HSC) pollutants or hazards 10 = Reserved for future navigation status amendments for ships carrying DG, HS or MP, or IMO Category A Pollutants or Hazards (WIG) 11 = Motor boat towed astern (regional use) 12 = Motor boat sailing or towed sideways (regional use) 13 = Reserved for future use 14 = AIS-SART, MOB-AIS, RLS-AIS 15 = Not defined = defect (used by AIS-SART, MOB-AIS and RLS-AIS tests)
Rate of Turn (ROT)	8	0 to +126 = turning right at 708 ° per min or more 0 to -126= turning left at 708 ° per min or more Values between 0 and 708 ° per min are encoded by: ROT_{AIS}= 4,733 SQRT(ROT_{sensor}) degrees per min, where ROT_{sensor} is the rate of rotation entered by an external rate of rotation indicator (TI). ROT_{AIS} is rounded to the nearest integer value.

		+127= turning right more than 5 ° for 30 s (no TI available) -127= turning left more than 5 ° for 30 s (no TI available) –128 (80 hex) indicates that no turn information is available (default). ROT data should not be derived from COG information
SOG (Speed over ground)	10	Ground speed in steps of 1/10 knot (0-102.2 knots) 1,023 = not available, 1,022 = 102, 2 knots or more
Position accuracy	1	0 = low accuracy (>10m) 1 = high accuracy (<10m) 0 = default
Longitude	28	Longitude in 1/10 000 min (±180 °, East = positive (complement to 2), West = negative (complement to 2). 181 = ($6791AC0_h$) = not available = defect)
Latitude	27	Latitude in 1/10 000 min (±90 °, North = positive (2's complement), South = negative (2's complement); 91 = (3412140h) = not available = default)
Course Over Ground (COG)	12	0 – 3599 = valid COG; 3600 = COG not available (default); 3601 – 4095 = not in use
True heading	9	Grades (0-359) (511 indicates not available = default)
Time display	6	0 - 59 = valid time or defect 60 = timestamp not available 61 = positioning system in manual input mode 62 = electronic position determination system works in dead reckoning mode 63 = positioning system does not work
Special maneuver indicator	2	0 = not available = defect 1 = no special maneuver 2 = special maneuver (i.e. agreement on internal waterway)
Not available - Reserved	3	Not employed. It should be zeroed. Reserved for future use
RAIM flag	1	RAIM (Receiver Autonomous Integrity Monitoring) flag of electronic position determining device. 0 = RAIM not in use = defect 1 = RAIM in use.
Communication status	19	Data message containing SOTDMA and ITDMA protocols.

Number of bits	168	

Table 2. AIS message 5 [14]

Parameter	Number of bits	Description
Message ID	6	The identifier of this message is 5
Repeat indicator	2	Used by the repeater to indicate how many times a message has been repeated: 0 = defect 3 = not repeat any more
User ID	30	ISMM number
AIS version indicator	2	0 = station conforming to Recommendation ITU-R M.1371-1 1 = station in accordance with ITU-R Recommendation M.1371-3 (or later) 2 = station in accordance with ITU-R Recommendation M.1371-5 (or later) 3 = station subject to future issues
IMO number	30	0= not available = defect - Not applicable to SAR aircraft 0000000001-0000999999 not used 0001000000-0009999999 = valid IMO number; 0010000000-1073741823 = official flag state number
Call sign	42	7 6-bit ASCII characters, @@@@@@@ = not available = default
Name	120	Maximum 20 6-bit ASCII characters, as defined in Table 44 «@@@@@@@@@@@@@@@@@@@@» = not available = default
Type of ship and type of cargo	8	0 = not available or no ship = default 0 = type of ship not available (default) 10 – 19 = not in use 20 – 28 = wing in ground (WIG) aircraft 29 = search and rescue (SAR) aircraft 30 = fishing 31 – 32 = tugboat 33 = dredger; 34 = dive vessel;

		35 = military vessel 36 = sailing vessel 37 = pleasure craft 38 – 39 = not in use 40 – 49 = high speed craft (HSC) 50 = pilot boat 51 = SAR vessel 52 = tugboat 53 = port tender 54 = anti-pollution craft 55 = law enforcement boat 56 – 57 = local boat 58 = medical transport 59 = special craft 60 – 69 = passenger vessel 70 – 79 = cargo vessel 80 – 89 = tanker vessel 90 – 99 = other 100-199 = reserved, for regional use 200-255 = reserved, for future use Not applicable to SAR aircraft
Global dimension / reference position	30	A, B, C and D dimensions are given: A = forward length; 0 – 511; 511 = 511m or longer B = aft length; 0 – 511; 511 = 511m or longer C = port beam; 0 – 61; 61 = 61m or longer D = starboard beam; 0 – 61; 61 = 61m or longer
Type of electronic position determining device	4	0 = undefined (default) 1 = GPS 2 = GLONASS 3 = GPS / GLONASS combined 4 = Loran-C 5 = Chayka 6 = integrated navigation system 7 = guarded 8 = Galileo 9-14 = not employed 15 = internal GNSS
Estimated Time of Arrival (ETA)	20	Estimated time of arrival; MMDDHHMM UTC Bits 19-16: month; 1-12; 0 = not available = default Bits 15-11: day; 1-31; 0 = not available = default Bits 10-6: h; 0-23; 24 = not available = default

		Bits 5-0: min; 0-59; 60 = not available = default.
Maximum current static draught	8	In 1/10 m, 255 = draught 25.5 m or greater, 0 = not available = default; in accordance with IMO
Destination	120	Maximum 20 6-bit ASCII characters; @@@@@@@@@@@@@@@@@@@@ = not available.
DTE	1	Value from 0 to 1, where: 0 = DTE available; 1 = DTE not available.
Spare	1	Not in use, set to 0
TOTAL	424	

3.2.3. AIS data management

Original AIS data was collected through a receiving station located at the Department of Nautical Sciences and Engineering at the Barcelona School of Nautical Studies (UPC-BarcelonaTech). Fig. 5 shows the location of the antenna within the roof of the building along with the receiver (Class B-AIS, SeaTraceR AIS Class B Transponder S.287).

The location of the station allows monitoring all the shipping activity within the Port of Barcelona and around 120 nautical miles. The raw AIS radio messages captured by the station were decoded –details given below- in order to convert them to a more human-friendly format and derive the following information for each ship and time step. From all the information provided by messages 1,2,3 and 5, only the following items are selected:

- <u>Maritime Mobile Service Identity number (MMSI):</u> A unique identification number for each vessel AIS station.
- <u>International Maritime Organisation number (IMO</u>): A unique identifier for each ship.
- <u>Date:</u> The year, month, day, hour and second field of the UTC time when the AIS message was sent.
- <u>Ship and cargo type</u>: description of the ship type and the cargo carried by the ship.
- <u>Latitude, longitude:</u> Geographic position coordinates.
- <u>Navigation status:</u> description of the engine activity (e.g hotelling, maneuvering, cruising).
- <u>Speed Over Ground:</u> real time velocity of the ship.
- <u>Rate of Turn:</u> real time turning velocity of the ship.
- <u>Vessel dimensions:</u> length, beam and draft of the ship.

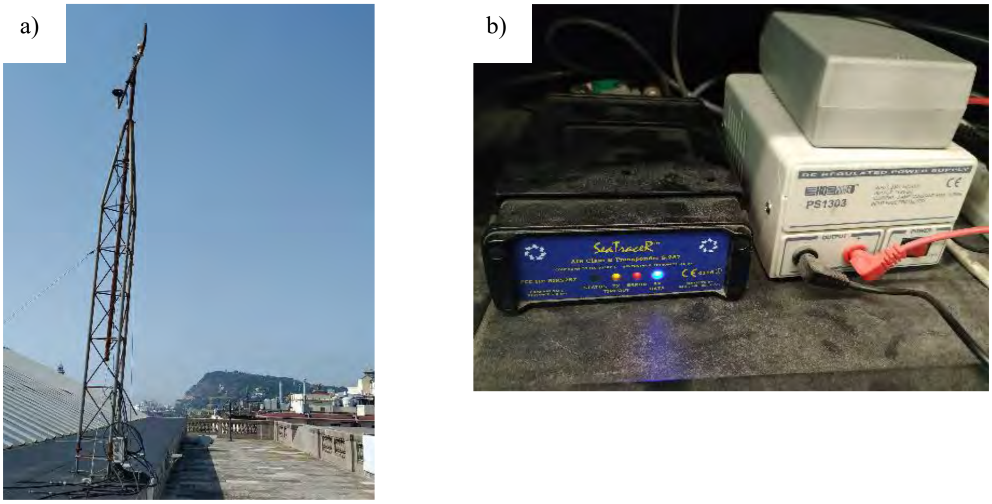


Fig. 5. AIS receiver located at the roof of the Barcelona School of Nautical Studies (FNB-UPC). a) antena of the AIS System; b) AIS receiver

The AIS receiver is connected to a RaspberryPi that stores all the messages using NMEA system in hourly files. The main problem of this system is that the AIS dynamic messages only carry the second of the time-stamp when the message was sent. Therefore, local minute and second of the RasPi are added as the last four digits in each message. Results comparing the second when the message was sent and the second when it was received are detailed in the results section and were also published in [15].

The management process of the data is summarized in Fig. 6. All processes were performed using Python platform in a Linux environment.

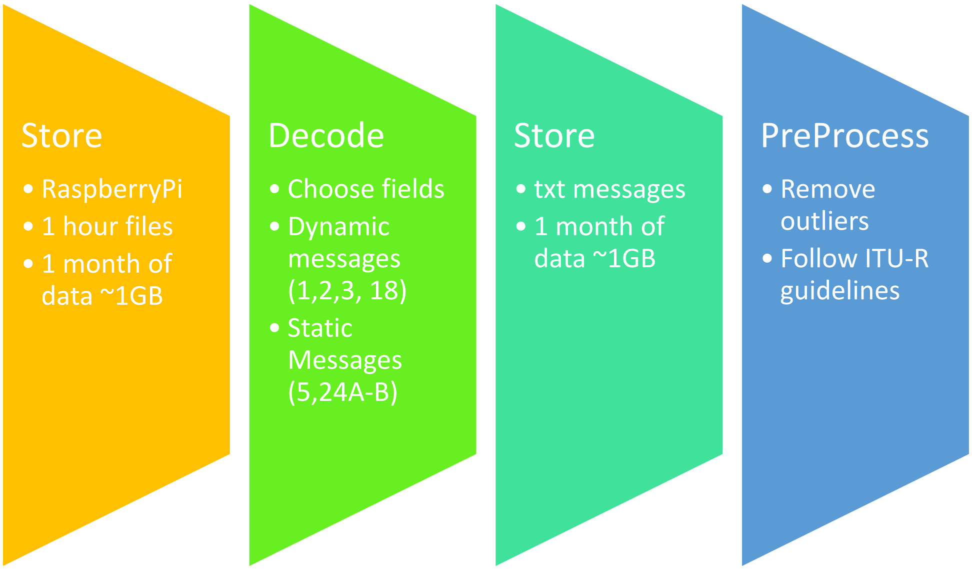


Fig. 6. Flow chart of AIS data management

The folders are organized as shown in Fig. 7. Directory *src* contains the scripts needed to pre-process-which include the decoding algorithms. Data is stored in the *data* directory and separated so raw un-processed messages are not overwritten, *interim* folder is where decoded and filtered messages are saved and *processed* folder is for specific processing of the AIS data depending on the project they are used for.

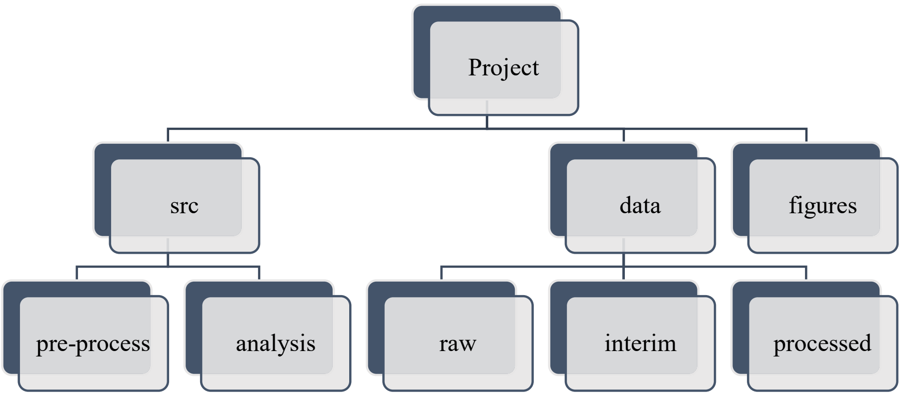


Fig. 7. Folder organization of the project

Decoding system

Script detailed in section 8.2, *AIS_decoder.py*, uses the public library pyAISm available on github. It also uses *AISfunctions.py*, section 8.1,where home tailored functions are defined and needed to write decoded messages following a specific format.

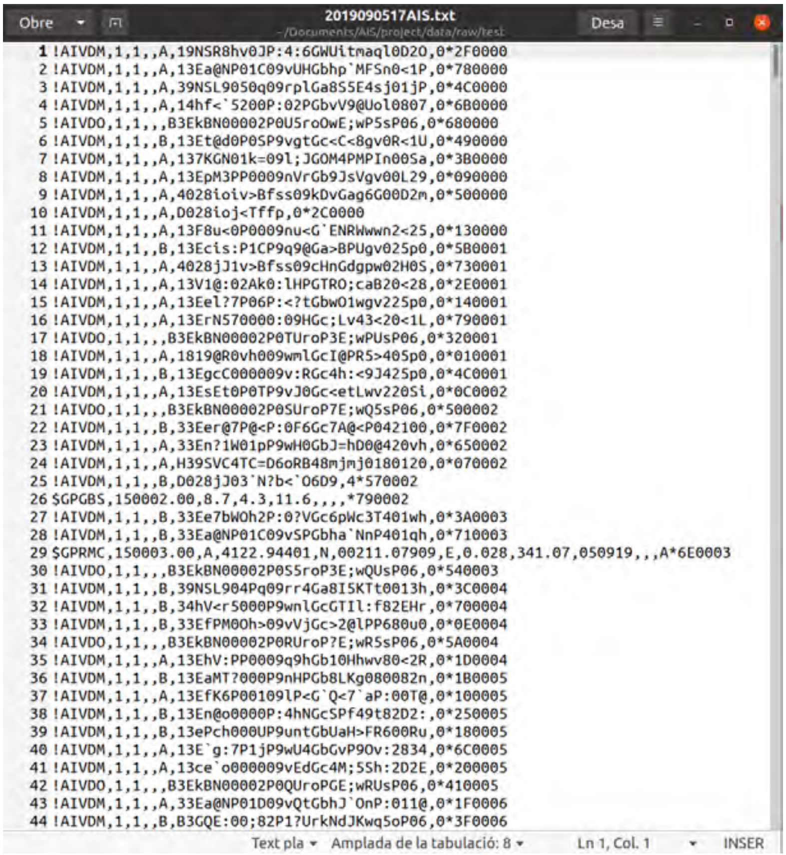
An example of the original non decoded messages is shown in Fig. 8, along with the four added digits at the end of each message indicating the timestamp (minute and second) of the local RasPi.

The input to *AIS_decoder.py* is a list of all the hour-files the user needs to decode. This list should be previously created by typing into the terminal window the following sentence: "ls *.txt>files_list.txt". The folder containing the list of txt files is a user input as well. The name of this folder has to be also repeated in the *interim* directory.

The output of *AIS_decoder.py* is a list of 5 different txt files with the decoded messages containing the specific information of each message. Headers are not included and shall be checked using the recommendations [14] and also *AISfunctions.py*:

- decoded_123.txt, see Fig. 9.
- decoded_5.txt, see Fig. 10.
- decoded_18.txt, see Fig. 11.
- decoded_24A.txt, see Fig. 12
- decoded_24B.txt, see Fig. 12

All the decoded messages have an initial column storing the time stamp with the minute and second of the local RasPi (e.g. 20190905140000 is YearMonthDayHourMinuteSecond)




```
!AIVDM,1,1,,A,19NSR8hv0JP:4:6GWUitmaql0D2O,0*2F0000
!AIVDM,1,1,,A,13Ea@NP01C09vUHGbhp`MFSn0<1P,0*780000
!AIVDM,1,1,,A,39NSL9050q09rplGa8S5E4sj01jP,0*4C0000
!AIVDM,1,1,,A,14hf<`5200P:02PGbvV9@Uol0807,0*6B0000
!AIVDO,1,1,,,B3EkBN00002P0U5roOwE;wP5sP06,0*680000
!AIVDM,1,1,,B,13Et@d0P0SP9vgtGc<C<8gv0R<1U,0*490000
!AIVDM,1,1,,A,137KGN01k=09l;JGOM4PMPIn00Sa,0*3B0000
!AIVDM,1,1,,A,13EpM3PP0009nVrGb9JsVgv00L29,0*090000
!AIVDM,1,1,,A,4028ioiv>Bfss09kDvGag6G00D2m,0*500000
!AIVDM,1,1,,A,D028ioj<Tffp,0*2C0000
!AIVDM,1,1,,A,13F8u<0P0009nu<G`ENRWwwn2<25,0*130000
!AIVDM,1,1,,B,13Ecis:P1CP9q9@Ga>BPUgv025p0,0*5B0001
!AIVDM,1,1,,A,4028jJ1v>Bfss09cHnGdgpw02H0S,0*730001
!AIVDM,1,1,,A,13V1@:02Ak0:lHPGTRO;caB20<28,0*2E0001
!AIVDM,1,1,,A,13Eel?7P06P:<?tGbwO1wgv225p0,0*140001
!AIVDM,1,1,,A,13ErN570000:09HGc;Lv43<20<1L,0*790001
!AIVDO,1,1,,,B3EkBN00002P0TUroP3E;wPUsP06,0*320001
!AIVDM,1,1,,A,1819@R0vh009wmlGcI@PR5>405p0,0*010001
!AIVDM,1,1,,B,13EgcC000009v:RGc4h:<9J425p0,0*4C0001
!AIVDM,1,1,,A,13EsEt0P0TP9vJ0Gc<etLwv220Si,0*0C0002
!AIVDO,1,1,,,B3EkBN00002P0SUroP7E;wQ5sP06,0*500002
!AIVDM,1,1,,B,33Eer@7P@<P:0F6Gc7A@<P042100,0*7F0002
!AIVDM,1,1,,A,33En?1W01pP9wH0GbJ=hD0@420vh,0*650002
!AIVDM,1,1,,A,H39SVC4TC=D6oRB48mjmj0180120,0*070002
!AIVDM,1,1,,B,D028jJ03`N?b<`O6D9,4*570002
$GPGBS,150002.00,8.7,4.3,11.6,,,,*790002
!AIVDM,1,1,,B,33Ee7bWOh2P:0?VGc6pWc3T401wh,0*3A0003
!AIVDM,1,1,,B,33Ea@NP01C09vSPGbha`NnP401qh,0*710003
$GPRMC,150003.00,A,4122.94401,N,00211.07909,E,0.028,341.07,050919,,,A*6E0003
!AIVDO,1,1,,,B3EkBN00002P0S5roP3E;wQUsP06,0*540003
!AIVDM,1,1,,B,39NSL904Pq09rr4Ga8I5KTt0013h,0*3C0004
!AIVDM,1,1,,B,34hV<r5000P9wnlGcGTIl:f82EHr,0*700004
!AIVDM,1,1,,B,33EfPM0Oh>09vVjGc>2@lPP680u0,0*0E0004
!AIVDO,1,1,,,B3EkBN00002P0RUroP?E;wR5sP06,0*5A0004
!AIVDM,1,1,,A,13EhV:PP0009q9hGb10Hhwv80<2R,0*1D0004
!AIVDM,1,1,,B,13EaMT?000P9nHPGb8LKg080082n,0*1B0005
!AIVDM,1,1,,A,13EfK6P00109lP<G`Q<7`aP:00T@,0*100005
!AIVDM,1,1,,B,13En@o0000P:4hNGcSPf49t82D2:,0*250005
!AIVDM,1,1,,B,13ePch000UP9untGbUaH>FR600Ru,0*180005
!AIVDM,1,1,,A,13E`g:7P1jP9wU4GbGvP9Ov:2834,0*6C0005
!AIVDM,1,1,,A,13ce`o000009vEdGc4M;5Sh:2D2E,0*200005
!AIVDO,1,1,,,B3EkBN00002P0QUroPGE;wRUsP06,0*410005
!AIVDM,1,1,,A,33Ea@NP01D09vQtGbhJ`OnP:011@,0*1F0006
!AIVDM,1,1,,B,B3GQE:00;82P1?UrkNdJKwq5oP06,0*3F0006
```

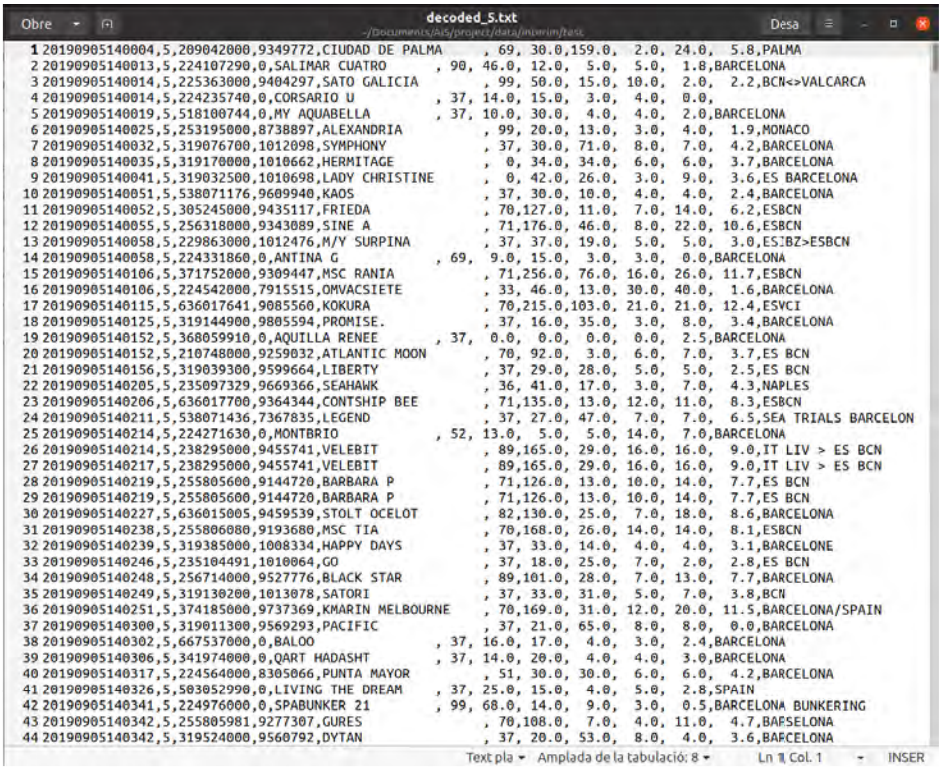

Fig. 8. Example of a raw messages file stored by the Raspberry Pi

Obre decoded_123.txt Desa
~/Documents/AIS/project/data/interim/test

```
20190905140000,59,1,224063980,10,-128.0,   0.0,2.16503,41.31772,135.6,511
20190905140000,59,1,209042000, 0,   5.0, 212.0,2.26073,41.20663,185.0,185
20190905140000,01,1,538071176, 0,  -7.0,   0.0,2.18398,41.38036, 13.6,164
20190905140000,00,1,224271630, 0,-128.0,   0.0,2.16517,41.31746,145.8,511
20190905140001,02,1,224127820, 0,   0.0,   0.0,2.16573,41.34755,294.0,116
20190905140001,00,1,224542000, 0,-128.0,   0.0,2.15362,41.29685, 39.5,511
20190905140001,01,1,224097340, 7,-128.0,  32.0,2.36138,41.40131, 20.5,511
20190905140001,02,1,224318960, 0,-128.0,   0.0,2.17889,41.37520,308.2,511
20190905140003,47,1,636017641, 0,   0.0, 183.0,2.11363,41.20208,215.1,216
20190905140003,03,1,225394000,15,-128.0,  71.0,2.15358,41.34215, 90.7,511
20190905140003,03,1,224142890, 0,-128.0,   0.0,2.16110,41.34272,224.3,511
20190905140004,04,1,256318000, 0,   0.0,  74.0,2.18203,41.28897,342.0,342
20190905140004,05,1,224107290, 0,   0.0,   0.0,2.14529,41.30183,259.0,303
20190905140005,06,1,225363000, 0,   0.0,  37.0,2.17028,41.33807,190.2, 11
20190905140005,05,1,224235740, 0,   0.0,   0.0,2.20075,41.38476,360.0,317
20190905140006,05,3,319076700, 5,   0.0,   0.0,2.18479,41.36981,197.0,186
20190905140006,06,1,224569880, 0,-128.0,   0.0,2.15275,41.34558, 42.3,511
20190905140006,05,1,209042000, 0,  -7.0, 212.0,2.26068,41.20625,185.0,184
20190905140007,07,1,224331860, 0,-128.0,  90.0,2.20476,41.38529, 45.4,511
20190905140007,07,3,224022650, 0, 127.0,   0.0,2.17903,41.37530,333.1,291
20190905140008,08,1,224022660, 0,-128.0,   2.0,2.18540,41.37640,304.7,511
20190905140008,09,3,319393000, 5,   0.0,   0.0,2.18405,41.37965,291.7,344
20190905140009,09,1,224350720, 0,   0.0,   0.0,2.18260,41.36488,273.3,252
20190905140009,09,3,224022650, 0, 127.0,   1.0,2.17902,41.37530,333.1,292
20190905140009,10,1,224334000, 0,-128.0,   0.0,2.17923,41.37544,360.0,511
20190905140009,54,1,636017641, 0,   0.0, 183.0,2.11317,41.20159,215.1,216
20190905140010,10,1,224097340, 7,-128.0,  36.0,2.36145,41.40145, 31.1,511
20190905140010,10,1,224271630, 0,-128.0,   0.0,2.16518,41.31746,208.1,511
20190905140011,11,1,605266120, 0,   0.0,  85.0,2.35471,41.14071,139.8,145
20190905140011,12,1,538071176, 0,  -7.0,   0.0,2.18397,41.38036, 13.6,163
20190905140011,12,1,224127820, 0,   0.0,   0.0,2.16574,41.34756,295.4,116
20190905140012,11,3,224022650, 0, 127.0,   1.0,2.17902,41.37530,333.1,293
20190905140012,11,1,225394000,15,-128.0,  74.0,2.15393,41.34222, 69.7,511
20190905140012,12,1,224318960, 0,-128.0,   0.0,2.17888,41.37521,315.6,511
20190905140012,11,1,209042000, 0,  11.0, 212.0,2.26060,41.20547,185.0,185
20190905140013,13,1,224304660, 7,   0.0,  33.0,2.06435,41.23481, 57.0, 60
20190905140013,14,1,256318000, 0,   0.0,  74.0,2.18190,41.28930,342.0,342
20190905140014,14,1,224142890, 0,-128.0,   0.0,2.16110,41.34272,224.3,511
20190905140014,14,3,224235740, 0,   0.0,   0.0,2.20075,41.38476,360.0,317
20190905140015,16,1,225363000, 0,   0.0,  38.0,2.17025,41.33789,188.3, 11
20190905140015,59,1,636017641, 0,   0.0, 183.0,2.11285,41.20125,215.1,216
20190905140016,16,1,224569880, 0,-128.0,   0.0,2.15274,41.34559, 66.5,511
20190905140016,17,1,224107290, 0,   0.0,   0.0,2.14529,41.30183,265.2,303
20190905140016,16,3,224022650, 0, 127.0,   1.0,2.17902,41.37530,333.1,294
```

Text pla ▾ Amplada de la tabulació: 8 ▾ Ln 2, Col. 15 ▾ INSER

Fig. 9. Example of decoded ClassA dynamic messages (1,2,3)



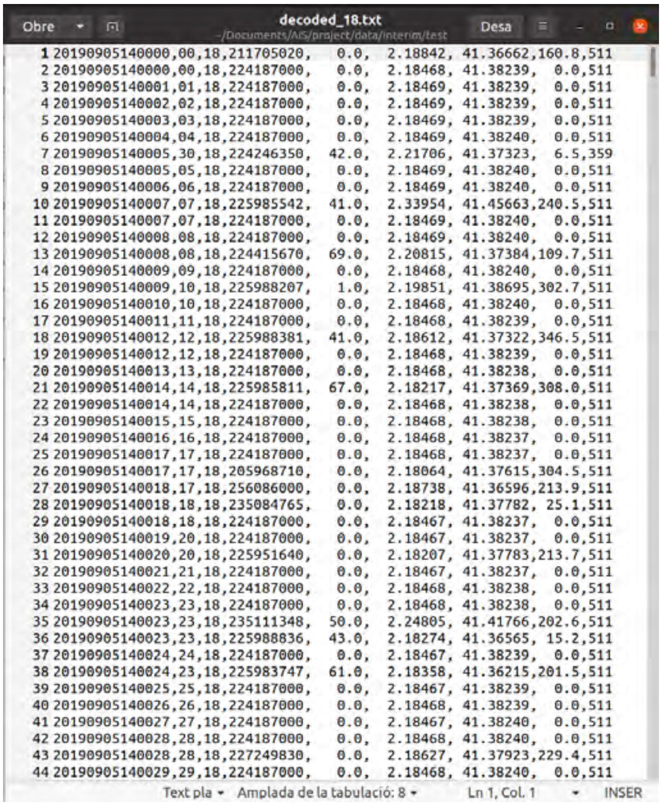
decoded_5.txt

~/Documents/AIS/project/data/interim/test

```
20190905140004,5,209042000,9349772,CIUDAD DE PALMA      , 69, 30.0,159.0,  2.0, 24.0,  5.8,PALMA
20190905140013,5,224107290,0,SALIMAR CUATRO      , 90, 46.0, 12.0,  5.0,  5.0,  1.8,BARCELONA
20190905140014,5,225363000,9404297,SATO GALICIA         , 99, 50.0, 15.0, 10.0,  2.0,  2.2,BCN<>VALCARCA
20190905140014,5,224235740,0,CORSARIO U          , 37, 14.0, 15.0,  3.0,  4.0,  0.0,
20190905140019,5,518100744,0,MY AQUABELLA        , 37, 10.0, 30.0,  4.0,  4.0,  2.0,BARCELONA
20190905140025,5,253195000,8738897,ALEXANDRIA           , 99, 20.0, 13.0,  3.0,  4.0,  1.9,MONACO
20190905140032,5,319076700,1012098,SYMPHONY             , 37, 30.0, 71.0,  8.0,  7.0,  4.2,BARCELONA
20190905140035,5,319170000,1010662,HERMITAGE            ,  0, 34.0, 34.0,  6.0,  6.0,  3.7,BARCELONA
20190905140041,5,319032500,1010698,LADY CHRISTINE       ,  0, 42.0, 26.0,  3.0,  9.0,  3.6,ES BARCELONA
20190905140051,5,538071176,9609940,KAOS                 , 37, 30.0, 10.0,  4.0,  4.0,  2.4,BARCELONA
20190905140052,5,305245000,9435117,FRIEDA               , 70,127.0, 11.0,  7.0, 14.0,  6.2,ESBCN
20190905140055,5,256318000,9343089,SINE A               , 71,176.0, 46.0,  8.0, 22.0, 10.6,ESBCN
20190905140058,5,229863000,1012476,M/Y SURPINA          , 37, 37.0, 19.0,  5.0,  5.0,  3.0,ESIBZ>ESBCN
20190905140058,5,224331860,0,ANTINA G            , 69,  9.0, 15.0,  3.0,  3.0,  0.0,BARCELONA
20190905140106,5,371752000,9309447,MSC RANIA            , 71,256.0, 76.0, 16.0, 26.0, 11.7,ESBCN
20190905140106,5,224542000,7915515,OMVACSIETE           , 33, 46.0, 13.0, 30.0, 40.0,  1.6,BARCELONA
20190905140115,5,636017641,9085560,KOKURA               , 70,215.0,103.0, 21.0, 21.0, 12.4,ESVCI
20190905140125,5,319144900,9805594,PROMISE.             , 37, 16.0, 35.0,  3.0,  8.0,  3.4,BARCELONA
20190905140152,5,368059910,0,AQUILLA RENEE       , 37,  0.0,  0.0,  0.0,  0.0,  2.5,BARCELONA
20190905140152,5,210748000,9259032,ATLANTIC MOON        , 70, 92.0,  3.0,  6.0,  7.0,  3.7,ES BCN
20190905140156,5,319039300,9599664,LIBERTY              , 37, 29.0, 28.0,  5.0,  5.0,  2.5,ES BCN
20190905140205,5,235097329,9669366,SEAHAWK              , 36, 41.0, 17.0,  3.0,  7.0,  4.3,NAPLES
20190905140206,5,636017700,9364344,CONTSHIP BEE         , 71,135.0, 13.0, 12.0, 11.0,  8.3,ESBCN
20190905140211,5,538071436,7367835,LEGEND               , 37, 27.0, 47.0,  7.0,  7.0,  6.5,SEA TRIALS BARCELON
20190905140214,5,224271630,0,MONTBRIO            , 52, 13.0,  5.0,  5.0, 14.0,  7.0,BARCELONA
20190905140214,5,238295000,9455741,VELEBIT              , 89,165.0, 29.0, 16.0, 16.0,  9.0,IT LIV > ES BCN
20190905140217,5,238295000,9455741,VELEBIT              , 89,165.0, 29.0, 16.0, 16.0,  9.0,IT LIV > ES BCN
20190905140219,5,255805600,9144720,BARBARA P            , 71,126.0, 13.0, 10.0, 14.0,  7.7,ES BCN
20190905140219,5,255805600,9144720,BARBARA P            , 71,126.0, 13.0, 10.0, 14.0,  7.7,ES BCN
20190905140227,5,636015005,9459539,STOLT OCELOT         , 82,130.0, 25.0,  7.0, 18.0,  8.6,BARCELONA
20190905140238,5,255806080,9193680,MSC TIA              , 70,168.0, 26.0, 14.0, 14.0,  8.1,ESBCN
20190905140239,5,319385000,1008334,HAPPY DAYS           , 37, 33.0, 14.0,  4.0,  4.0,  3.1,BARCELONE
20190905140246,5,235104491,1010064,GO                   , 37, 18.0, 25.0,  7.0,  2.0,  2.8,ES BCN
20190905140248,5,256714000,9527776,BLACK STAR           , 89,101.0, 28.0,  7.0, 13.0,  7.7,BARCELONA
20190905140249,5,319130200,1013078,SATORI               , 37, 33.0, 31.0,  5.0,  7.0,  3.8,BCN
20190905140251,5,374185000,9737369,KMARIN MELBOURNE     , 70,169.0, 31.0, 12.0, 20.0, 11.5,BARCELONA/SPAIN
20190905140300,5,319011300,9569293,PACIFIC              , 37, 21.0, 65.0,  8.0,  8.0,  0.0,BARCELONA
20190905140302,5,667537000,0,BALOO               , 37, 16.0, 17.0,  4.0,  3.0,  2.4,BARCELONA
20190905140306,5,341974000,0,QART HADASHT        , 37, 14.0, 20.0,  4.0,  4.0,  3.0,BARCELONA
20190905140317,5,224564000,8305066,PUNTA MAYOR          , 51, 30.0, 30.0,  6.0,  6.0,  4.2,BARCELONA
20190905140326,5,503052990,0,LIVING THE DREAM    , 37, 25.0, 15.0,  4.0,  5.0,  2.8,SPAIN
20190905140341,5,224976000,0,SPABUNKER 21        , 99, 68.0, 14.0,  9.0,  3.0,  0.5,BARCELONA BUNKERING
20190905140342,5,255805981,9277307,GURES                , 70,108.0,  7.0,  4.0, 11.0,  4.7,BAFSELONA
20190905140342,5,319524000,9560792,DYTAN                , 37, 20.0, 53.0,  8.0,  4.0,  3.6,BAFCELONA
```

Text pla ▾ Amplada de la tabulació: 8 ▾ Ln 1, Col. 1 ▾ INSER

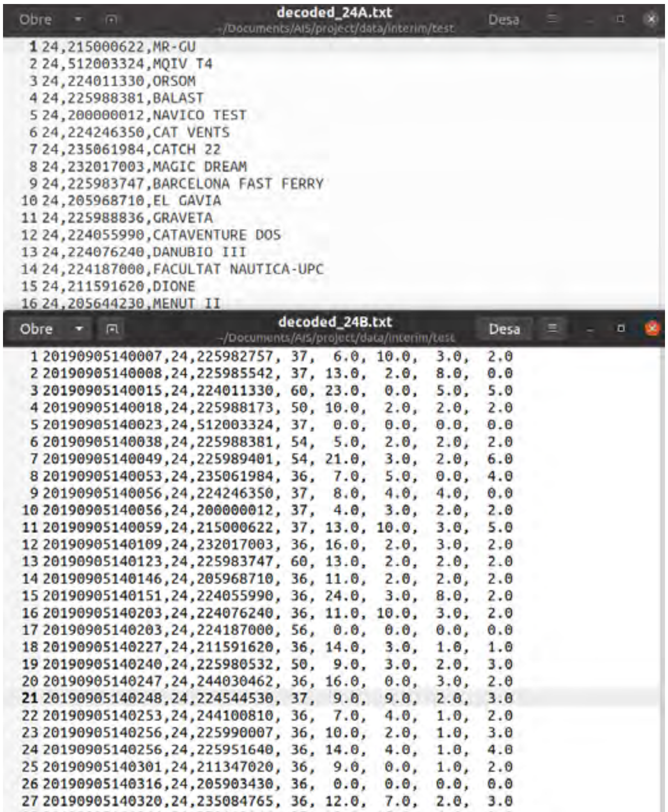
Fig. 10. Example of decoded ClassA static messages (5)



decoded_18.txt
~/Documents/AIS/project/data/interim/test

```
 1 20190905140000,00,18,211705020,   0.0,  2.18842, 41.36662,160.8,511
 2 20190905140000,00,18,224187000,   0.0,  2.18468, 41.38239,  0.0,511
 3 20190905140001,01,18,224187000,   0.0,  2.18469, 41.38239,  0.0,511
 4 20190905140002,02,18,224187000,   0.0,  2.18469, 41.38239,  0.0,511
 5 20190905140003,03,18,224187000,   0.0,  2.18469, 41.38239,  0.0,511
 6 20190905140004,04,18,224187000,   0.0,  2.18469, 41.38240,  0.0,511
 7 20190905140005,30,18,224246350,  42.0,  2.21706, 41.37323,  6.5,359
 8 20190905140005,05,18,224187000,   0.0,  2.18469, 41.38240,  0.0,511
 9 20190905140006,06,18,224187000,   0.0,  2.18469, 41.38240,  0.0,511
10 20190905140007,07,18,225985542,  41.0,  2.33954, 41.45663,240.5,511
11 20190905140007,07,18,224187000,   0.0,  2.18469, 41.38240,  0.0,511
12 20190905140008,08,18,224187000,   0.0,  2.18469, 41.38240,  0.0,511
13 20190905140008,08,18,224415670,  69.0,  2.20815, 41.37384,109.7,511
14 20190905140009,09,18,224187000,   0.0,  2.18468, 41.38240,  0.0,511
15 20190905140009,10,18,225988207,   1.0,  2.19851, 41.38695,302.7,511
16 20190905140010,10,18,224187000,   0.0,  2.18468, 41.38240,  0.0,511
17 20190905140011,11,18,224187000,   0.0,  2.18468, 41.38239,  0.0,511
18 20190905140012,12,18,225988381,  41.0,  2.18612, 41.37322,346.5,511
19 20190905140012,12,18,224187000,   0.0,  2.18468, 41.38239,  0.0,511
20 20190905140013,13,18,224187000,   0.0,  2.18468, 41.38238,  0.0,511
21 20190905140014,14,18,225985811,  67.0,  2.18217, 41.37369,308.0,511
22 20190905140014,14,18,224187000,   0.0,  2.18468, 41.38238,  0.0,511
23 20190905140015,15,18,224187000,   0.0,  2.18468, 41.38238,  0.0,511
24 20190905140016,16,18,224187000,   0.0,  2.18468, 41.38237,  0.0,511
25 20190905140017,17,18,224187000,   0.0,  2.18468, 41.38237,  0.0,511
26 20190905140017,17,18,205968710,   0.0,  2.18064, 41.37615,304.5,511
27 20190905140018,17,18,256086000,   0.0,  2.18738, 41.36596,213.9,511
28 20190905140018,18,18,235084765,   0.0,  2.18218, 41.37782, 25.1,511
29 20190905140018,18,18,224187000,   0.0,  2.18467, 41.38237,  0.0,511
30 20190905140019,20,18,224187000,   0.0,  2.18467, 41.38237,  0.0,511
31 20190905140020,20,18,225951640,   0.0,  2.18207, 41.37783,213.7,511
32 20190905140021,21,18,224187000,   0.0,  2.18467, 41.38237,  0.0,511
33 20190905140022,22,18,224187000,   0.0,  2.18468, 41.38238,  0.0,511
34 20190905140023,23,18,224187000,   0.0,  2.18468, 41.38238,  0.0,511
35 20190905140023,23,18,235111348,  50.0,  2.24805, 41.41766,202.6,511
36 20190905140023,23,18,225988836,  43.0,  2.18274, 41.36565, 15.2,511
37 20190905140024,24,18,224187000,   0.0,  2.18467, 41.38239,  0.0,511
38 20190905140024,23,18,225983747,  61.0,  2.18358, 41.36215,201.5,511
39 20190905140025,25,18,224187000,   0.0,  2.18467, 41.38239,  0.0,511
40 20190905140026,26,18,224187000,   0.0,  2.18468, 41.38239,  0.0,511
41 20190905140027,27,18,224187000,   0.0,  2.18467, 41.38240,  0.0,511
42 20190905140028,28,18,224187000,   0.0,  2.18468, 41.38240,  0.0,511
43 20190905140028,28,18,227249830,   0.0,  2.18627, 41.37923,229.4,511
44 20190905140029,29,18,224187000,   0.0,  2.18468, 41.38240,  0.0,511
```


Fig. 11. Example of decoded ClassB dynamic messages (18)


decoded_24A.txt
~/Documents/AIS/project/data/interim/test

```
24,215000622,MR-GU
24,512003324,MQIV T4
24,224011330,ORSOM
24,225988381,BALAST
24,200000012,NAVICO TEST
24,224246350,CAT VENTS
24,235061984,CATCH 22
24,232017003,MAGIC DREAM
24,225983747,BARCELONA FAST FERRY
24,205968710,EL GAVIA
24,225988836,GRAVETA
24,224055990,CATAVENTURE DOS
24,224076240,DANUBIO III
24,224187000,FACULTAT NAUTICA-UPC
24,211591620,DIONE
24,205644230,MENUT II
```

decoded_24B.txt
~/Documents/AIS/project/data/interim/test

```
20190905140007,24,225982757, 37,  6.0, 10.0,  3.0,  2.0
20190905140008,24,225985542, 37, 13.0,  2.0,  8.0,  0.0
20190905140015,24,224011330, 60, 23.0,  0.0,  5.0,  5.0
20190905140018,24,225988173, 50, 10.0,  2.0,  2.0,  2.0
20190905140023,24,512003324, 37,  0.0,  0.0,  0.0,  0.0
20190905140038,24,225988381, 54,  5.0,  2.0,  2.0,  2.0
20190905140049,24,225989401, 54, 21.0,  3.0,  2.0,  6.0
20190905140053,24,235061984, 36,  7.0,  5.0,  0.0,  4.0
20190905140056,24,224246350, 37,  8.0,  4.0,  4.0,  0.0
20190905140056,24,200000012, 37,  4.0,  3.0,  2.0,  2.0
20190905140059,24,215000622, 37, 13.0, 10.0,  3.0,  5.0
20190905140109,24,232017003, 36, 16.0,  2.0,  3.0,  2.0
20190905140123,24,225983747, 60, 13.0,  2.0,  2.0,  2.0
20190905140146,24,205968710, 36, 11.0,  2.0,  2.0,  2.0
20190905140151,24,224055990, 36, 24.0,  3.0,  8.0,  2.0
20190905140203,24,224076240, 36, 11.0, 10.0,  3.0,  2.0
20190905140203,24,224187000, 56,  0.0,  0.0,  0.0,  0.0
20190905140227,24,211591620, 36, 14.0,  3.0,  1.0,  1.0
20190905140240,24,225980532, 50,  9.0,  3.0,  2.0,  3.0
20190905140247,24,244030462, 36, 16.0,  0.0,  3.0,  2.0
20190905140248,24,224544530, 37,  9.0,  9.0,  2.0,  3.0
20190905140253,24,244100810, 36,  7.0,  4.0,  1.0,  2.0
20190905140256,24,225990007, 36, 10.0,  2.0,  1.0,  3.0
20190905140256,24,225951640, 36, 14.0,  4.0,  1.0,  4.0
20190905140301,24,211347020, 36,  9.0,  0.0,  1.0,  2.0
20190905140316,24,205903430, 36,  0.0,  0.0,  0.0,  0.0
20190905140320,24,235084765, 36, 12.0,  7.0,  2.0,  3.0
```

Fig. 12. Example of decoded ClassB static messages (24A-B)

Pre-Process

The pre-process consists on an initial filtering of outliers based on, first a physical latitude-longitude boundaries of the zone of study and later on the limits already stated in [14] for what it is defined as Non Available Data.

The preprocess of the data is done through *1_PreProcess.ipynb*, see section 8.3, using Jupyter software, a web based interactive python computing platform. Input files are the output of the AIS_decoder.py script plus a decision on which class of AIS data will be filtered. Output files are csv format files containing the headers, examples shown in Fig. 13.

Messages containing all variables with Non Available Data are deleted. At the same time, for computing purposes, Non Available Data is substituted by NaN's aiming to easily create a mask.

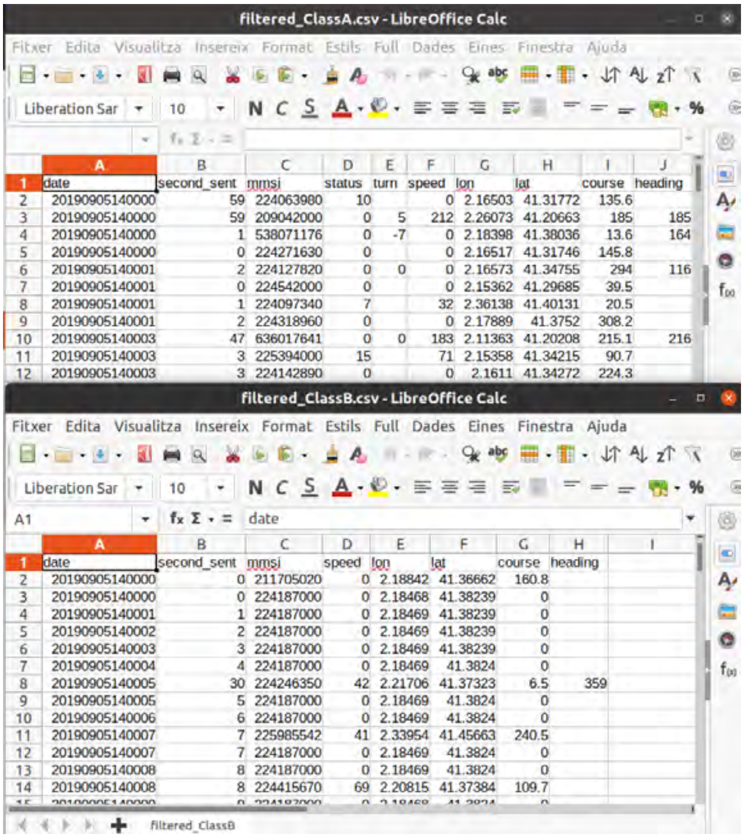


Fig. 13. Example of the messages decoded and filtered

3.3. AIS analysis of port call and maritime traffic evolution

In order to obtain the number of port calls per ship, AIS position reports were filtered to determine whether vessels were located inside the port area. A circle centered in position longitude 002º05.5'E and latitude 41º21.2'N and with a radius r = 4.4 nm was considered, as it represented a circle tangent to both harbour entrances. To avoid false calls introduced by vessels already within the harbour limits at the beginning of the study, all the vessels in port at 00:00 UTC on March 1, 2020 were dropped from the dataset. Thus, we obtained data on specific vessels entering the port, dates and duration of stay in port.

Since the Sea-Web IHS Markit database contains IMO vessel-specific technical information, only merchant vessels were considered from the AIS dataset. The resulting dataset was split into 3 different subsets: (1) passenger vessel, (2) cargo vessel and (3) tanker vessel. AIS messages also contain information about vessel navigational status (namely AIS status). AIS status is manually introduced by the crew and determines emission frequency of AIS messages broadcast by the AIS station. We analyzed the evolution of port calls (vessel stops inside the port) and maritime traffic (vessel movements within a 30 nm range, inside and outside the port) considering AIS status: (a) *Underway* using engine, (b) *At Anchor*, (c) *Not Under Command* and (d) *Moored*.

3.4. Emissions model

Ship emissions were computed independently for each vessel based on vessel operation mode and engine technical data from the IHS Markit database, following the workflow shown in Fig. 14. Emissions were consequently estimated using emission models that strongly depend on vessel operation mode.

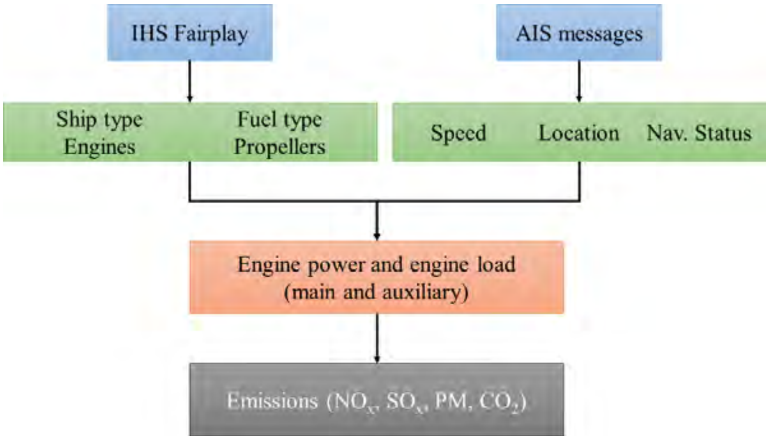


Fig. 14. Workflow of the methodology for the estimation of ship traffic emissions

Power demand, and thereby ship emissions, vary with vessel dynamic behavior and operation mode. Hence, emission models related to shipping within a spatial region typically consider separate emission modes for each vessel, i.e. (1) cruising, (2) manoeuvring and (3) hoteling. Moreover, we included a fourth mode, (4) anchored, to better discretize the impact of anchored vessels.

Assigning the proper emission mode is crucial for accurate pollutant emission estimation. For example, vessels use different fuel types for different manoeuvres and procedures. Even though AIS provides navigational status information, status is a fixed variable in the vessel. AIS status is switched by the ship crew manually, and therefore it is vulnerable to human error and delay. Hence, for every vessel, we selected one of the four emission modes based on vessel speed and location in the port area. Table 3 shows the emission mode used to estimate vessel emissions according to vessel speed and location (inside or outside port premises) and the corresponding navigational status provided by AIS status data.

Table 3. Emission mode according to a) vessel speed and b) location

Emission mode	AIS status	Vessel speed	Vessel location
Cruising	Underway	≥ 1.5 knots	Outside port premises
Manoeuvring	Moored	≥ 0.5 knots	Inside port premises
	Underway	< 1.5 knots	
Hoteling	Moored	< 0.5 knots	Inside port premises
Anchored	At Anchor	< 1.5 knots	Outside port premises

The number of vessels *Not Under Command* was very small, and therefore is not considered in the research.

Emissions were estimated by the Ship Traffic Emission Assessment Model (STEAM) algorithm [11], [16], [17] due to its high accuracy, simplicity and compatibility with AIS data [18] compared to other methods [18][18][18][18][18][18][19].

The STEAM model was initially developed to study vessel emissions in the Baltic Sea. The preliminary model included a methodology to assess fuel consumption and CO_2, SO_2 and NO_X emissions using extensive AIS data and a comprehensive database with all the required technical information on the vessels involved. The model was further enhanced in 2012, the STEAM v.2 algorithm. This version includes PM emissions, fuel consumptions and instantaneous power.

Total emissions from each air pollutant, E_i, were calculated following the Tier I approach [19]:

$$E_i = EF_i \cdot FC_i \qquad \textbf{Eq. (1)}$$

where EF_i and FC_i are the Emission Factors and Total Fuel Consumption, respectively, for the pollutants i = CO_2, SO_2, NO_X, PM.

The Emission Factors (EF) were computed independently for each air pollutant [20]. Three different methodologies were used depending on the air pollutant nature, namely (1) fuel-related for CO_2 and SO_2, (2) engine-related for NO_X and (3) with special considerations for PM. For fuel-related emissions, EFs were obtained based on the engine instantaneous specific fuel consumption and the chemistry of fuel burnt in each emission mode. Information on fuel type, i.e. low sulphur heavy fuel oil (LSHFO), marine gasoil (MGO) and liquefied natural gas (LNG), was readily available in the IHS Markit database for all the vessels involved. For engine-related emissions, EFs were obtained according to the maximum values recommended by the International Maritime Organisation (IMO). As these values are year and engine revolution dependent, the required information was gathered from the IHS Markit database for all the vessels. Only when such data were not readily available, medium-speed engine (500 rpm) was selected as the most representative value [11], [21]. For special considerations (PM), the method developed by [11] was selected because of its accuracy and easy implementation.

Total fuel consumption for each pollutant, FC_i was computed by Eq. (2) based on instantaneous power ($P_{1,2}$) and Specific Fuel Consumption (*SFC)*, which, in turn, depends on engine and fuel type and emission mode [16], [22]:

$$FC_i = \sum_p \left[\Delta t_{1,2} \cdot \sum_e \left(P_{1,2} \cdot SFC_{1,2}^{e,m,p} \right) \right] \qquad \textbf{Eq. (2)}$$

with $\Delta t_{1,2}$ being the time difference between consecutive waypoints (hours), $SFC_{1,2}$ the instantaneous specific fuel consumption (g/kW·h) and $P_{1,2}$ the instantaneous engine power between consecutive waypoints (kW). Sub-indexes e: Engine type, i.e. main engine or auxiliary engine; m: Fuel type, i.e. LSHFO, MGO or LNG; and p: Vessel emission mode, i.e. cruising, anchored, manoeuvring or hoteling.

The methodology to estimate *SFC* was based on the parabolic curve developed by [11], in which the instantaneous specific fuel consumption, $SFC_{1,2}$, is computed through a base and a relative value, as in Eq. (3):

$$SFC_{1,2} = SFC_{base} \cdot SFC_{rel} \qquad \textbf{Eq. (3)}$$

where SFC_{base} is the base specific fuel consumption (g/kW·h) obtained from the IHS Markit database and SFC_{rel} is the relative specific fuel consumption (g/kW·h) based on engine load and computed by Eq. (4) [11] as obtained from assessment of specific fuel consumption curves by engine manufacturers:

$$\mathrm{SFC_{rel}} = 0.455 \cdot \mathrm{EL}_{1,2}^2 - 0.71 \cdot \mathrm{EL}_{1,2} + 1.28 \qquad \textbf{Eq. (4)}$$

where $EL_{1,2}$ is the instantaneous engine load factor (%) taken as

$$EL_{1,2} = EL_{max} \cdot \left(\frac{v_{1,2}}{v_{service}}\right)^3 \qquad \textbf{Eq. (5)}$$

with $v_{1,2}$ being the vessel speed between two points in space and $v_{service}$ the vessel service speed defined in the ship design.

Finally, $P_{1,2}$ in Eq. (2) is calculated by the Propeller Law [22], as shown in Eq. (6):

$$P = k \cdot v^3 \qquad \textbf{Eq. (6)}$$

with P being the generic vessel power (kW), k the power to speed constant (kW·s/m), and v the vessel speed (m/s). The power to speed constant is a parameter that needs to be calibrated. To do so, the actual installed power is considered slightly higher than the actual required power. Then, the maximum engine load factor, EL_{max} , is the ratio between the service and the maximum installed power [16]:

$$EL_{max} = \frac{P_{service}}{P_{installed}} \qquad \textbf{Eq. (7)}$$

where $P_{service}$ and $P_{installed}$ are the service power related to the service speed (kW) and the maximum installed power (kW), respectively. EL_{max} is then related to the Maximum Continuous Rating (MCR) of the engine in %. By combining Eq. (6) and Eq. (7), constant k is easily obtained:

$$k = \frac{EL_{max} \cdot P_{installed}}{v_{service}^3} \qquad \textbf{Eq. (8)}$$

Installed power and service speed are both known from the IHS Markit database. Among all the values in the literature [16], [22], [23], we selected the prevailing value of maximum engine load EL_{max} = 80%.

Using the results for constant k in Eq. (8), the instantaneous power was computed from Eq. (5) based on the vessel actual speed, $v_{1,2}$, provided by AIS messages. As AIS data contained information on vessel speed and position, engine power and engine load were merged into instantaneous power [24]. In [22], distance-over-time speeds are preferred to those provided by AIS messages. However, given the limited size of the location and based on literature research [21], [25], AIS speeds were considered this time as vessel operation is unique to owners and crew. Therefore, using real-time data increased the reliability of the results. For hoteling and anchored emission modes, auxiliary engine power was preferred over main engine output. However, when auxiliary engines were not used, 20% and 10% load factors were considered for main engines, respectively [20], [26].

3.5. Air quality and meteo monitoring

We developed a sensing unit to monitor air quality. It is integrated in a customized electronic board with wireless communication capabilities to send acquired data to the cloud in real-time. See Fig. 15.

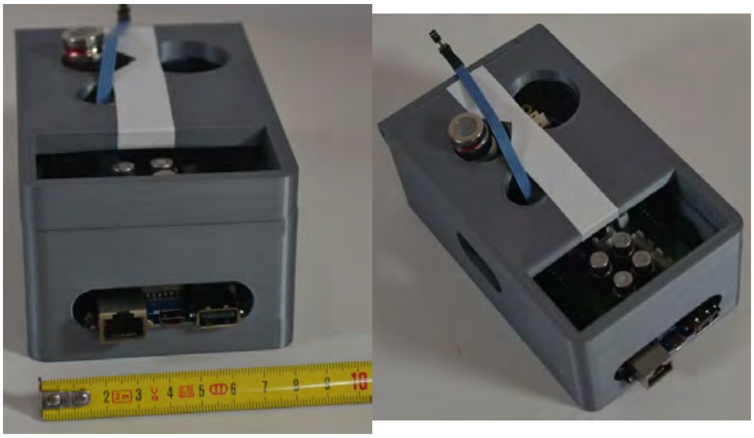


Fig. 15. Sensor unit to monitor air quality and meteo.

The gas sensing system is an heterogeneous sensor array where the sensors are exposed directly to the environment, with no measurement gas cell. The absence of a measurement chamber makes the response time of the system shorter, since the slow dynamics of the chamber are eluded, but, in turn, it also makes the system sensitive to air turbulence present in the vicinity of the sensors.

Specifically, the sensing unit is designed such that it can hold four metal oxide (MOX) gas sensors, two carbon dioxide sensors, a carbon monoxide sensor, and temperature and humidity sensors. MOX gas sensors show broad response to volatiles, although the sensing layer can be adapted to favour the sensitivity to selected gases.

Hence, to enhance the system selectivity and sensitivity, the selected MOX sensors are based on different sensing layers, which are commercially available and provided by Figaro Inc. They operate isothermally, applying a 5V constant voltage on the built-in sensor heater. To enable sensor technology benchmark, the system can interface two carbon dioxide sensors.

Finally, temperature and humidity sensors are also included to monitor environmental conditions and to compensate sensors' cross-sensitivity. The sensor array is integrated with a customized board that communicates with an Arduino Yún platform that features Atheros AR9331 to enable wireless communication.

The microcontroller was programmed to perform:

i) Continuous data acquisition from the sensor array through 10-bit resolution analog-to-digital converters at a sampling rate of 20 s.

ii) Temperature and humidity collection by means of i2c communication protocol.

iii) Data storage in a SD memory card for back-up purposes and data communication through local Wifi network to send most recent data to a data server.

Finally, a custom 3D printed enclosure was designed and implemented for the sensing units. The enclosure provides mechanical protection to the sensing unit while enables direct environment sampling of the sensors.

Table 4. Sensors and target compounds

Sensor and provider	Target
SHT-75, Sensirion	Temperature, humidity
MG811, Hanwei Co.	Carbon dioxide
CozIR-A, Gas Sensing Solutions Co.	Carbon dioxide
CO-B4, Alphasense Co.	Carbon monoxide
TGS 2602, Figaro Inc	VOCs, Ammonia, H2S
TGS 2611, Figaro Inc	VOCs, Methane

4. Results

4.1. Port calls and maritime traffic

This section presents the analysis and results of the evolution of maritime traffic (inside and outside port premises), fuel consumption and the derived emissions inventory of major air pollutants within the area of Barcelona for a 5-month period (March to July 2020). Four different sub-periods were considered, as shown in Fig. 16: Pre-lockdown (March 1-16), Lockdown (March 16-June 22), Home quarantine and strict lockdown (April 6-13) and Post-lockdown (June 22 - July 31). This should shed some light on the impact that political decisions taken during the pandemic might have had on the shipping industry and related emissions.

In the period from March 1 through July 31, 2020, a total of 11,860,409 AIS messages were processed within the 30 nm range.

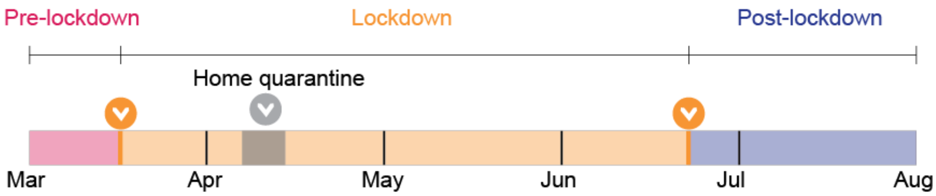


Fig. 16. Period of analysis: 2020. Sub-periods: Pre-lockdown (March 1-16), Lockdown (March 16 - June 22), Home quarantine and strict lockdown (April 6-13), Post-lockdown (June 22 - July 31)

Our methodology to identify port calls using AIS data showed high accuracy (97.1%) when compared to the number of port calls provided by the Port of Barcelona. In particular, during the period March-July 2020, a total of 2734 ship calls were reported in the Port of Barcelona, of which 56.5% corresponded to cargo vessels, 15.0% to tanker vessels and 28.5% to passenger vessels. Total ship calls recorded in the period March-July 2020 dropped to 27.6% compared to those recorded in the same 5-month period during the previous years, see Fig. 17a. This figure illustrates this decrease in total calls and the number of port calls per ship type, with a prominent fall of 49.2% for passenger vessels, followed by cargo vessels (16%) and tanker vessels (1.2%). Fig. 17b plots the average daily port calls for the different ship types and periods (pre-lockdown, lockdown, home quarantine, post-lockdown) during the 5-month period. Differences between pre-lockdown and the 2016-2019 period can be attributed to the impact of China's lockdown at the beginning of 2020. Yet, pre-lockdown values are comparable to those of the 2016-2019 period.

Results in Fig. 17b show a reduction of 27.9% in daily port calls during lockdown, and a more significant drop of 30.1% during home quarantine compared to the pre-lockdown scenario. The post-lockdown period clearly shows signs of recovery, with a lower decrease compared to pre-lockdown 12.4% drop.

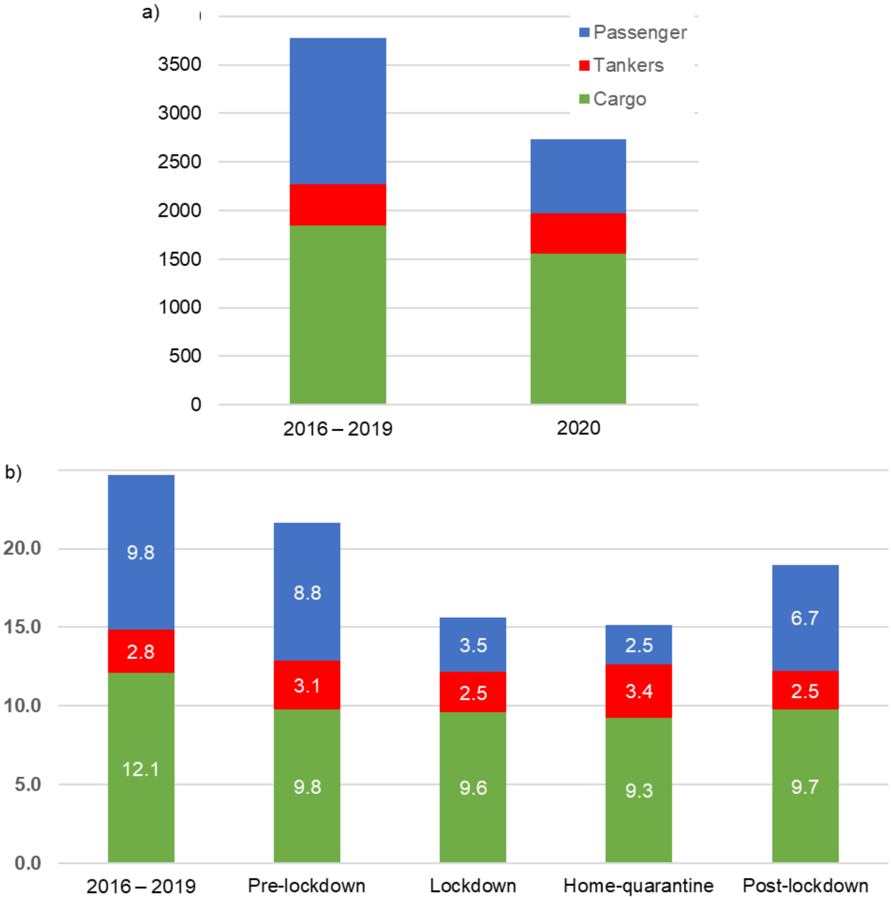


Fig. 17. Number of port calls by ship type during the March-July period. a) Total number of port calls: average for the March-July 2016-2019 period, and 2020. b) average of port calls for the March-July 2016-2019 period and according to analysis periods

Beyond aggregated maritime traffic, instantaneous AIS data enable a closer look at ship activity depending on ship type. During the 5-month period, 1160 different unique vessels were reported by AIS. Daily activity during that period is presented in Fig. 18. The total number of vessels (in black) shows a local minimum on March 16, when lockdown entered into force, followed by a growth until the last week of March. After reaching a second minimum on April 1, this number grew again to achieve maximum values by April 7, followed by a slight reduction over time. The global minimum on June 9 coincides with a weather event, i.e. a storm occurred during the second week of June in the city of Barcelona [27]

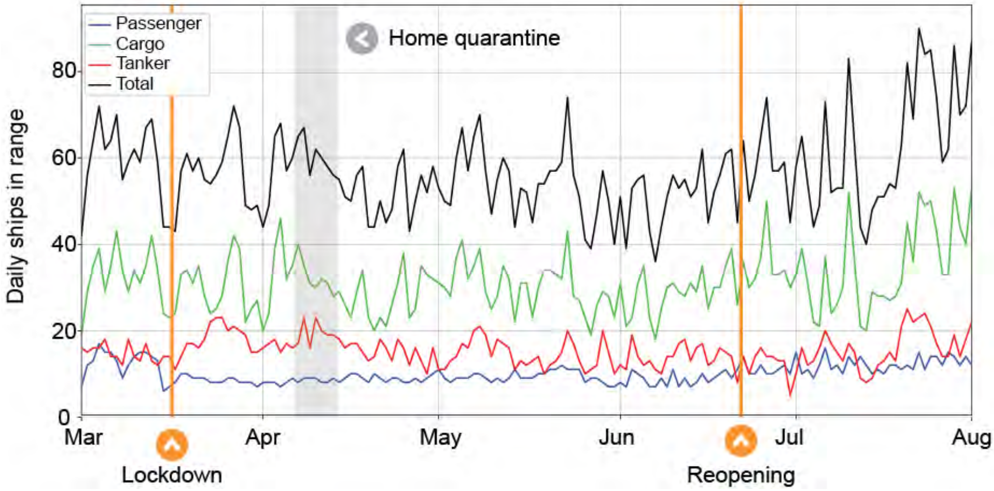


Fig. 18. Daily count of ships in the 30 nm range, detailed by ship type, from March 1 to July 31, 2020

Different trends according to ship type can also be observed in Fig. 18. Both cargo and tanker traffic (~35 and ~18 different vessels/day, respectively) have a higher impact on the daily curve of vessels in the area than passenger vessels (~10 different vessels/day). Overall, passenger vessels represent an average of 17.2% of traffic in the 30 nm range, tanker vessels represent 33.2% and cargo vessels almost half of the total traffic, i.e. 49.6%.

Table 5 lists the daily number of ships in the 30 nm range. Ship activity in the area over the four sub-periods shows an unexpected growth of 2.1% in total traffic during home quarantine compared to the pre-lockdown maritime traffic baseline. A drop of 8.4% during lockdown can also be observed. Maritime activity around the port area clearly increases during post-lockdown, which is in accordance with the recovery in the number of port calls inside port premises observed in Fig. 17.

Table 5. Average daily number of ships in the 30 nm range

	Daily number of ships	Variations from pre-lockdown (%)			
		Cargo	Tanker	Passenger	Total
Pre-lockdown	19.8	-	-	-	-
Lockdown	18.1	-7.2%	4.9%	-27.3%	-8.4%
Home quarantine	20.2	1.7%	29.7%	-29.9%	2.1%
Post-lockdown	20.8	9.4%	3.6%	-3.2%	5.3%

AIS status data revealed the specific actions of the vessels. Fig. 19 shows that most of the vessels were *Underway* (55.6%) while the rest were *Moored* (34.9%) or *At Anchor* (9.2%). The number of vessels *Not Under Command* is merely residual (~0.3%), and is therefore, not considered in the research. As seen in Fig. 19, among all the periods, home quarantine has the lowest percentage of vessels *Underway*, which results in the highest values of *Moored* vessels and vessels *At Anchor*. Pre- and Post- lockdown values are consistent and very similar, which confirms that the recovery started during pre-lockdown.

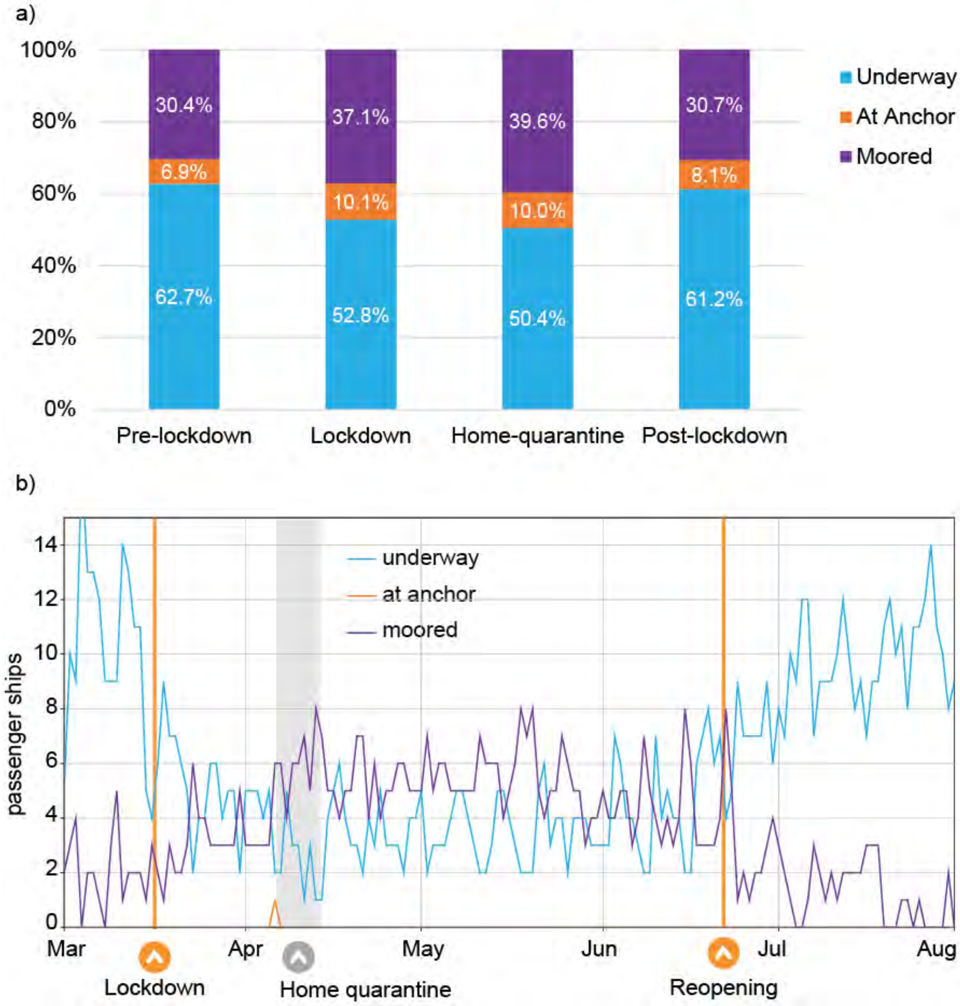


Fig. 19. a) Distribution of AIS status (%) for all vessels in the 30 nm range per period, from March 1 to July 31, 2020; b) Daily distribution of AIS status in passenger vessels

Fig. 19b plots a high variability in AIS status for passenger vessels over lockdown, as opposed to cargo and tanker vessels. Also note that during lockdown and home quarantine, the daily number of *Moored* passenger vessels is higher than that of passenger vessels *Underway*.

Table 6shows the variations in AIS status for the four analysis periods, using the pre-lockdown scenario as the baseline. Passenger vessels *Underway* drop to 46% and 65% during lockdown and home quarantine, respectively, whereas the number of *Moored* vessels is two and three times higher, respectively, than during pre-lockdown. Variations in cargo and tanker vessels are similar, i.e. a decrease in vessels *Underway* and an increase in vessels *At Anchor*, although less prominent than for passenger vessels.

Table 6. Variations in daily AIS status from pre-lockdown per ship type. UW: underway, AA: at anchor, M: moored. Values in %

	Passenger			Cargo			Tanker			Total		
	UW	AA	M	UW	AA	M	UW	AA	M	UW	AA	M
Pre-lockdown	-	-	-	-	-	-	-	-	-	-	-	-
Lockdown	-46	0	225	-2	31	-4	-10	54	-5	-16	47	22
Home quarantine	-65	1	315	-2	49	-8	-11	7	8	-20	45	30
Post-lockdown	3	0	-17	-5	17	6	-4	7	3	-2	18	1

Table 7. Variations in average daily speed

Pre-lockdown	-	-	-	-
Lockdown	7.7%	-2.5%	3.5%	-4.4%
Home quarantine	9.9%	-6.2%	4.3%	-9.1%
Post-lockdown	10.1%	3.5%	7.3%	6.0%

Following the decreasing trend of vessels *Underway* in Fig. 19a, the lockdown period also saw a fall in average speed of all traffics (i.e. -4.4%), especially during home quarantine (i.e. -9.1%), see Table 7. In contrast, average speed for passenger vessels increased, mostly due to the drop in the number of cruise ships, which usually sail at speeds between 15 and 17 knots, whereas ferries (in operation despite lockdown restrictions) sail faster than 19 knots. Likewise, average speed was higher for tanker vessels too. This is due to the spot market operation, by which larger and slower vessels do not operate because of low fuel demand whereas smaller and faster tanker vessels can continue operating, leading to a rise in average speed. The most significant reduction is found for cargo vessels, with average speeds 2.5% and 6.2% lower during lockdown and home quarantine, respectively. This is because cargo vessels operated under slow steaming conditions, as instructed by maritime companies, particularly where they have an allocated berth.

A daily evolution of average speed per ship-type presented in Fig. 20**Error! No s'ha trobat l'origen de la referència.** indicates that the average daily speed dropped during the last two weeks of March and throughout April, slightly recovered by May and reached a higher, more common value by July. The observed trend is consistent with uncertainties related to the economic situation at that time. In fact, this speed reduction is also linked to an increase in the number of vessels during these periods.

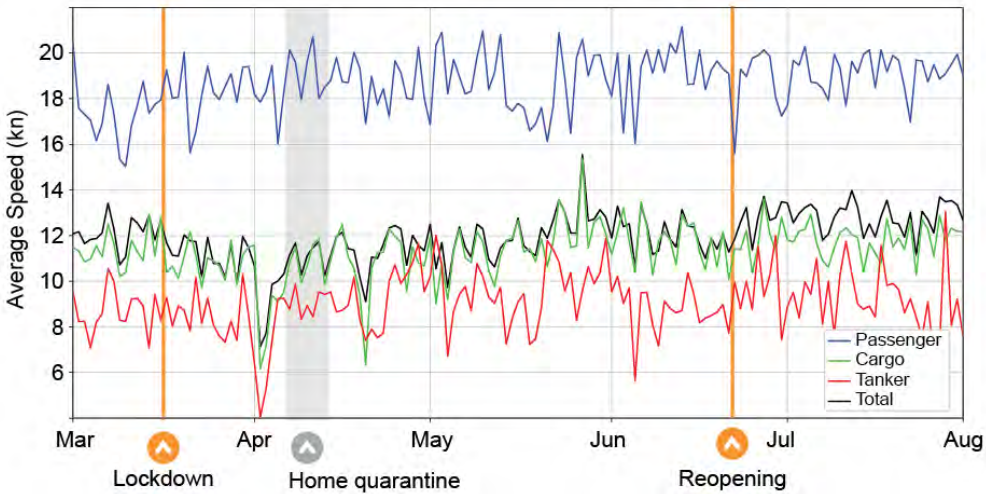


Fig. 20. Average speed for all vessels in the 30 nm range per period, from March 1 to July 31, 2020

Number of port calls, number of unique vessels and average speed show different behaviour during lockdown and during home quarantine. For instance, Table 8 indicates a significant decrease in port calls of passenger vessels during lockdown but a rise in unique vessels. This is mainly due to the fact that passenger vessels maintaining the activity were regular ferry lines whereas cruise ships remained docked in the port area, and therefore they only account for a single call. This is confirmed by the increase in passenger vessel average speed. Cargo vessel average speed is lower but, surprisingly, the dramatic fall of 6.2% during home quarantine is accompanied by a growth in the number of vessels in the area. By contrast, tanker vessel speed increased slightly whereas the number of vessels grew up to 29.7% with a minor growth of less than 1% in tanker vessel calls.

Table 8. Variations in daily port calls, number of vessels and speed during lockdown and home quarantine

	Lockdown			Home quarantine		
	Calls	Vessels	Speed	Calls	Vessels	Speed
Cargo	-0.1%	-7.2%	-2.5%	-0.4%	1.7%	-6.2%
Tanker	-0.6%	4.9%	3.5%	0.9%	29.7%	4.3%
Passenger	-5.3%	-27.3%	7.7%	-1.0%	-29.9%	9.9%
Total	-27.9%	-8.4%	-4.4%	-30.1%	2.1%	-9.1%

4.2. Emission of pollutants

Total fuel consumption depends on vessel type and the emission mode under which the engine operates. Total fuel consumption and emissions from March 2020 to July 2020 were computed using the methodology in section 3.4 and Table 9 shows total values for every pollutant calculated by the STEAM v.2 method.

Table 9. Total values of fuel consumption and emissions in the 30 nm range from March 2020 to July 2020

		Total value (tons)
Fuel consumption		40421
Emissions	CO_2	108603
	SO_2	403
	NOx	2309
	PM	81

Pre-lockdown daily consumptions show a higher average, with a total of 277 tons/day compared to lockdown, with a fuel consumption of 266 tons/day. Interestingly, the highest and lowest values are consecutive, on March 14 and March 16, when lockdown entered into force. Peaks concentrate mostly at the beginning of every month whereas lower consumptions were reported towards the end. A time evolution of fuel consumption during the same 5-month period is presented in Fig. 21. Fig. 21 also plots a steady behavior of fuel consumption, with an average daily value of 275 tons/day during home quarantine.

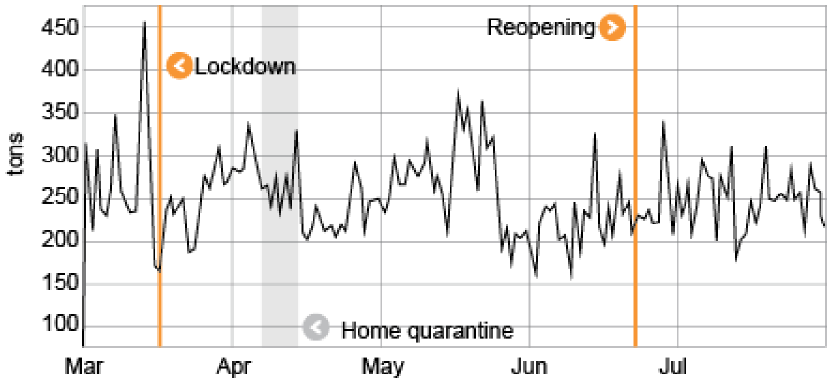


Fig. 21. Daily fuel consumption from March 1 to July 31, 2020

Even though only ~55% of the vessels were *Underway* (Fig. 19), these vessels contribute most to total fuel consumption and related emissions. As for emission modes, total fuel consumption for vessels cruising is dominant (73.6%), followed by hoteling (15%), anchored (7.4%) and manoeuvring (4%), see Fig. 23a. Regarding ship type, passenger vessels account for up to 46.2% of total fuel consumption, followed by cargo vessels (38.1%) and tanker vessels (15.7%), see Fig. 23b. Although passenger vessels were the second largest group in the number of port calls and third in the number of vessels in the 30 nm range, they dominate total fuel consumption. Fig. 22 shows the evolution of the fuel consumption per type of fuel. Types of fuel were defined by IHS-Markit database. As detailed in the methods section, auxiliary engines were assumed to consume MGO, but . Fig. 22 does not distinguish between engines.

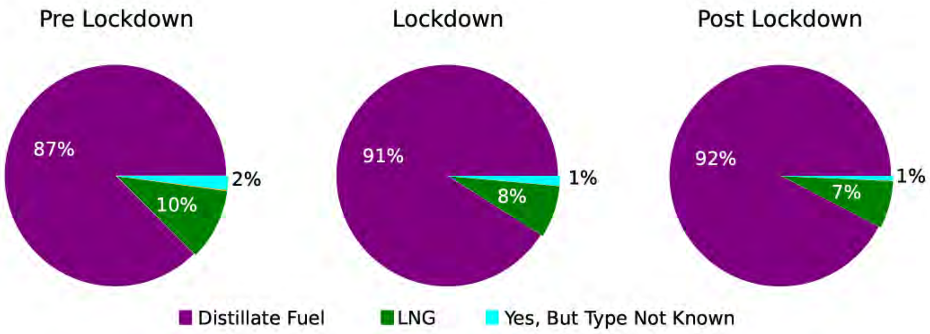


Fig. 22. Fuel consumption in the different phases, depending on the type of fuel

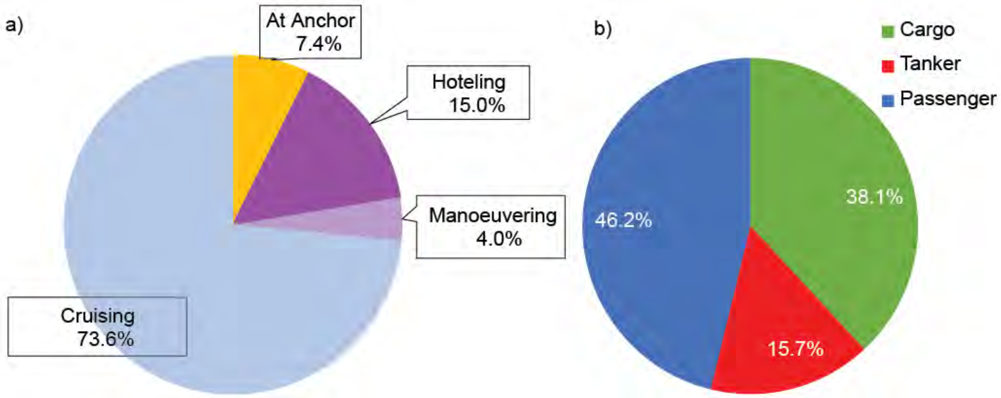


Fig. 23. Fuel consumption from March 1 to July 31, 2020 according to a) emission mode and b) ship type

Fig. 24 compares the daily evolution of fuel consumption, average speed, and number of ships in the 30 nm range. Significant changes occur during home quarantine (Fig. 24a), with sharp gradients in the number of ships in the 30 nm range, whereas average speed and fuel consumption remain fairly constant. Also, during the same sub-period, the total daily number of vessels *Underway* and *Moored* decreases whereas *At Anchor* vessels increase to reach similar levels (Fig. 24b).

(FNB-UPC)

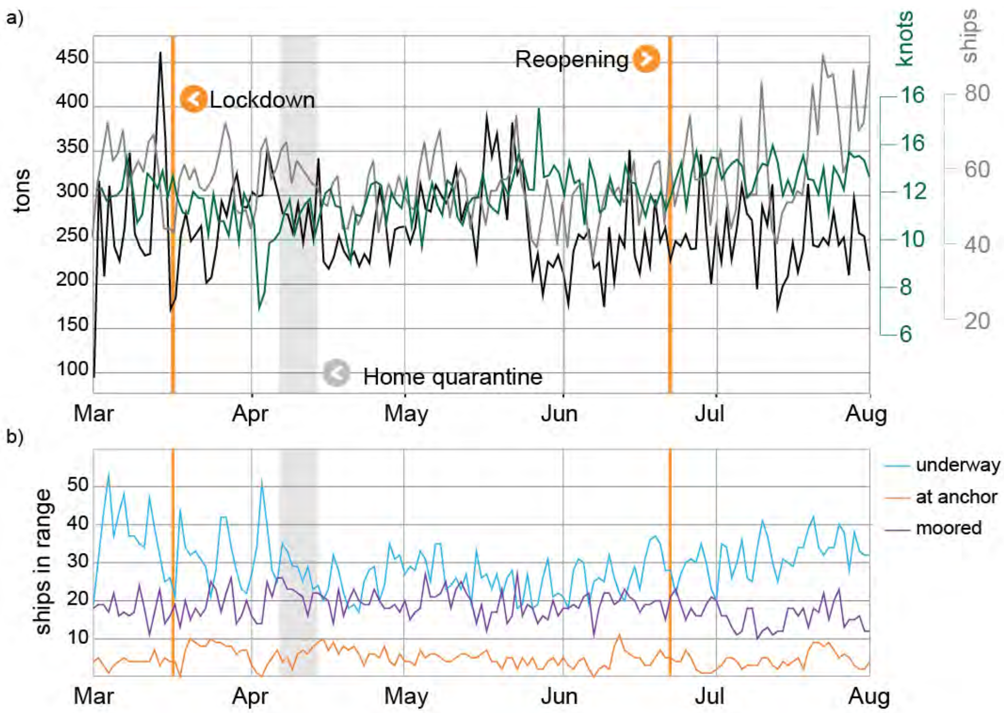


Fig. 24. Daily evolution of a) fuel consumption (black), average speed (dark green) and number of unique ships in the 30 nm range (gray) and b) AIS status

CO_2, SO_2, NO_x and PM emissions were estimated using Eq. (1) and daily evolution of these pollutants is shown in Fig. 25. Trends for pollutant emissions are similar to those for fuel consumption in terms of distribution per ship type. This can be observed in **Error! No s'ha trobat l'origen de la referència.**a, where fuel consumption and emissions generated by cargo vessels are ~40%, tanker vessels produce ~15% of emissions and consume the same percentage of fuel, and passenger vessels are responsible for ~45% of emissions. However, several differences are observed when looking at the variations for pollutants and fuel consumption over the sub-periods in **Error! No s'ha trobat l'origen de la referència.**b. Whereas fuel consumption and CO_2 emissions fall with a small increase in NOx during lockdown, during home quarantine fuel consumption decreases slightly but pollutant emissions increase substantially. Interestingly, the post-lockdown scenario turns out to show the best results in terms of fuel consumption and pollutant emissions.

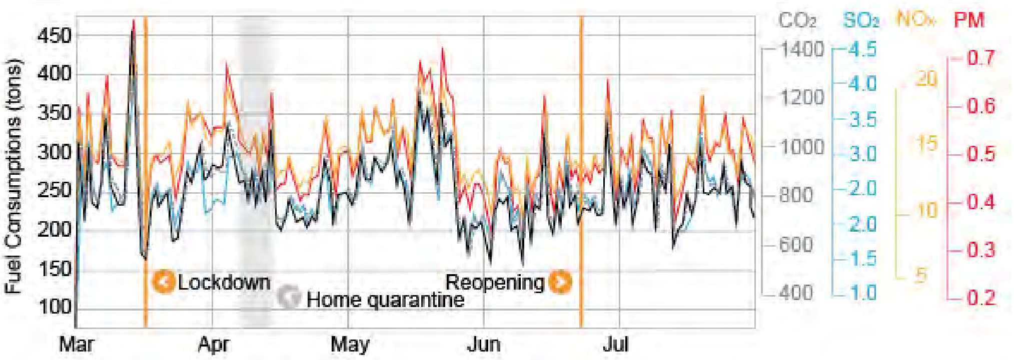


Fig. 25. Daily variations in CO_2 (dashed gray), SO_2 (blue), NOx (orange) and PM (red) emissions from March 1 to July 31, 2020

Table 10. a) Average distribution and b) evolution of fuel consumption (FC) and emissions per ship type

a)	**FC**	**CO_2**	**SO_2**	**NOx**	**PM**
Cargo	38.1%	38.7%	41.0%	42.2%	39.0%
Tanker	15.7%	16.5%	15.3%	14.9%	15.2%
Passenger	46.2%	44.2%	43.6%	42.7%	45.8%

b)	FC	CO_2	SO_2	NOx	PM
Pre-lockdown	-	-	-	-	-
Lockdown	-4.0%	-1.8%	0.0%	1.3%	0.0%
Home quarantine	-0.7%	3.0%	3.7%	7.3%	3.8%
Post-lockdown	-9.4%	-8.0%	-7.4%	-4.0%	-3.8%

CO_2 emission rates are mostly related to cruising (72%) whereas hoteling accounts for 16.6%, 1.6 higher than fuel consumption (Fig. 23a). This is due to the use of MGO during hoteling and LSHFO during cruising, with the former having a slightly higher quantity of carbon content.

Changes in contribution to total SO_2 emissions per emission mode were small when compared to fuel consumption. This is partly due to the fact that both LSHFO and MGO have by default similar sulphur contents. Distribution by ship type is quite similar to the other fuel-related emissions and fuel consumption. It is worth noting that the slight fall in tanker and passenger vessel emissions was related to the fact that some of these vessels are powered by LNG, with residual sulphur content leading to an overall reduced contribution of these types of vessel to total SO_2 emissions.

NOx emissions show a similar trend to that of fuel consumption, with some differences in the magnitude of peaks in Table 9 because NOx emissions are not fuel but engine related. This type of emissions is heavily dependent on engine characteristics and revolutions. This is the main reason why cruising-related emissions are higher than in previous cases, contributing 75.8% compared to 14.6% for hoteling, 6.1% for anchored, and 3.5% for manoeuvring mode. These differences are due to the engine configuration introduced in the model, as preference was given to auxiliary engines with higher revolutions and lower emission factors during hoteling compared to main engines operating at their fullest during cruising. The same applies to cargo and tanker vessels. These have greater contributions to NOx emissions than the more modern passenger vessels which, in compliance with Tier II requirements, typically operate with medium-speed engines with lower NOx emission factors than their counterpart.

In relation to PM emissions, variability between days was mostly noticed during the most restrictive days, Fig. 25. Since PM emissions are both fuel and engine related, very similar mode trends to those of NOx emissions were found.

Fig. 26 compares the traffic values obtained from AIS data with fuel consumption as a function of ship type. It can be observed that the contribution of each ship type is dissimilar to the distribution of vessels in the 30 nm range during the 5-month period. Surprisingly, passenger vessels, which only represent an average of 17.2% of the traffic in the 30 nm range, have the highest fuel consumption and are responsible for the highest emissions, with a contribution of 46.2%. On the other hand, cargo vessels represent almost half of the total traffic (49.6%) and account for 38.1% of total emissions and fuel consumption. At the lower end, tanker vessels represent 33.2% of total traffic but account for ~15% of emissions and fuel consumption only.

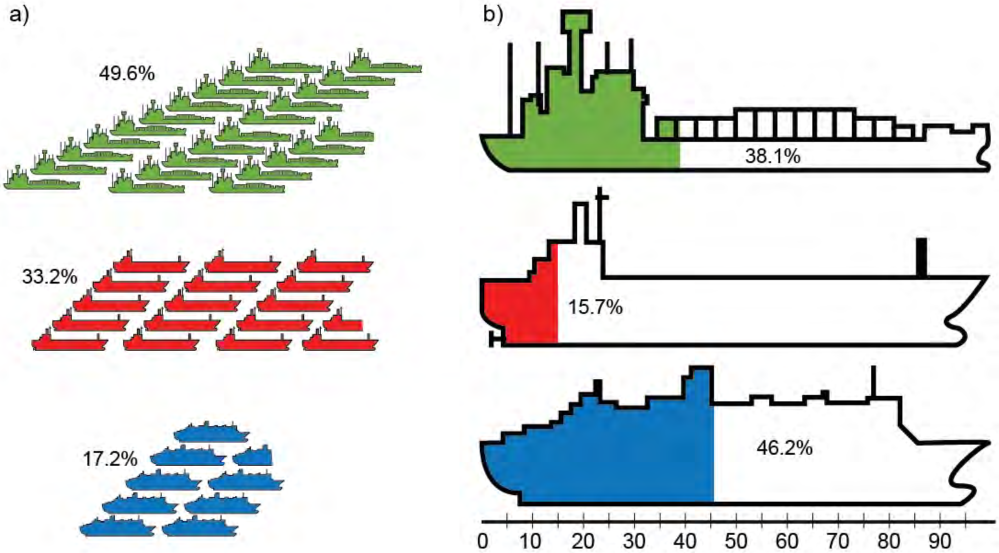


Fig. 26. Average distribution (%) of a) maritime traffic and b) fuel consumption per ship type (top: cargo, middle: tanker, bottom: passenger)

Analysis of the navigational status of maritime traffic within the port hinterland and fuel consumption (and therefore, pollutant emission) contribution shows that ships *Underway* represent ~55% of traffic while producing up to 70% of emissions (seeFig. 27). In contrast, *Moored* vessels represent ~34.9% of all vessels but have an emission contribution of less than 20%.

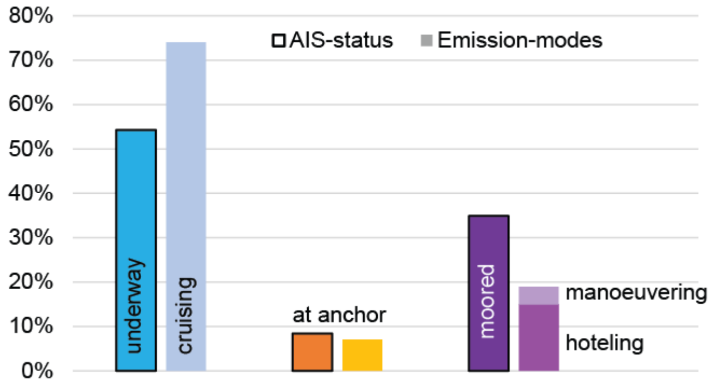


Fig. 27. Average distribution (%) of traffic maritime using AIS status -dark border- and fuel consumption per emission mode -no border, data from Fig. 26a

4.3. Air pollutant measurements

Measurements in different areas of the city cannot be directly linked to air pollutant emissions from vessels due to different reasons. However, Fig. 28 shows some correlation at first sight before the lockdown and in mid-May with an increase in emissions produced by maritime traffic and air quality measurements in different neighbours.

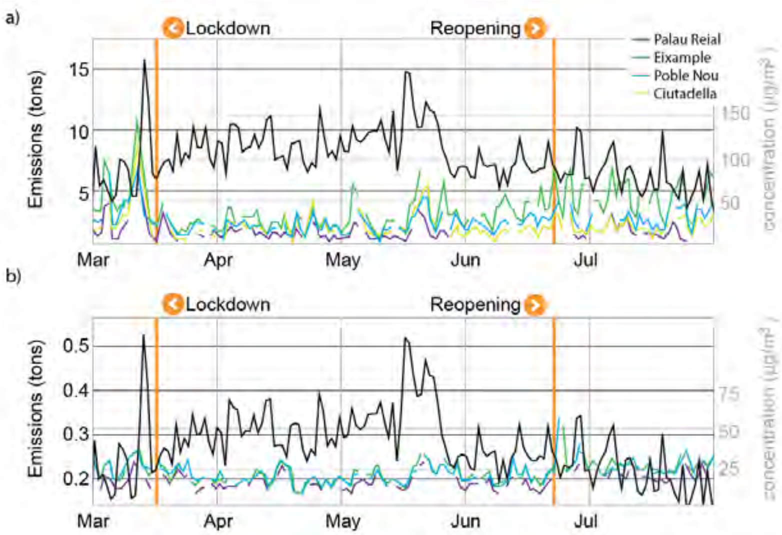


Fig. 28. Estimated emissions from the observed maritime traffic and the emission model -black lines a)NO_x, b)PM-. Pollutant emissions measured at four different air-quality stations are also included (Ciutadella and Poble Nou are closer to the port of Barcelona)

Air pollutant measurements from previous years indicate that there was a significant drop in the measured pollutants Fig. 29-left in the city of Barcelona. This result can be linked to the prominent decrease in the wheeled traffic Fig. 29-right.

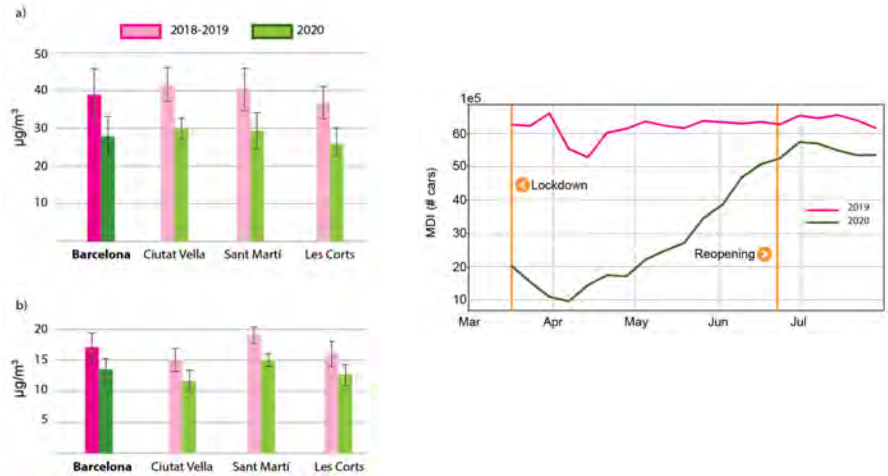


Fig. 29. Left: Year-averaged emissions in four districts of the city of Barcelona. a) NO_2, b) PM. Ciutat Vella is closer to the port. Source: Consorci Sanitari de Barcelona; right) Evolution of the Mean Daily Intensity (MDI) of wheeled traffic in the area of Great Barcelona, in total number of cars. Source: Autoritat del Transport Metropolità

Wheeled traffic decreases significantly and therefore the decrease in NOx and PM in Barcelona can only be attributed to wheeled traffic.

4.4. Air quality measurements

Initial results on the sensor unit are shown in Fig. 30, with CO2 measurements within the laboratory. During the process of the sensor development, some interesting results have come out regarding the use of low-cost-sensor units: in order to be suited within the vessel environment, an air diluting mechanism is needed in order to lower down the concentration of the gas.

Otherwise, if these sensors are to be used onboard, it is advisable that they are located far enough of the engine exhaust. The initial idea within the project is to locate four sensors in the extremes of the vessel and measure wind speed and direction at the same time to account for its influence on the measurements in each point.

The rest of the sensors are under development and calibration results will be shared with the IAMU community in the near future.

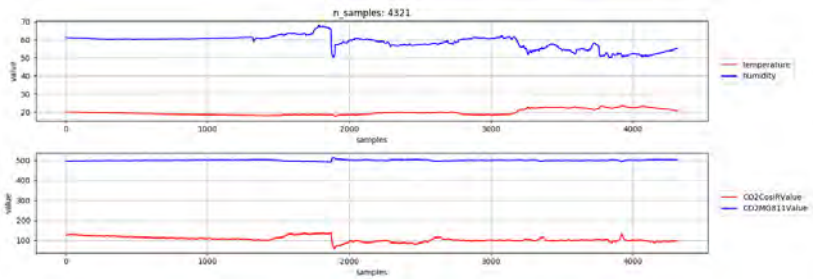


Fig. 30. CO2 sensor measurements after filtering the signal.

5. Discussion

The research presented within this final report yields interesting results regarding the impact of the COVID19 pandemic in the maritime traffic and their correspondent pollutant emissions in the vicinity of the Port of Barcelona. Prior to the discussion of the results obtained herein, authors and reviewers think it is worth discussing about the methods and their limitations.

The source of the data used is and AIS antenna located at the roof of the Barcelona School of Nautical Studies' building. AIS data was originally designed to improve the safety of vessels in the sea, but has now turned into an interesting source of data for scientists regardless of its different sources of errors: the first source is the existence of *non-available-data* which is described in [14] and accounted for in the decoding process, the second is the inherent error of the gps (~8 m, although an initial study, [15], showed differences in x and y, with y having a mean error of ~10m). However, the main issue concerning AIS data is the missing values and the potential outliers in most of the variables. In the present study, positional outliers are captured by means of geographical limits, but no in-detail analysis is carried out in each trip to detect individual outliers. The speed, which is used to compute the emissions, is not filtered either and, although it can potentially be a source of error, no spike has been found. There is another common problem with AIS data, which is the presence of significant gaps -although the data is very closed to an urban area with minimum problems on coverage. This would potentially indicate a sub-estimation of the results, in particular with *underway* vessels, with segments of a track not being recorded. However, the formulas to compute the fuel consumption do consider the time difference between points and the distance considered (~30nm) is small enough minimizing this error. Further work is being done nowadays to detect outliers and accurately reconstruct trajectories with significant gaps.

In terms of the methodology used to compute the fuel consumption [11,16,17] already discuss about the fairness of the STEAM methodology by comparing it with top-down methodologies where the fuel consumption is known from public bodies. The present research did not have these sources of data available and, therefore, STEAM method was chosen. There is, although, an important consideration on the STEAM method within the harbour areas. STEAM method works well with large-distance trips where the engine regime does not change significantly. However, during in-port manoeuvres, changes in the engine regime have a clear impact and may affect the real fuel consumption. Thus, authors suspect that the STEAM method underestimates the real fuel consumption within harbour areas and further research is needed to focus in these specific areas.

Results show that, during lockdown the number of vessels dropped by 8.4% compared to the pre-lockdown scenario, with a rise in tanker vessels of 4.9%, see Table 5. However, during home quarantine the traffic increase was slightly above 2%, with an increase in cargo and tanker vessels between 1.7% and 29.7%, respectively. Moreover, as seen in Table 8, these vessels remained within the area for a longer time, which can be demonstrated by the fact that i) the number of vessels *At Anchor*, drifting for orders or *Moored* in the port of Barcelona increased, see Fig. 19 and Table 6; ii) the overall speed of the vessels decreased because of cargo vessels following slow-steaming orders, see Fig. 20 and Table 7, and iii) the number of vessels returning to their homeport increased as global economy was being shut down. It is worth noticing that at the beginning of March fuel prices fell by 65%, reaching negative values by April 20, 2020. Reduced global fuel demand and impossibility to completely stop fuel extraction resulted in larger vessels turning into floating oil storage units drifting at sea waiting for oil buyers. Therefore, the reduction in speed is consistent with the global situation.

Differences between port calls and ships in the area (especially cargo and tanker vessels Table 8) are due to the maritime traffic parallel to the coast that does not call at the Port of Barcelona. However, in general, the increase in the number of vessels was partly related to a higher number of vessels with static or quasi-static positions, i.e. *Moored* and *At Anchor* during lockdown, especially during home quarantine, see Fig. 19a. In fact, it is not that more vessels arrived but rather that vessels did not leave, increasing their time in port or in the anchorage area. Moreover, vessels *At Anchor* do not account for port calls since they are outside the port area.

The growth in the number of vessels during home quarantine is due to the vessels *At Anchor*. On the contrary, the number of *Moored* vessels decreased, which can be explained by uncertainties related to berth availability, port operations and mandatory quarantines for crew before operations. Decreased trade resulted in a fall in port operations compared to previous years. Hence, ship managers ordered vessels to adopt slow steaming practices and adapt their schedules to just-in-time arrivals. Over home quarantine, emissions exceeded pre-lockdown values, Table 10b. In the same period, the number of tanker and cargo vessels increased by 24.8% and 9%, respectively in Table 8, with most being *Underway*. Therefore, emissions during home quarantine come from maritime traffic dedicated to the trading of goods.

Although there is the preconceived idea of an apparent relationship between pollutant emissions, fuel consumption and number of vessels in the area, several other factors play an important role in the final pollutant emission values. Results indicate that maritime emissions did not decrease according to the observed decline in maritime traffic within the 30 nm range. A clear example is the mismatch between the higher number of vessels (2.1%) and associated fuel consumption (-4.0%) and emissions (between 3-7.3%) during home quarantine. The issue can be traced back to Fig. 24, as the average speed of vessels decreased by 9.1%, the number of vessels *Underway and Moored* dropped too, and the number of *At Anchor* vessels increased. This is even more obvious during the lockdown where the number of vessels decreased by 8.4% whereas emissions did not decrease as much (CO_2 showed a reduction of 1.8%) and some pollutants also increased (NO_x increased by 1.3%). Hence, the change in AIS status also had an impact on global values.

The average distribution of fuel consumption per ship type is of the order of that of emissions per ship type for all four pollutants, see Fig. 25. This is, cargo vessels consume ~40% of the total fuel and produce the same amount of pollutants with very few differences between them. Likewise, tanker vessels consume ~15% of fuel and are the cause of the same trend of emissions. Finally, passenger vessels consume ~45% of fuel and produce the same percentage of pollutants.

The growth in NOx emissions during lockdown -i.e.1.3%- can be associated with the higher number of tanker vessels in the area. Although passenger vessel traffic also increased, this did not contribute to the growth because most of it was *Moored* inside the port, and *Moored* passenger vessels use auxiliary engines, which produce slightly lower NOx emissions.

The scenario presented in Fig. 26 is clearly related to the fact that passenger vessels operate at high load constantly and sail at very high speeds. Their shipboard services have a high power demand, which explains their higher installed power. It is then clear that, despite the lower number of this type of vessels, they are a major pollutant in the area, and their contribution was related to the increase in average speed. There is a strong correlation between vessel operation mode and overall contribution to pollution, as vessels trading at lower speeds are in fact more environmentally friendly, as plotted in Fig. 27. This is in accordance with the fact that greener shipping can only be accomplished by reducing average vessel speeds and, of course, rethinking passenger vessel operations, as they were responsible for almost half of the emissions despite accounting for only 17.2% of the traffic in the area.

Post-lockdown fuel consumption and pollutant emission values are significantly lower (~8%) than pre-lockdown ones due to not only the increase in passenger vessels *Underway* but also important decreases in cargo and tanker vessels *Underway*. Although passenger vessels are responsible for most of the emissions, the rise in the number of this type of vessels was not significant enough to return to pre-lockdown pollutant emission values due to the decrease in the number of cargo and tanker vessels *Underway*.

Emissions decrease in the city of Barcelona but our model suggests that emissions due to maritime traffic do not decrease. Wheeled traffic decreases significantly and therefore the decrease in NOx and PM in Barcelona can only be attributed to wheeled traffic. Therefore, pollution in the area of Barcelona does not come from the Port.

6. Conclusions

The COVID-19 pandemic had an impact on maritime traffic and related emissions owing to changes in maritime traffic. Pre- and post-lockdown maritime traffic values show a clear mismatch, as recovery from the initial economic downfall was not the same for all ship types.

Regarding maritime traffic in the 30 nm range, the pandemic brought a slight decrease of 8.4% in the total number of reported vessels in the area compared to pre-lockdown, with a surprising increase of 2.1% during home quarantine. However, this change did not match the decrease in the number of port calls in Barcelona (27.9%). The difference is due to the way vessels operated, i.e. average speeds reduced by 4.4% and increased number *At Anchor* and *Moored* vessels. It is not that more vessels were reported, but that the ones that were already in the 30 nm range stayed over for longer periods. Cargo and tanker vessels managed to weather the situation by adjusting capacity to real-time demand and showed early signs of recovery as of July 31, 2020. However, passenger vessels succumbed badly to travel restrictions due to the pandemic, and although ferry traffic was resumed when reopening was allowed, ongoing uncertainties related to a still spreading virus do not forecast a smooth second semester for the business.

Fuel consumption and fuel-related and engine-related emission values were, on average, higher during lockdown owing to an increased number of vessels in the area. Although fuel consumption decreased by 4%, NOx emissions rose because of the different operation modes. This is even clearer during home quarantine, when fuel consumption values were almost the same as pre-lockdown ones, but there were increases of 3.0% in CO_2, 3.7% in SO_2, 7.3% in NOx and 3.8% in PM. The rise in SO_2 was associated with a lower consumption of LNG driven by a higher use of auxiliary engines while vessels were *Moored* and *At Anchor.* It is worth noting that the growth in fuel consumption and emissions during early lockdown and home quarantine was well below the rise in the number of vessels as a result of reduced speeds and an increased number of *At Anchor* and *Moored* vessels. Distribution of ship types in the 30 nm range does not correlate that of fuel consumption and emissions. In fact, passenger vessels were surprisingly responsible for more than 40% of fuel consumption and emissions, but represented only 17.2% of all vessels. The reason lies in higher installed outputs arising from higher power demands to sustain shipboard services and higher-than-average trading speeds. This raises the current ongoing discussion about sustainability of passenger business other than ferry crossings.

Air quality results indicate that the contribution of maritime traffic at the city of Barcelona is not as important as the contribution of wheeled traffic and industrial activity. Although no industrial activity data was provided, wheeled traffic significantly decreased with the covid mobility related restrictions in parallel to the concentration of air pollutants.

7. References

[1] WHO, "Novel Coronavirus – China," *World Health Organization, disease uotbreak news*, 2020. .

[2] G. F. Ficetola and D. Rubolini, "Containment measures limit environmental effects on COVID-19 early outbreak dynamics," *Sci. Total Environ.*, vol. 761, p. 144432, 2021, doi: 10.1016/j.scitotenv.2020.144432.

[3] UNCTAD, *Review of Maritime Transport 2020*. 2020.

[4] C. Doumbia-Henry, "Shipping and COVID-19: protecting seafarers as frontline workers," *WMU J. Marit. Aff.*, vol. 19, no. 3, pp. 279–293, 2020, doi: 10.1007/s13437-020-00217-9.

[5] Instituto de Salud Carlos III, "Distribución geográfica del COVID en España," *Centro Nacional de Epidemiología*, 2020. .

[6] A. Tobías *et al.*, "Changes in air quality during the lockdown in Barcelona (Spain) one month into the SARS-CoV-2 epidemic," *Sci. Total Environ.*, vol. 726, 2020, doi: 10.1016/j.scitotenv.2020.138540.

[7] I. Ruiz-Guerra, V. Molina-Moreno, F. J. Cortés-García, and P. Núñez-Cacho, "Prediction of the impact on air quality of the cities receiving cruise tourism: the case of the Port of Barcelona," *Heliyon*, vol. 5, no. 3, 2019, doi: 10.1016/j.heliyon.2019.e01280.

[8] M. Gonzalez-Aregall and R. Bergqvist, "Green port initiatives for a more sustainable port-city interaction: The case study of Barcelona," in *Maritime Transport and Regional Sustainability*, 2019.

[9] J. M. Baldasano, "COVID-19 lockdown effects on air quality by NO2 in the cities of Barcelona and Madrid (Spain)," *Sci. Total Environ.*, vol. 741, no. 2, 2020, doi: 10.1016/j.scitotenv.2020.140353.

[10] IHS Markit, "Sea-web database," 2020. .

[11] J. P. Jalkanen, L. Johansson, J. Kukkonen, A. Brink, J. Kalli, and T. Stipa, "Extension of an assessment model of ship traffic exhaust emissions for particulate matter and carbon monoxide," *Atmos. Chem. Phys.*, vol. 12, no. 5, pp. 2641–2659, 2012, doi: 10.5194/acp-12-2641-2012.

[12] A. Mujal-Colilles, J. N. Guarasa, J. Fonollosa, T. Llull, and M. Castells-Sanabra, "COVID-19 impact on maritime traffic and corresponding pollutant emissions. The case of the Port of Barcelona," *J. Environ. Manage.*, vol. 310, p. 114787, May 2022, doi: 10.1016/j.jenvman.2022.114787.

[13] J. Fonollosa, D. Martín-López, J. Nieto, M. Castells-Sanabra, and A. Mujal-Colilles, "Estimation of vessel emissions and contribution to overall pollution in port-cities," 2022.

[14] ITU-R, "Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band," 2014.

[15] A. Mujal-Colilles, C. Bagés, and J. Fonollosa, "Maneuvering and operational strategies using AIS data," in *Developments in Maritime Technology and Engineering*, 2021, pp. 151–154.

[16] J.-P. Jalkanen, A. Brink, J. Kalli, H. Pettersson, J. Kukkonen, and T. Stipa, "A modelling system for the exhaust emissions of marine traffic and its application in the Baltic Sea area," *Atmos. Chem. Phys.*, vol. 9, no. 23, pp. 9209–9223, Dec. 2009, doi: 10.5194/acp-9-9209-2009.

[17] L. Johansson, J. P. Jalkanen, and J. Kukkonen, "Global assessment of shipping emissions in 2015 on a high spatial and temporal resolution," *Atmos. Environ.*, 2017, doi: 10.1016/j.atmosenv.2017.08.042.

[18] M. Castells-Sanabra, C. Borén, R. van der Meer, J. Torralbo, and S. Ordás, "Existing emission

calculation methods applied to monitoring, reporting and verification (Mrv) on board," *Nase More*, vol. 67, no. 2, pp. 163–171, 2020, doi: 10.17818/NM/2020/2.9.

[19] C. Trozzi, "EMEP/EEA air pollutant emission inventory guidebook - 2013: 1.A.3.d.i, 1.A.3.d.ii, 1.A.4.c.iii International navigation, national navigation, national fishing," 2013.

[20] J. P. Jalkanen, "A modelling system for the exhaust emissions of marine traffic and its application in the Baltic Sea area. Suplement 1: The procedure for calculating the SO x emission factor from fuel sulphur content is given below . The units are given in parenthesis .," vol. 85, no. C, pp. 1–2, 2009, doi: 10.1086/599017.

[21] X. Sun, Z. Tian, R. Malekian, and Z. Li, "Estimation of vessel emissions inventory in Qingdao port based on big data analysis," *Symmetry (Basel).*, 2018, doi: 10.3390/sym10100452.

[22] L. Goldsworthy and B. Goldsworthy, "Modelling of ship engine exhaust emissions in ports and extensive coastal waters based on terrestrial AIS data - An Australian case study," *Environ. Model. Softw.*, vol. 63, pp. 45–60, Jan. 2015, doi: 10.1016/j.envsoft.2014.09.009.

[23] J. Coello, I. Williams, D. A. Hudson, and S. Kemp, "An AIS-based approach to calculate atmospheric emissions from the UK fishing fleet," *Atmos. Environ.*, vol. 114, pp. 1–7, Aug. 2015, doi: 10.1016/j.atmosenv.2015.05.011.

[24] J.-P. Jalkanen, A. Brink, J. Kalli, H. Pettersson, J. Kukkonen, and T. Stipa, "Atmospheric Chemistry and Physics A modelling system for the exhaust emissions of marine traffic and its application in the Baltic Sea area," 2009. Accessed: Aug. 20, 2021. [Online]. Available: www.atmos-chem-phys.net/9/9209/2009/.

[25] T. K. Liu, Y. S. Chen, and Y. T. Chen, "Utilization of vessel automatic identification system (AIS) to estimate the emission of air pollutant from merchant vessels in the port of kaohsiung," *Aerosol Air Qual. Res.*, vol. 19, no. 10, pp. 2341–2351, Oct. 2019, doi: 10.4209/aaqr.2019.07.0355.

[26] J. N. Guarasa, S. Ordás, J. Francesc, and X. Martínez De Osés, "Study on the impact of cruise ships calling at Barcelona in the city air quality Bachelor's Thesis," Universitat Politècnica de Catalunya, 2017. Accessed: Dec. 04, 2020. [Online]. Available: https://upcommons.upc.edu/handle/2117/107741.

[27] Observatori Fabra, "Monthly summaries 2020." .

8. Scripts

8.1. Functions

8.1.1. AIS functions

```python
import pyAISm
from math import cos, radians
import pandas as pd
import numpy as np


typeAna=[1,2,3,5,18,24]
i123=0
i5=0
i18=0
i24A=0
i24B=0

def fer_tira_123(data):
   #serveix pels tipus 1,2,3
   Lesp=['type','mmsi','status','turn', 'speed','lon','lat','course','heading']
   Lval=[]
   s=data['second'] # els segons els posem davant per poder afegir la rersta de time
   a='{}'.format(s)
   if len(a)==1:
      a='0'+a

   Lval.append(a)
   for esp in Lesp:
      Lval.append(data[esp])
   st=',{:2s},{},{},{:2d},{:6.1f},{:6.1f},{:.5f},{:.5f},{:5.1f},{:3d}\n'.format(Lval[0],Lval[1
   return st

def fer_tira_5(data):
   #serveix pels tipus 5
   Lesp=['type','mmsi','imo','shipname', 'shiptype','to_bow','to_stern', 'to_port','to_starboa
#   Lesp=['type','mmsi','shipname', 'shiptype','to_bow','to_stern', 'to_port','to_starboard',
   Lval=[]
   for esp in Lesp:
      Lval.append(data[esp])
      if (esp =="destination"):
          if ("," in Lval[10]):
              Lval[10]= Lval[10].replace(",","")
   st='{},{},{},{:20s},{:3d},{:5.1f},{:5.1f},{:5.1f},{:5.1f},{:5.1f},{}\n'.format(Lval[0],Lval
#   st='{},{},{:20s},{:3d},{:5.1f},{:5.1f},{:5.1f},{:5.1f},{:5.1f}\n'.format(Lval[0],Lval[1],L
   return st

def fer_tira_18(data):
   #serveix pels tipus 18
   Lesp=['type','mmsi', 'speed','lon','lat','course','heading']
   Lval=[]
   s=data['second'] # els segons els posem davant per poder afegir la rersta de time
   a='{}'.format(s)
   if len(a)==1:
      a='0'+a
   Lval.append(a)
   for esp in Lesp:
      Lval.append(data[esp])
   st=',{:2s},{},{},{:6.1f},{:9.5f},{:9.5f},{:5.1f},{:3d}\n'.format(Lval[0],Lval[1],Lval[2],Lva
   return st

def fer_tira_24A(data):
   #serveix pels tipus 24A
   Lesp=['type','mmsi', 'shipname']
   Lval=[]
   for esp in Lesp:
       if esp in data: Lval.append(data[esp])
       else: Lval.append('NoName')
   st='{},{},{:20s}\n'.format(Lval[0],Lval[1],Lval[2])
```

```
    return st

def fer_tira_24B(data):
    if not( pyAISm.is_auxiliary_craft(data['mmsi'])):
        Lesp=['type','mmsi', 'shiptype','to_bow','to_stern', 'to_port','to_starboard']
        Lval=[]
        for esp in Lesp:
           # Lval.append(data[esp])
             if esp in data: Lval.append(data[esp])
             else: Lval.append(0)
#        st='{},{},{:3d},{:5.1f},{:5.1f},{:5.1f},{:5.1f},{:1s}\n'.format(Lval[0],Lval[1],Lval[2
        st='{},{},{:3d},{:5.1f},{:5.1f},{:5.1f},{:5.1f}\n'.format(Lval[0],Lval[1],Lval[2],Lval[3
        return st
    else:
        Lesp=['type','mmsi', 'shiptype','mothership_mmsi']
        Lval=[]
        for esp in Lesp:
           # Lval.append(data[esp])
             if esp in data: Lval.append(data[esp])
             else: Lval.append(0)
#        st='{},{},{:3d},{:6s},{:6s},{:6s},{:6s},{:8d}\n'.format(Lval[0],Lval[1],Lval[2],'*','*
        st='{},{},{:3d},{:6s},{:6s},{:6s},{:6s}\n'.format(Lval[0],Lval[1],Lval[2],'*','*','*','
        return st

def trenca_tira(tira):
    L=[]
    n=tira.find('*')
    L.append(tira[0:n+3])
    L.append(tira[n+3:])
    return L


def msgIsGood(mes):
    """ Els missatges tipus 5 poden tenir una longitut incorrecte.
    aquesta funcio si es true pot passar el missatge"""
    message_type = pyAISm.get_msg_type(mes)  # evitem els GPS i semblants
    if not(message_type =='!AIVDM' or message_type == '!AIVDO'):
           return False
    if (pyAISm.compute_checksum(mes).strip() != pyAISm.get_checksum(mes).strip()):
       return False

#    if (mes[14]==','): #elimina missatges estil !AIVDM,1,1,,A,,0*26 pq són erronis
#        return False

    ddd=get_data(mes)
    if ddd=='':
        return False
    else:

        ttt=get_tipus(mes)

    if (ttt==5 and len(ddd)==168):
       return False
    return True


def get_tipus(msg):
    aaa=pyAISm.get_payload(msg)
    bbb=pyAISm.decod_payload(aaa)

    return int(bbb[0:6],2)
def get_part24(msg):
     """ per el typus 24 dona Ao B segons"""
     bbb=get_data(msg)
```

```
    a=bbb[38:40]
    if a=='01':
        return 'B'
    else:
        return 'A'

def get_data(msg):
    aaa=pyAISm.get_payload(msg)
    bbb=pyAISm.decod_payload(aaa)

    return bbb

#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
def rev_estimate(lati,loni, latf, lonf):
    DL = (lonf - loni)*60   #minutes
    Dl = (latf - lati)*60   #minutes
    lm = (lati+latf)/2  #degrees
    A = DL*cos(radians(lm))
    dist = pow((pow(A,2)+pow(Dl,2)),0.5)
    dist = dist*1.852     #km

    return dist
#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
def split_date(date):
    dat = pd.DataFrame(index = range(date.shape[0]), columns = ['year','month','day','hour','m
    dat= dat.fillna(0)
    #date = np.zeros((idata, 6))
    aux = date/1e10
    dat.year = aux.round(0)

    aux = date/1e8 - dat.year*1e2
    dat.month = aux.apply(np.floor)

    aux = date/1e6 - dat.year*1e4 - dat.month*1e2
    dat.day = aux.apply(np.floor)

    aux = date/1e4 - dat.year*1e6 - dat.month*1e4 - dat.day*1e2
    dat.hour =  aux.apply(np.floor)

    aux = date/1e2 - dat.year*1e8 - dat.month*1e6 - dat.day*1e4 - dat.hour*1e2
    dat.minute = aux.apply(np.floor)

    dat.second_received = date - dat.year*1e10 - dat.month*1e8 - dat.day*1e6 - dat.hour*1e4- da

    return dat

#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
def filt_tstamp(data):
    ind = np.where(data >= 60)        #bad tstamp

    return ind
```

8.1.2. Traffic analysis functions

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import pandas as pd
import numpy as np
"""
@author: Anna Mujal-Colilles
"""

#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
def apb_lim(data,r):      #filters lat-lon boundaries of the Port of Barcelona based on circle
    R = 3440.06             # radious of the earth in nautical miles
    fnaut = np.radians([2.184643,41.38247])  #longitude, latitude facultat de nàutica in radiar
    data = data.loc[((R**2*(((np.radians(data.lon)-fnaut[0])*np.cos((np.radians(data.lat)+fnaut
    return data

#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
def inport(lat,lon):
    R = 3440.06
    r = 4.4
    #b = np.radians([2.092140,41.353021])
    b = np.radians([2.091666667,41.35333333])
    value = ((R**2*(((np.radians(lon)-b[0])*np.cos((np.radians(lat)+b[1])/2))**2+(np.radians(la
    value = value <= r**2
    return value
```

8.2. Decoding script

```python
# Project directory
ProjPath = '/home/anna/Documents/AIS/project/'

import sys
import pyAISm
from pathlib import Path
sys.path.append(ProjPath)

#Import local functions script

from src.functions.AISfunctions import *

# Previous step:
    #Open a terminal window within the folder containing the txt hourly files of messages you w
    #type ls *.txt>files_list.txt

# Initial variables
folder = 'test'
llist_arxius='files_list.txt'   #every line is one hour
arrel_out='decoded'   # don't finish the line with \n


# Create a the output folder
directory = Path(ProjPath + 'data/interim/'+folder+'/')
directory.mkdir(exist_ok=True)

# Output files
f123=open(ProjPath + 'data/interim/'+folder+'/'+arrel_out+'_123.txt','w')
f5=open(ProjPath + 'data/interim/'+folder+'/'+arrel_out+'_5.txt','w')
f18=open(ProjPath + 'data/interim/'+folder+'/'+arrel_out+'_18.txt','w')
f24A=open(ProjPath + 'data/interim/'+folder+'/'+arrel_out+'_24A.txt','w')
f24B=open(ProjPath + 'data/interim/'+folder+'/'+arrel_out+'_24B.txt','w')

# Counters
n=0
m=0 #error counter
i=0  # number of AIS messages with succesfull checknum
agrup=[1,2,3]

# Decoding process

with open(ProjPath + 'data/raw/'+folder+'/'+llist_arxius,'r') as f_arx:

   for arx in f_arx:
      n=n+1
      print(arx)
      tim_fix=arx[0:10]

      with open(ProjPath + 'data/raw/'+folder+'/'+ arx.rstrip('\n')) as f:
          msg='»'
          while msg:
             i=i+1
    #          print(i)
             msg=f.readline() # sinocomenca amb !AIVDM o !AIVDO passi a lalinia seguent
   #          print(msg)
             L=trenca_tira(msg)
             msg=L[0]    #tira sense cua efegida
             hm=L[1]     # hora i minut afegit
             tim_tira=tim_fix+hm
             tim_tira=tim_tira.rstrip('\n')
             if (msgIsGood(msg)== False):
                continue
             try:
                ais_data = pyAISm.decod_ais(msg.strip())
```

```
            except:
               m=+1  #detectat un error
               continue
            if (len(ais_data) ==1 ) :
               #print("trobat una part")  #no s'escriu i es deixa per si completa l'altra part
               continue
            categoria=ais_data['type']
            if categoria in typeAna:
#               print(categoria)
               if (categoria ==1 or categoria ==2 or categoria ==3 ):
                #  print(ais_data)
                  tira=fer_tira_123(ais_data)
                  tira=tim_tira+tira
                  f123.write(tira)
                  i123=i123+1
               if (categoria==5):
                     tira=fer_tira_5(ais_data)
                     tira = tim_tira+','+tira  #afegeix la data d'adquisicio del missatges al
                     f5.write(tira)
                     i5 += 1
               if (categoria==18):
                     tira=fer_tira_18(ais_data)
                     tira=tim_tira+tira
                     f18.write(tira)
                     i18 += 1
               if (categoria==24):
                  if get_part24(msg)=='A':
                     tira=fer_tira_24A(ais_data)
                     f24A.write(tira)
                     i24A += 1
                  else:
                     tira=fer_tira_24B(ais_data)
                     tira = tim_tira+','+tira  #afegeix la data d'adquisicio del missatges al
                     f24B.write(tira)
                     i24B += 1
            else:
               continue
    print(' Arxius entrats   {:8d} de hora'.format(n))
    print(' Reportats        {:8d} typus 123'.format(i123))
    print(' Reportats        {:8d} typus 5'.format(i5))
    print(' Reportats        {:8d} typus 18'.format(i18))
    print(' Reportats        {:8d} typus 24A'.format(i24A))
    print(' Reportats        {:8d} typus 24B'.format(i24B))
    print(' Reportats totals {:8d} '.format(i))
    print(' Errors divbersos {:8d}'.format(m))


    # Close files with decoded messages

    f_arx.close()
    f123.close()
    f5.close()
    f18.close()
    f24A.close()
    f24B.close()
```

8.3. Pre-Process

1_PreProcess

May 13, 2022

This script can only be run after decoding the messages with AIS_decoder.py

1 Import libraries

[1]:
```
import numpy as np
import pandas as pd
from pathlib import Path
```

2 Globals

[2]:
```
ProjPath = '/home/anna/Documents/AIS/project/'
```

[3]:
```
folder = 'test'
```

Type of messages to be decoded: ClassA or ClassB

[4]:
```
mtype = 'ClassA'     #ClassA or ClassB
```

2.1 Latitude and Longitude boundaries

[5]:
```
lat_bounds = [39.3619611, 43.2312167]
lon_bounds = [0.152775, 4.8690028]
```

2.2 Non Available Data

Check ITU-R Recommendations

[6]:
```
nad_lat = 91
nad_lon = 181
nad_sog = 1023
nad_rot = -128
nad_cog = 360
nad_hdg = 511
```

3 Load files

[7]:
```
if (mtype=='ClassA'):
    mes = pd.read_csv(ProjPath+'data/interim/'+folder+'/decoded_123'+'.
↪txt',sep=",",header=None)
    mes.columns =['date','second_sent','type','mmsi','status','turn',␣
↪'speed','lon','lat','course','heading']
    mes = mes.drop(columns = ['type'])
else:
    mes = pd.read_csv(ProjPath+'data/interim/'+folder+'/decoded_18'+'.
↪txt',sep=",",header=None)
    mes.columns =['date','second_sent','type','mmsi',␣
↪'speed','lon','lat','course','heading']
    mes = mes.drop(columns = ['type'])
sarx = mtype
```

3.1 initial amount of data

[8]:
```
i_data = len(mes.index)
i_data
```

[8]: 133955

4 Filter dynamic messages

4.1 LatLon boundaries

[9]:
```
ind = np.where((mes.lat < lat_bounds[0]) | (mes.lat > lat_bounds[1]))
mes = mes.drop(mes.index[ind])
len(mes)
```

[9]: 133776

[10]:
```
ind = np.where((mes.lon < lon_bounds[0]) | (mes.lon > lon_bounds[1]))
mes = mes.drop(mes.index[ind])
len(mes)
```

[10]: 133776

4.2 Non Available Data

4.2.1 latitude

[11]:
```
ind = np.where(mes.lat == nad_lat)
err = np.shape(ind)[1]/i_data*100
err
```

[11]: 0.0

[12]:
```
mes.lat = mes.lat.replace(nad_lat,np.nan)
```

4.2.2 longitude

[13]:
```
ind = np.where(mes.lon == nad_lon)
err = np.shape(ind)[1]/i_data*100
err
```

[13]: 0.0

[14]:
```
mes.lon = mes.lon.replace(nad_lon, np.nan)
```

4.2.3 speed

[15]:
```
ind = np.where(mes.speed == nad_sog)
err = np.shape(ind)[1]/i_data*100
err
```

[15]: 0.23515359635698554

[16]:
```
mes.speed = mes.speed.replace(nad_sog,np.nan)
```

4.2.4 ROT

[17]:
```
if (mtype=='ClassA'):
    ind = np.where(mes.turn == nad_rot)
    err = np.shape(ind)[1]/i_data*100
    print(err)
    mes.turn = mes.turn.replace(nad_rot,np.nan)
```

32.635586577582025

4.2.5 COG

[18]:
```
ind = np.where(mes.course == nad_cog)
err = np.shape(ind)[1]/i_data*100
err
```

[18]: 5.527229293419432

[19]:
```
mes.course = mes.course.replace(nad_cog,np.nan)
```

3 Data loading and filtering

3.1 Data loading

[6]:
```python
# Data loading
## This section reads all available static and dynamic AIS data and converts␣
↪them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(projPath+'/data/interim/'+fold)

    # Dynamic data range filter
    t = pd.read_csv('filtered_ClassA.csv', sep=",", usecols =␣
↪['date','mmsi','lat','lon','status','speed'])
    t.speed = t.speed/10
    t = apb_lim(t, r)
    m = np.unique(t.mmsi)

    # Static data loading
    aux = pd.read_csv('decoded_5.txt', sep = ",")
    aux.columns = ['date','type','mmsi','IMO','shipname',␣
↪'shiptype','to_bow','to_stern', 'to_port','to_starboard','draught',␣
↪'destination']
    aux=aux.drop(columns = ['date','type','IMO','shipname','to_bow','to_stern',␣
↪'to_port','to_starboard','draught','destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset = 'mmsi', keep = "first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m,t,aux)

print("data length:", len(s))
s.head()
```

```
data length: 11932625
```

8.4. Traffic analysis

8.4.1. Port calls

In [ ]:
```
projPath = '/home/anna/Documents/AIS/project/'
```

Developed by: Javier Nieto-Guarasa

Supervised by: Anna Mujal-Colilles, PhD

Polytechnic University of Catalonia

Import libraries

In [ ]:
```
import pandas as pd
import numpy as np
import math
import os

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, AutoMinor
from matplotlib import ticker, cm
import sys
```

In [ ]:
```
sys.path.append(projPath)
```

In [ ]:
```
from src.functions.AIS_spatialFunctions import apb_lim, inport
```

Global variables

In [ ]:
```
folder = ['202003','202004','202005','202006','202007']
labelship = ['Passenger','Cargo','Tankers']
months = ['March', 'April', 'May', 'June', 'July']
```

Load data

In [ ]:
```
# Data loading
## This section reads all available static and dynamic AIS data and converts

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(projPath+'/data/interim/'+fold)

    # Dynamic data range filter + Inport function
    t = pd.read_csv('filtered_ClassA.csv', sep=",", usecols = ['date','mmsi',
    t = apb_lim(t,r)
    t['inport'] = inport(t.lat, t.lon) # This function returns a bool (T/F) v
    t = t.drop(columns = ['lat','lon'])
    m = np.unique(t.mmsi)

    Static data loading
```

```
    aux = pd.read_csv('decoded_5.txt', sep = ",")
    aux.columns =['date','type','mmsi','IMO','shipname', 'shiptype','to_bow',
    aux=aux.drop(columns = ['date','type','IMO','shipname','to_bow','to_stern

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset='mmsi',keep="first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m,t,aux)

print("data length:", len(s))
s.head()
```

Time filtering

In [ ]:

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops dupli

s.date = pd.to_datetime(s['date'], format = '%Y%m%d%H%M%S')
s['date'] = s.date.dt.round('1h')
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')

print("data length:", len(s))
s.head()
```

Vessel calls

Call filter

In [ ]:

```
# Call filter
## This section groups data into bunches of in-port and out-port datasets pe
## A vessel call is considered as a bool value change from in-port = False to

df = s
df['mmsi2'] = df.mmsi
df = df.groupby(['mmsi2'])
m = np.unique(s.mmsi)
calls = pd.DataFrame()
for i in range(0,len(df)):
    k = df.get_group(m[i])
    k['groupno'] = k.inport.diff().cumsum().fillna(0)
    result = k.groupby(['groupno']).agg(['first'])
    result.inport = result.inport.astype(int)
    result = result.loc[(result.inport['first'] == 1)]
    calls = calls.append(result, ignore_index = True)
    del(k, result)
del(df, m, i)
```

```
calls = calls.stack().reset_index().drop(columns = ['level_0', 'level_1', 'ir

print("data length:", len(calls))
calls.head()
```

Call refining

In [ ]:

```
# Call refining
## This section drops all calls dated 2020-03-01 00:00:00, so as to prevent p

calls = calls.loc[calls.date != '2020-03-01 00:00:00']
calls = calls.loc[calls.mmsi != 224022660]
calls = calls.loc[calls.mmsi != 224022650]
calls = calls.loc[calls.mmsi != 224334000]
calls['date'] = calls.date.dt.round('1d')
calls = calls.loc[calls.status != 1]

print("data length:", len(calls))
calls.head()
```

Daily calls

Figures - total

In [ ]:

```
# Total figures by day
## This section groups and counts the number of calls per day along a 5-montl

byday = calls.groupby(['date']).size().reset_index(name = 'number_of_calls')
print("data length:", len(byday))
byday.head()
```

Plot - total

In [ ]:

```
# Total figures by day (plot)
## This section shows the evolution in the number of calls per day along a 5-

fig, ax = plt.subplots(figsize=(15,7))
calls.groupby(['date']).count()['shiptype'].plot(ax=ax, color = 'k').legend([
plt.ylabel('daily calls')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-06-30'))
ax.set_ylim([5,35])
plt.title('Daily number of calls')
plt.grid()
plt.show()
```

Figures - per shiptype

In [ ]:

```
# Total figures by day and shiptype
## This section groups and counts the number of calls per day and shiptype al

calls['group'] = pd.cut(calls.shiptype, 3, right=False, labels=labelship)
bydayngroup = calls.groupby(['date','group']).size().to_frame('number_of_call
bydayngroup.head()
```

Plot - per shiptype

In [ ]:

```
# Total figures by day and shiptype (plot)
## This section shows the evolution in the number of calls per day and shipty

calls['group'] = pd.cut(calls.shiptype, 3, right=False, labels=labelship)
fig, ax = plt.subplots(figsize=(15,7))
calls.groupby(['date','group']).count().fillna(0)['shiptype'].unstack().plot(
calls.groupby(['date']).count()['shiptype'].plot(ax=ax, color= 'k').legend(la
plt.ylabel('daily calls')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,35])
plt.title('Daily number of calls')
plt.grid()
```

Calls per ship

In [ ]:

```
# Total calls by ship
## This section groups and counts the number of calls per ship along a 5-mont

byship = calls.groupby(['mmsi']).size().reset_index(name = 'number_of_calls')
byship = byship.sort_values(by = ['number_of_calls'], ascending = False )

print("total number of ships:", len(byship))
byship.head()
```

Calls per shiptype

Figures - total

In [ ]:

```
# Total calls by shiptype
## This section groups and counts the number of calls per shiptype along a 5-

bygroup = calls.groupby(['group']).size().reset_index(name = 'number_of_calls
bygroup
```

Pie chart - total

In [ ]:

```
# Total calls by shiptype (plot)
## This section shows in a pie chart the number of calls per shiptype along a

fig, ax = plt.subplots()
sizes = [bygroup.iloc[0]['number_of_calls'], bygroup.iloc[1]['number_of_calls
ax.pie(sizes, labels=labelship, autopct='%1.1f%%', shadow=False, startangle=9
ax.axis('equal')
plt.title('Calls by shiptype')
plt.show()
```

Figures - per month

In [ ]:

```
# Total calls by month and shiptype
## This section groups and counts the number of calls per shiptype and month

calls['month'] = calls.date.apply(lambda x: x.month)
calls.month = pd.cut(calls.month, 5, right=False, labels=months)
calls = calls.drop(columns = ['date'])
bymonth = calls.groupby(['month','group']).size().reset_index(name = 'number_
bymonth.head()
```

Pie chart - per month

In [ ]:

```
# Total calls by month and shiptype (plot)
## This section shows in a pie chart the number of calls per shiptype and mo

calls.groupby(['month','group']).size().unstack(level = 0).plot.pie(subplots
figsize = (25,20), autopct='%1.1f%%', colors = ['#0000FF','#32CD32','#FF0000'
plt.legend(loc = 'best')
plt.title('Calls by shiptype')
plt.show()
```

In [ ]:

8.4.2. Maritime traffic

AISdata_status

May 13, 2022

[1]:
```
projPath = '/home/anna/Documents/AIS/project/'
```

Developed by: Javier Nieto-Guarasa Supervised by: Anna Mujal-Colilles, PhD Polytechnic University of Catalonia

1 Import libraries

[2]:
```
import pandas as pd
import numpy as np
import os
import sys

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter,␣
 ↪AutoMinorLocator)
from matplotlib import ticker, cm
```

[3]:
```
sys.path.append(projPath)
```

[4]:
```
from src.functions.AIS_spatialFunctions import apb_lim, inport
```

2 Global variables

[5]:
```
folder = ['202003','202004','202005','202006','202007']
labelship = ['Passenger','Cargo','Tankers']
labelstatus = ['Underway', 'At Anchor', 'NUC', 'Moored']
```

3 Data loading and filtering

3.1 Data loading

[6]:
```
# Data loading
## This section reads all available static and dynamic AIS data and converts␣
↪them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(projPath+'/data/interim/'+fold)

    # Dynamic data range filter
    t = pd.read_csv('filtered_ClassA.csv', sep=",", usecols =␣
↪['date','mmsi','lat','lon','status','speed'])
    t.speed = t.speed/10
    t = apb_lim(t, r)
    m = np.unique(t.mmsi)

    # Static data loading
    aux = pd.read_csv('decoded_5.txt', sep = ",")
    aux.columns = ['date','type','mmsi','IMO','shipname',␣
↪'shiptype','to_bow','to_stern', 'to_port','to_starboard','draught',␣
↪'destination']
    aux=aux.drop(columns = ['date','type','IMO','shipname','to_bow','to_stern',␣
↪'to_port','to_starboard','draught','destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset = 'mmsi', keep = "first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m,t,aux)

print("data length:", len(s))
s.head()
```

```
data length: 11932625
```

[6]:

	date	mmsi	status	speed	lon	lat	shiptype
0	20200301000000	247243600	0	0.0	2.17512	41.36382	60
1	20200301000001	209293000	5	0.0	2.17313	41.35071	60
2	20200301000002	224878000	5	3.6	2.17696	41.36235	89
3	20200301000002	229866000	0	7.8	2.17758	41.30095	80
4	20200301000003	225423000	0	18.0	2.25558	41.21787	65

4 Time filtering

[7]:

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops␣
↪duplicated values based on dt = 1h basis

s.date = pd.to_datetime(s['date'], format = '%Y%m%d%H%M%S')
s['date'] = s.date.dt.round('1d')
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')

print("data length:", len(s))
s.head()
```

data length: 8182

[7]:

	date	mmsi	status	speed	lon	lat	shiptype
0	2020-03-01	247243600	0	0.0	2.17512	41.36382	60
1	2020-03-01	209293000	5	0.0	2.17313	41.35071	60
2	2020-03-01	224878000	5	3.6	2.17696	41.36235	89
3	2020-03-01	229866000	0	7.8	2.17758	41.30095	80
4	2020-03-01	225423000	0	18.0	2.25558	41.21787	65

5 Status filtering

[8]:

```
# Status filtering
## This section assigns group and status to each dataset and drops status other␣
↪than "Underway", "At Anchor", "NUC" or "Moored"

s['group'] = pd.cut(s.shiptype, 3, right=False, labels=labelship)
s = s.loc[(s.status != 3) & (s.status != 4) & (s.status != 6) & (s.status != 7)␣
↪& (s.status != 8) & (s.status != 15)]
s.status.loc[s.status == 5] = 3
s['vsl_status'] = pd.cut(s.status, 4, right=False, labels=labelstatus)

print("data length:", len(s))
s.head()
```

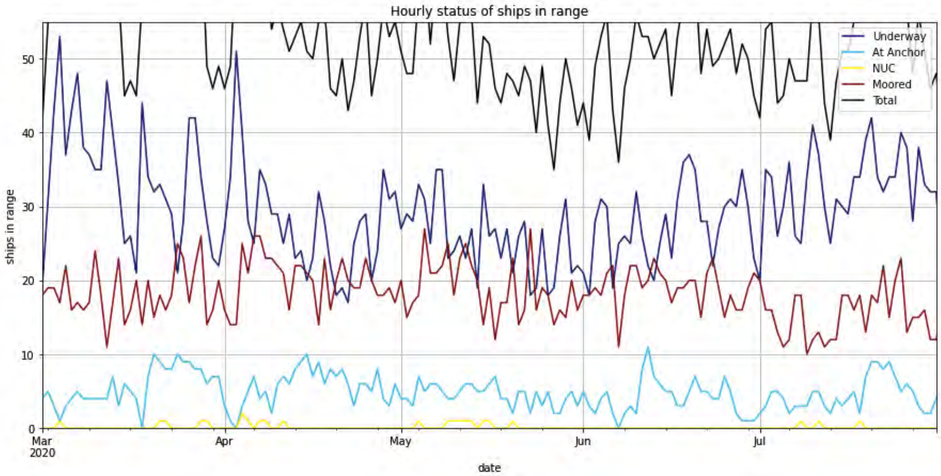
data length: 8130

[8]:
```
         date       mmsi  status  speed      lon       lat  shiptype  \
0  2020-03-01  247243600       0    0.0  2.17512  41.36382        60
1  2020-03-01  209293000       3    0.0  2.17313  41.35071        60
2  2020-03-01  224878000       3    3.6  2.17696  41.36235        89
3  2020-03-01  229866000       0    7.8  2.17758  41.30095        80
4  2020-03-01  225423000       0   18.0  2.25558  41.21787        65

       group vsl_status
0  Passenger   Underway
1  Passenger     Moored
2    Tankers     Moored
3    Tankers   Underway
4  Passenger   Underway
```

6 Hourly status of ships in range

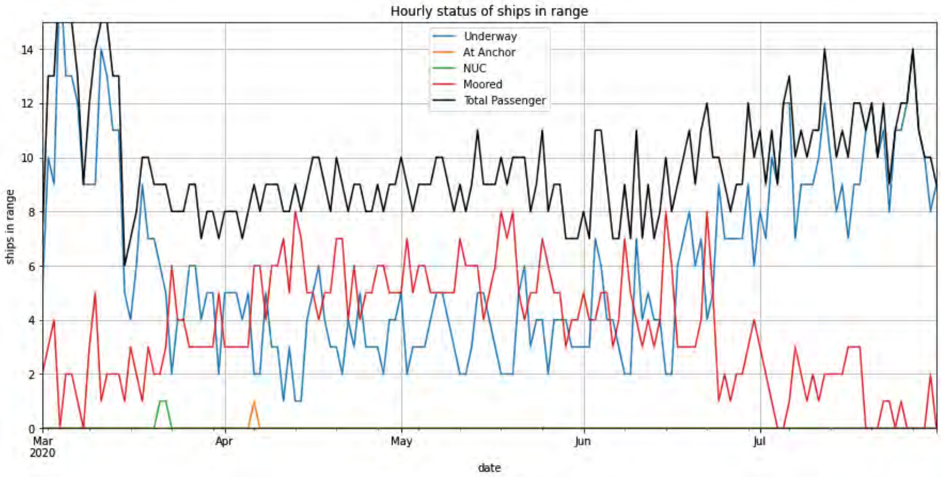
6.1 Total

[9]:
```
# Hourly status of ships in range
## This section shows the hourly evolution of status of vessels in range along␣
↪a 5-month time period

n_clusters = 10
cmap = plt.get_cmap('jet', n_clusters)
fig, ax = plt.subplots(figsize=(15,7))
s.groupby(['date','vsl_status']).count()['status'].unstack().fillna(0).
↪plot(ax=ax, colormap = cmap)
s.groupby(['date']).size().plot(ax=ax, color= 'k').legend(labelstatus+['Total'])
plt.ylabel('ships in range')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,55])
plt.title('Hourly status of ships in range')
plt.grid()
plt.show()
```



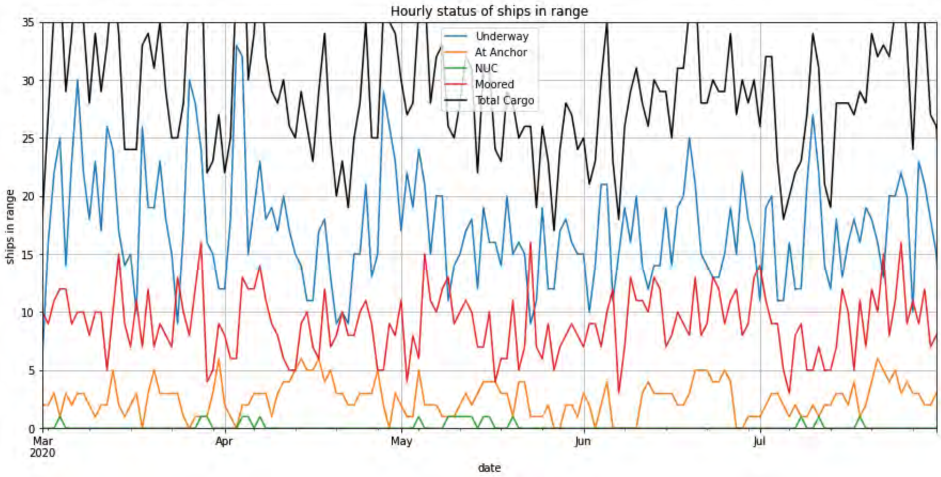
6.2 Passenger

[10]:
```
# Hourly status of passenger ships in range
## This section shows the hourly evolution of status of passenger vessels in␣
↪range along a 5-month time period

sf = s.loc[s.group == 'Passenger']
fig, ax = plt.subplots(figsize=(15,7))
sf.groupby(['date','vsl_status'])['status'].count().unstack().fillna(0).
↪plot(ax=ax)
sf.groupby(['date']).size().plot(ax=ax, color= 'k').legend(labelstatus+['Total␣
↪Passenger'])
plt.ylabel('ships in range')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,15])
plt.title('Hourly status of ships in range')
plt.grid()
plt.show()
```



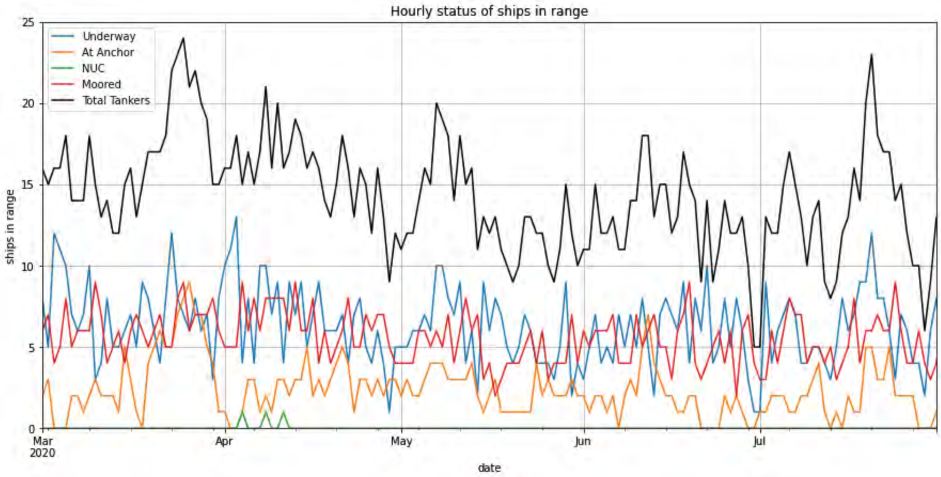
6.3 Cargo

[11]:
```
# Hourly status of cargo ships in range
## This section shows the hourly evolution of status of cargo vessels in range␣
↪along a 5-month time period

sf = s.loc[s.group == 'Cargo']
fig, ax = plt.subplots(figsize=(15,7))
sf.groupby(['date','vsl_status']).count()['status'].unstack().fillna(0).
↪plot(ax=ax)
sf.groupby(['date']).size().plot(ax=ax, color= 'k').legend(labelstatus+['Total␣
↪Cargo'])
plt.ylabel('ships in range')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,35])
plt.title('Hourly status of ships in range')
plt.grid()
plt.show()
```



6.4 Tankers

[12]:
```
# Hourly status of tanker ships in range
## This section shows the hourly evolution of status of tanker vessels in range␣
↪along a 5-month time period

sf = s.loc[s.group == 'Tankers']
fig, ax = plt.subplots(figsize=(15,7))
sf.groupby(['date','vsl_status']).count()['status'].unstack().fillna(0).
↪plot(ax=ax)
sf.groupby(['date']).size().plot(ax=ax, color= 'k').legend(labelstatus+['Total␣
↪Tankers'])
plt.ylabel('ships in range')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,25])
plt.title('Hourly status of ships in range')
plt.grid()
plt.show()
```


Hourly status of ships in range
Underway
At Anchor
NUC
Moored
Total Tankers
ships in range
date
Mar
2020
Apr
May
Jun
Jul
0
5
10
15
20
25

8.5. Emissions

AISemissions_db

May 13, 2022

1 AIS emissions - method 1

[ ]:
```
projPath = '/home/anna/Documents/AIS/project/'
```

Developed by: Javier Nieto-Guarasa Supervised by: Anna Mujal-Colilles, PhD Polytechnic University of Catalonia

2 Import libraries

[ ]:
```
import pandas as pd
import numpy as np
import math
import os
import sys

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

import seaborn as sns

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter,␣
 ↪AutoMinorLocator)
from matplotlib import ticker, cm
```

[ ]:
```
sys.path.append(projPath)
```

[ ]:
```
from src.functions.AIS_spatialFunctions import apb_lim, inport
```

3 Global variables

[ ]:
```
folder = ['202003','202004','202005','202006','202007']
labelship = ['Passenger','Cargo','Tankers']
labelstatus = ['Underway', 'At Anchor', 'NUC', 'Restricted Maneuverability',␣
 ↪'Constrained by her draught', 'Moored',
               'Aground', 'Sailing', 'Error', 'Towing', 'Undefined']
```

3.1 Engine data

```
[ ]: EL = 0.80             # Average design engine load on merchant ships␣
      ↪(Jalkanen et al, 2012)
     SFOC = 200            # Average specific fuel oil consumption (g/kWh)␣
      ↪(Jalkanen et al, 2009)
     SFOC_AE = 220         # Average specific fuel oil consumption of aux. engines␣
      ↪(g/kWh) (Jalkanen et al, 2012)
     rpm = 500             # Average working revolutions on medium speed engines␣
      ↪(Jalkanen et al, 2009)
```

3.2 Fuel qualities

```
[ ]: SC_fuel = 0.5         # Sulfur content of Light Fuel Oil (%)
     CC_fuel = 86          # Carbon content of Light Fuel Oil as per ISO 8217 (%)
     SC_diesel = 0.5       # Sulfur content of Marine Gasoil (%)
     CC_diesel = 87.5      # Carbon content of Marine Gasoil as per ISO 8217 (%)
     SC_lng = 4e-3         # Sulfur content of LNG (%)
     CC_lng = 75           # Carbon content of LNG (%)
```

3.3 Element properties

```
[ ]: m_S = 32.0655         # Molar mass of sulfur (g/mol)
     m_SO2 = 64.06436      # Molar mass of sulfur dioxide (g/mol)
     m_C = 12.01           # Molar mass of carbon (g/mol)
     m_CO2 = 44.0886       # Molar mass of carbon dioxide (g/mol)
```

3.4 PM calculation

```
[ ]: ef_ec = 0.08          # Emission factor for elementary carbon (g/kWh)
     ef_oc = 0.2           # Emission factor for organic carbon (g/kWh)
     ef_ash = 0.06         # Emission factor for ashes (g/kWh)
     oc_el = 1.025         # Organic carbon related to engine load (dimensionless)
```

4 Data loading and filtering

```
[ ]: # Data loading
     ## This section reads all available static and dynamic AIS data and converts␣
      ↪them into a workable pandas DataFrame

     r = 30 # Enter range radius in nautical miles (1nm = 1852m)

     s = pd.DataFrame()
     for fold in folder:
         os.chdir(projPath+'/data/interim/'+fold)
```

```
    # Dynamic data range filter + Inport function
    t = pd.read_csv('filtered_ClassA.csv', sep=",", usecols =␣
↪['date','mmsi','lat','lon','status', 'speed'])
    t.speed = t.speed/10
    t = apb_lim(t, r)
    t['inport'] = inport(t.lat, t.lon) # This function returns a bool (T/F)␣
↪value to the question "Is the vessel in po
    m = np.unique(t.mmsi)

    # Static data loading
    aux = pd.read_csv('decoded_5.txt', sep = ",")
    aux.columns =['date','type','mmsi','IMO','shipname',␣
↪'shiptype','to_bow','to_stern', 'to_port','to_starboard','draught',␣
↪'destination']
    aux=aux.drop(columns = ['date','type','shipname','to_bow','to_stern',␣
↪'to_port','to_starboard','draught','destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset='mmsi',keep="first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m,t,aux)

s.loc[s.speed > 40,'speed'] = 0
s = s.loc[s.IMO != 0]
print("data length:", len(s))
s.head()
```

[ ]:
```
p = pd.read_excel(projPath +'data/raw/emissions/' +'Particulars.xlsx')
p.loc[p.SFC == 0,'SFC'] = SFOC
p['ef_NOx'] = (44*rpm**-0.23)
p.loc[(p.Built < 2011) & (p.rpm < 130),'ef_NOx'] = 17
p.loc[(p.Built < 2011) & (p.rpm >= 130) & (p.rpm < 2000),'ef_NOx'] = (45*p.
↪rpm**-0.2)
p.loc[(p.Built < 2011) & (p.rpm >= 2000),'ef_NOx'] = 9.8
p.loc[(p.Built >= 2011) & (p.rpm < 130),'ef_NOx'] = 14
p.loc[(p.Built >= 2011) & (p.rpm >= 2000),'ef_NOx'] = 7.7
```

```
p = p.drop(columns = ['Built', 'rpm'])
```

5 Time filtering

[ ]:
```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops duplicated values based on dt = 1h basis

s.date = pd.to_datetime(s['date'], format = '%Y%m%d%H%M%S')
s['date'] = s.date.dt.round('1min')
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')

print("data length:", len(s))
s.head()
```

6 Status filtering

[ ]:
```
# Status filtering
## This section assigns group and status to each dataset based on the corresponding numerical value

s['group'] = pd.cut(s.shiptype, 3, right=False, labels = labelship)
s.loc[s.status == 0,'status'] = labelstatus[0]
s.loc[s.status == 1,'status'] = labelstatus[1]
s.loc[s.status == 2,'status'] = labelstatus[2]
s.loc[s.status == 3,'status'] = labelstatus[3]
s.loc[s.status == 4,'status'] = labelstatus[4]
s.loc[s.status == 5,'status'] = labelstatus[5]
s.loc[s.status == 6,'status'] = labelstatus[6]
s.loc[s.status == 8,'status'] = labelstatus[7]
s.loc[s.status == 10,'status'] = labelstatus[8]
s.loc[s.status == 11,'status'] = labelstatus[9]
s.loc[s.status == 15,'status'] = labelstatus[10]
s = s.drop(columns = ['shiptype'])

print("data length:", len(s))
s.head()
```

6.1 First dataset rearrangement

[ ]:
```
# Dataset rearrangement (1st)
## This section copies all datasets with status "Moored" and "At Anchor" with speeds < 3 knots and offsets the time by 1min
```

```
## This overcomes the fact that these vessels, due to their status, only␣
↪transmit dynamic data every 3 minutes

df = s.loc[(s.status == 'Moored') | (s.status== 'At Anchor')]
df = df.loc[df.speed < 3]
dupli_df = pd.concat([df]*3, ignore_index=True)
l_col_datetime = dupli_df.select_dtypes('datetime').columns
len_df = len(df)
dupli_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minutes=1)
dupli_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli_df))
dupli_df.head()
```

6.2 Second dataset rearrangement

[ ]:

```
# Dataset rearrangement (2nd)
## This section copies all datasets with status "Moored" and "At Anchor" with␣
↪speeds < 3 knots and offsets the time by 1min
## A 2nd rearrangement generates datasets that might not be available and␣
↪stabilizes the dynamic plot

dupli2_df = pd.concat([dupli_df]*3, ignore_index=True)
l_col_datetime = dupli2_df.select_dtypes('datetime').columns
len_df = len(dupli_df)
del(dupli_df)
dupli2_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minutes=1)
dupli2_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli2_df))
dupli2_df.head()
```

6.3 Third dataset rearrangement

[ ]:

```
# Dataset rearrangement (3rd)
## This section copies all datasets with status "Moored" and "At Anchor" with␣
↪speeds < 3 knots and offsets the time by 1min
## A 3rd rearrangement generates datasets that might not be available and␣
↪stabilizes the dynamic plot

dupli3_df = pd.concat([dupli2_df]*3, ignore_index=True)
l_col_datetime = dupli3_df.select_dtypes('datetime').columns
len_df = len(dupli2_df)
del(dupli2_df)
dupli3_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minutes=1)
dupli3_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)
```

```
print("data length:", len(dupli3_df))
dupli3_df.head()
```

6.4 Data appending

```
[ ]: # Final data appending
     ## This section appends the rearranged dynamic data for vessels with status␣
      ↪"Moored" and "At Anchor" with all other vessels
     ## The final data length is much larger than the initial one, as it guarantees␣
      ↪a dataset available per ship every 1min

     s = pd.concat([s,dupli3_df], ignore_index = True)
     s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')
     del(dupli3_df)

     print("data length:", len(s))
     s.head()
```

7 Emission inventory

7.1 Phase filtering

```
[ ]: # Phase filtering
     ## This section separates the database into the 4 stages: "Cruising", "At␣
      ↪Anchor", "Maneuvering" and "Hoteling"

     s['AE'] = 0.6
     s.inport = s.inport.astype(int)
     s['SC'] = SC_fuel/100
     s['CC'] = CC_fuel/100
     sf_in = s.loc[s.inport == 1]
     sf_hotelling = sf_in.loc[sf_in.speed <= 0.5]
     sf_maneuvering = sf_in.loc[sf_in.speed > 0.5]
     sf_out = s.loc[s.inport == 0]
     sf_anchor = sf_out.loc[sf_out.speed <= 1.5]
     sf_cruising = sf_out.loc[sf_out.speed > 1.5]
```

7.2 Crusing emissions

```
[ ]: # Cruising emissions
     ## This section computes the emissions of vessels in the "cruising" stage per␣
      ↪minute
     ## Main Engine loads are computed through the Propeller Law, whereas 60% is␣
      ↪assigned to auxiliary engines on cargo and tanker
```


(FNB-UPC)

```
## vessels, and 80% is assigned to those on passenger vessels
## Main engines are considered to burn their main fuel, whereas all auxiliary␣
↪engines burn MGO

sf_cruising = sf_cruising.merge(p, how = 'left', on = ['IMO','IMO'])
sf_cruising = sf_cruising.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp'])
sf_cruising.SC.loc[sf_cruising.Fuel == 'LNG'] = SC_lng/100
sf_cruising.CC.loc[sf_cruising.Fuel == 'LNG'] = CC_lng/100
sf_cruising.AE.loc[sf_cruising.group == 'Passenger'] = 0.8
```

[ ]:
```
sf_cruising['k'] = EL*sf_cruising.ENG_KW/((sf_cruising.service_speed*1852/
↪3600)**3)
sf_cruising['trans_KW'] = sf_cruising.k*(sf_cruising.speed*1852/3600)**3
sf_cruising['SFOC'] = sf_cruising.SFC*(0.455*(EL*(sf_cruising.speed/sf_cruising.
↪service_speed)**3)**2-0.17*(EL*(sf_cruising.speed/sf_cruising.
↪service_speed)**3)+1.28)
sf_cruising['SFOC_AE'] = SFOC_AE*(0.455*(sf_cruising.AE)**2-0.17*(sf_cruising.
↪AE)+1.28)
sf_cruising['FC'] = (sf_cruising.trans_KW*sf_cruising.SFOC*(1/60) + sf_cruising.
↪AE*sf_cruising.AUX_KW*sf_cruising.SFOC_AE*(1/60))*1e-6
sf_cruising['SO2'] = ((sf_cruising.SFOC*sf_cruising.SC/m_S)*m_SO2*sf_cruising.
↪trans_KW*(1/60) + sf_cruising.AE*(sf_cruising.SFOC_AE*SC_diesel/100/
↪m_S)*m_SO2*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['CO2'] = ((sf_cruising.SFOC*sf_cruising.CC/m_C)*m_CO2*sf_cruising.
↪trans_KW*(1/60) + sf_cruising.AE*(sf_cruising.SFOC_AE*CC_diesel/100/
↪m_C)*m_CO2*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['NOx'] = (sf_cruising.ef_NOx*sf_cruising.trans_KW*(1/60) +␣
↪sf_cruising.AE*(45*rpm**-0.2)*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['PM'] = (sf_cruising.trans_KW*(0.455*(EL*(sf_cruising.speed/
↪sf_cruising.service_speed)**3)**2-0.17*(EL*(sf_cruising.speed/sf_cruising.
↪service_speed)**3)+1.28)*((0.312*sf_cruising.SC)+(0.244*sf_cruising.
↪SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_cruising.AE*sf_cruising.AUX_KW*(0.
↪455*sf_cruising.AE**2-0.17*sf_cruising.AE+1.28)*((0.312*SC_diesel/100)+(0.
↪244*SC_diesel/100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_cruising = sf_cruising.drop(columns = ['inport', 'AE', 'Fuel', 'SC', 'CC',␣
↪'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', 'k',␣
↪'trans_KW', 'ef_NOx'])
```

7.3 Maneuvering emissions

[ ]:
```
# Maneuvering emissions
## This section computes the emissions of vessels in the "maneuvering" stage␣
↪per minute
## Main Engine loads are computed through the Propeller Law, whereas 70% is␣
↪assigned to auxiliary engines on cargo and tanker
## vessels, and 80% is assigned to those on passenger vessels
```

```
## All engines are considered to burn MGO

sf_maneuvering = sf_maneuvering.merge(p, how = 'left', on = ['IMO','IMO'])
sf_maneuvering = sf_maneuvering.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp'])
sf_maneuvering.SC = SC_diesel/100
sf_maneuvering.CC = CC_diesel/100
sf_maneuvering.AE.loc[sf_maneuvering.group == 'Passenger'] = 0.8
sf_maneuvering.AE.loc[sf_maneuvering.group != 'Passenger'] = 0.7
```

[ ]:

```
sf_maneuvering['k'] = EL*sf_maneuvering.ENG_KW/((sf_maneuvering.
    service_speed*1852/3600)**3)
sf_maneuvering['trans_KW'] = sf_maneuvering.k*(sf_maneuvering.speed*1852/
    3600)**3
sf_maneuvering['SFOC'] = sf_maneuvering.SFC*(0.455*(EL*(sf_maneuvering.speed/
    sf_maneuvering.service_speed)**3)**2-0.17*(EL*(sf_maneuvering.speed/
    sf_maneuvering.service_speed)**3)+1.28)
sf_maneuvering['SFOC_AE'] = SFOC_AE*(0.455*(sf_maneuvering.AE)**2-0.
    17*(sf_maneuvering.AE)+1.28)
sf_maneuvering['FC'] = (sf_maneuvering.trans_KW*sf_maneuvering.SFOC*(1/60) +
    sf_maneuvering.SFOC_AE*sf_maneuvering.AE*sf_maneuvering.AUX_KW*(1/60))*1e-6
sf_maneuvering['SO2'] = ((sf_maneuvering.SFOC*sf_maneuvering.SC/
    m_S)*m_SO2*sf_maneuvering.trans_KW*(1/60) + sf_maneuvering.
    AE*(sf_maneuvering.SFOC_AE*SC_diesel/100/m_S)*m_SO2*sf_maneuvering.AUX_KW*(1/
    60))*1e-6
sf_maneuvering['CO2'] = ((sf_maneuvering.SFOC*sf_maneuvering.CC/
    m_C)*m_CO2*sf_maneuvering.trans_KW*(1/60) + sf_maneuvering.
    AE*(sf_maneuvering.SFOC_AE*CC_diesel/100/m_C)*m_CO2*sf_maneuvering.AUX_KW*(1/
    60))*1e-6
sf_maneuvering['NOx'] = (sf_maneuvering.ef_NOx*sf_maneuvering.trans_KW*(1/60) +
    sf_maneuvering.AE*(45*rpm**-0.2)*sf_maneuvering.AUX_KW*(1/60))*1e-6
sf_maneuvering['PM'] = (sf_maneuvering.trans_KW*(0.455*(EL*(sf_maneuvering.
    speed/sf_maneuvering.service_speed)**3)**2-0.17*(EL*(sf_maneuvering.speed/
    sf_maneuvering.service_speed)**3)+1.28)*((0.312*sf_maneuvering.SC)+(0.
    244*sf_maneuvering.SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_maneuvering.
    AE*sf_maneuvering.AUX_KW*(0.455*sf_maneuvering.AE**2-0.17*sf_maneuvering.
    AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_diesel/
    100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_maneuvering = sf_maneuvering.drop(columns = ['inport', 'AE', 'Fuel', 'SC',
    'CC', 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', 'k',
    'trans_KW', 'ef_NOx'])
```

7.4 At Anchor emissions

[ ]:
```
# At Anchor emissions
## This section computes the emissions of vessels in the "at anchor" stage per␣
↪minute
## Main Engine loads are estimated at 10% for all vessels, whereas 70% is␣
↪assigned to auxiliary engines on passenger and
## tanker vessels, and 40% is assigned to those on cargo vessels
## Main engines are considered to burn their main fuel, whereas all auxiliary␣
↪engines burn MGO

sf_anchor = sf_anchor.merge(p, how = 'left', on = ['IMO','IMO'])
sf_anchor = sf_anchor.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp'])
sf_anchor.SC.loc[sf_anchor.Fuel == 'LNG'] = SC_lng/100
sf_anchor.CC.loc[sf_anchor.Fuel == 'LNG'] = CC_lng/100
sf_anchor.AE.loc[sf_anchor.group == 'Passenger'] = 0.7
sf_anchor.AE.loc[sf_anchor.group == 'Tankers'] = 0.7
sf_anchor.AE.loc[sf_anchor.group == 'Cargo'] = 0.4
sf_anchor.ENG_KW.loc[sf_anchor.AUX_KW != 0 ] = 0
```

[ ]:
```
sf_anchor['SFOC'] = sf_anchor.SFC*(0.455*(EL*0.1)**2-0.17*(EL*0.1)+1.28)
sf_anchor['SFOC_AE'] = SFOC_AE*(0.455*(sf_anchor.AE)**2-0.17*(sf_anchor.AE)+1.
↪28)
sf_anchor['FC'] = (0.1*sf_anchor.ENG_KW*sf_anchor.SFOC*(1/60) + sf_anchor.
↪AE*sf_anchor.AUX_KW*sf_anchor.SFOC_AE*(1/60))*1e-6
sf_anchor['SO2'] = (0.1*(sf_anchor.SFOC*sf_anchor.SC/m_S)*m_SO2*sf_anchor.
↪ENG_KW*(1/60) + sf_anchor.AE*(sf_anchor.SFOC_AE*SC_diesel/100/
↪m_S)*m_SO2*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['CO2'] = (0.1*(sf_anchor.SFOC*sf_anchor.CC/m_C)*m_CO2*sf_anchor.
↪ENG_KW*(1/60) + sf_anchor.AE*(sf_anchor.SFOC_AE*CC_diesel/100/
↪m_C)*m_CO2*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['NOx'] = (0.1*sf_anchor.ef_NOx*sf_anchor.ENG_KW*(1/60) + sf_anchor.
↪AE*(45*rpm**-0.2)*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['PM'] = (0.1*sf_anchor.ENG_KW*(0.455*(0.1)**2-0.17*(0.1)+1.28)*((0.
↪312*sf_anchor.SC)+(0.244*sf_anchor.SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) +␣
↪sf_anchor.AE*sf_anchor.AUX_KW*(0.455*sf_anchor.AE**2-0.17*sf_anchor.AE+1.
↪28)*((0.312*SC_diesel/100)+(0.244*SC_diesel/
↪100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_anchor = sf_anchor.drop(columns = ['inport', 'AE', 'Fuel', 'SC', 'CC',␣
↪'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', 'ef_NOx'])
```

7.5 Hotelling emissions

[ ]:
```
# Hoteling emissions
## This section computes the emissions of vessels in the "hoteling" stage per␣
↪minute
```

```
## Main Engine loads are estimated at 20% for all vessels, whereas 70% is␣
↪assigned to auxiliary engines on passenger and
## tanker vessels, and 40% is assigned to those on cargo vessels
## Main engines are considered to burn either MGO or LNG, whereas all auxiliary␣
↪engines burn MGO

sf_hotelling = sf_hotelling.merge(p, how = 'left', on = ['IMO','IMO'])
sf_hotelling = sf_hotelling.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp'])
sf_hotelling.SC.loc[sf_hotelling.Fuel == 'LNG'] = SC_lng/100
sf_hotelling.CC.loc[sf_hotelling.Fuel == 'LNG'] = CC_lng/100
sf_hotelling.SC.loc[sf_hotelling.Fuel != 'LNG'] = SC_diesel/100
sf_hotelling.CC.loc[sf_hotelling.Fuel != 'LNG'] = CC_diesel/100
sf_hotelling.AE.loc[sf_hotelling.group == 'Passenger'] = 0.7
sf_hotelling.AE.loc[sf_hotelling.group == 'Tankers'] = 0.7
sf_hotelling.AE.loc[sf_hotelling.group == 'Cargo'] = 0.4
sf_hotelling.ENG_KW.loc[sf_hotelling.AUX_KW != 0 ] = 0
```

[ ]:
```
sf_hotelling['SFOC'] = sf_hotelling.SFC*(0.455*(EL*0.2)**2-0.17*(EL*0.2)+1.28)
sf_hotelling['SFOC_AE'] = SFOC_AE*(0.455*(sf_hotelling.AE)**2-0.
↪17*(sf_hotelling.AE)+1.28)
sf_hotelling['FC'] = (0.2*sf_hotelling.ENG_KW*sf_hotelling.SFOC*(1/60) +␣
↪sf_hotelling.AE*sf_hotelling.AUX_KW*SFOC*(1/60))*1e-6
sf_hotelling['SO2'] = (0.2*(sf_hotelling.SFOC*sf_hotelling.SC/
↪m_S)*m_SO2*sf_hotelling.ENG_KW*(1/60) + sf_hotelling.AE*(sf_hotelling.
↪SFOC_AE*SC_diesel/100/m_S)*m_SO2*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['CO2'] = (0.2*(sf_hotelling.SFOC*sf_hotelling.CC/
↪m_C)*m_CO2*sf_hotelling.ENG_KW*(1/60) + sf_hotelling.AE*(sf_hotelling.
↪SFOC_AE*CC_diesel/100/m_C)*m_CO2*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['NOx'] = (0.2*sf_hotelling.ef_NOx*sf_hotelling.ENG_KW*(1/60) +␣
↪sf_hotelling.AE*(45*rpm**-0.2)*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['PM'] = (0.2*sf_hotelling.ENG_KW*(0.455*(0.2)**2-0.17*(0.2)+1.
↪28)*((0.312*sf_hotelling.SC)+(0.244*sf_hotelling.
↪SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_hotelling.AE*sf_hotelling.
↪AUX_KW*(0.455*sf_hotelling.AE**2-0.17*sf_hotelling.AE+1.28)*((0.
↪312*SC_diesel/100)+(0.244*SC_diesel/100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/
↪60))*1e-6
sf_hotelling = sf_hotelling.drop(columns = ['inport', 'AE', 'Fuel', 'SC', 'CC',␣
↪'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', 'ef_NOx'])
```

7.6 Data appending

[ ]:
```
# Emission data appending
## This section appends all 4 inventories per stage into a single consolidated␣
↪one

e = sf_cruising.append(sf_maneuvering, ignore_index = True)
```

```
e = e.append(sf_anchor, ignore_index = True)
e = e.append(sf_hotelling, ignore_index = True)

print("data length:", len(e))
e.head()
```

8 Results

8.1 Emission - Total

```
[ ]: # Fuel consumption and emissions
     ## This section computes the total fuel consumption and emissions (tons)

     e['date'] = e.date.dt.round('1d')
     ef = e.drop(columns = ['date','status','speed','lat','lon','group','Name'])
     ef.sum(axis = 0)
```

8.2 Emission - Day

```
[ ]: # Fuel consumption and emissions per day
     ## This section computes the fuel consumption and emissions (tons) per day

     e.groupby(['date'])['FC','SO2','CO2','NOx','PM'].sum()
```

8.3 Emission - Shiptype

```
[ ]: # Fuel consumption and emissions per shiptype
     ## This section computes the fuel consumption and emissions (tons) per shiptype

     e.groupby(['group'])['FC','SO2','CO2','NOx','PM'].sum()
```

8.4 Emission - Ship

```
[ ]: # Fuel consumption and emissions per ship
     ## This section computes the fuel consumption and emissions (tons) per ship

     e.groupby(['Name'])['FC','SO2','CO2','NOx','PM'].sum().sort_values(by = ['FC'],␣
     ↪ascending = False )
```

9 Plot

9.1 Fuel consumption

[ ]:
```
# Fuel consumption and emissions per ship
## This section computes the fuel consumption and emissions (tons) per ship

fig, ax = plt.subplots(figsize=(15,7))
e.groupby(['date'])['FC'].sum().plot(ax=ax).legend(['Total'])
plt.ylabel('tons')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
plt.title('Fuel consumption')
plt.grid()
plt.show()
```

[ ]:
```
fig.savefig(projPath+'figures/'+'FC.eps', format='eps')
```

[ ]:
```
ylim=[80,480]
```

9.2 CO2

[ ]:
```
# Fuel consumption and emissions per ship
## This section computes the fuel consumption and emissions (tons) per ship

fig, ax = plt.subplots(figsize=(15,7))
e.groupby(['date'])['CO2'].sum().plot(ax=ax).legend(['Total'])
plt.ylabel('tons')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
plt.title('CO2')
plt.grid()
plt.show()
```

[ ]:
```
fig.savefig(projPath+'figures/'+'CO2.eps', format='eps')
```

9.3 SO2

[ ]:
```
# Fuel consumption and emissions per ship
## This section computes the fuel consumption and emissions (tons) per ship

fig, ax = plt.subplots(figsize=(15,7))
e.groupby(['date'])['SO2'].sum().plot(ax=ax).legend(['Total'])
plt.ylabel('tons')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
plt.title('SO2')
plt.grid()
plt.show()
```

```
[ ]: fig.savefig(projPath+'figures/'+'SO2.eps', format='eps')
```

9.4 NOx

```
[ ]: # Fuel consumption and emissions per ship
     ## This section computes the fuel consumption and emissions (tons) per ship

     fig, ax = plt.subplots(figsize=(15,7))
     e.groupby(['date'])['NOx'].sum().plot(ax=ax).legend(['Total'])
     plt.ylabel('tons')
     ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
     plt.title('NOx')
     plt.grid()
     plt.show()
```

```
[ ]: fig.savefig(projPath+'figures/'+'NOx.eps', format='eps')
```

9.5 PM

```
[ ]: # Fuel consumption and emissions per ship
     ## This section computes the fuel consumption and emissions (tons) per ship

     fig, ax = plt.subplots(figsize=(15,7))
     e.groupby(['date'])['PM'].sum().plot(ax=ax).legend(['Total'])
     plt.ylabel('tons')
     ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
     plt.title('PM')
     plt.grid()
     plt.show()
```

```
[ ]: fig.savefig(projPath+'figures/'+'PM.eps', format='eps')
```

```
[ ]:
```

International Association of Maritime Universities

Meiwa Building 8F, 1-15-10 Toranomon, Minato-ku, Tokyo 105-0001, Japan

Tel : 81-3-6257-1812 E-mail : info@iamu-edu.org URL : http://www.iamu-edu.org

ISBN No. 978-4-907408-40-4