Assessing the Global COVID19 Impact on Air Transport with Open Data

Rainer Koelle
Performance Review Unit
EUROCONTROL
Brussels (Belgium)
rainer.koelle@eurocontrol.int

Fabio Lourenco Carneiro Barbosa
Subdepartment of Operations
DECEA
Rio de Janeiro (Brazil)
barbosaflcb@fab.mil.br

***Abstract*—This paper approaches the impact of the pandemic as a massive service disruption of the pre-pandemic global connectivity and regional air transport networks. In particular, the project aims to provide data analytical evidence for policy success and transformation of the air transportation system. As an aspirational goal, the industry aims to recover in a “greener” manner. The project builds on openly available data sets. The paper will be produced in a reproducible manner making the data, code, and its processing available to interested reseachers and practitioners. The open assessment will provide policy makers with a tool to assess the reaction to local or regional measures.**

```
## set bookdown specs/defaults for high quality output
#---------- check for the settings

## theme default for ggplot
theme_set(
  theme_minimal()
  )

###Preparatory codes

#If someone needs filters below, here we can control any selective sample
#Filtering Brazilian and European data to reduce the sample (and csv) size
#Currently, those filters are not in use in the code, they are here just in case.
bra_10_apts <- c("SBBR", "SBGR", "SBSP", "SBKP", "SBRJ", "SBGL", "SBCF", "SBSV", "SBPA", "SBCT
eur_apts <- c("EHAM","LFPG","EGLL","EDDF","EDDM","LEMD","LIRF","LEBL","EGKK","LSZH")
study_airports <- c(bra_10_apts, eur_apts)

###Preparing airport file
#NOTE: the airports.csv file below is the downloaded file from www.ourairports.org.
airports <- read_csv("data-raw/airports.csv") %>% mutate(continent = case_when(is.na(continent

#There are missing airports and too much variables
apt_countries <- airports %>% transmute(ICAO = ident, CTRY = iso_country) %>% add_count(ICAO =

#If you need to write
#write_csv(apt_countries, "./data/apt_countries.csv")
#apt_countries is ready.

# Associate the regions
a <- airports %>% filter(continent == "EU") %>% select(iso_country) %>% unique()
eur_countries <- a$iso_country
#eur_countries is ready.
```

```
#CURRENT COMMENTS AND TO-DO'S
# I HAVE DOWNLOADED 3 FILES FOR NOW (APR/19, APR/2
#IT'S IN THE DATA-RAW FOLDER (NOT SHARED WITH GITH
```

I. Introduction

This paper is heavily informed by the work of (Strohmeier et.al. 2021).

For many years, many concerns of the global air traffic management community has been directed to the evident problem of imbalances between capacity and demand. The pressing, increasing demand for air transport registered in the last decade not only has already produced challenging delay management practices, but also fostered projections of even worse scenarios. EUROCONTROL (____), for example, argued that delays in Europe could reach up to 20 minutes per flight in 2040, in stark contrast to the 12 minutes per flight, as registered in 2016.

In the above scenario, many disturbances on the air navigation system could represent a real threat to multiple stakeholders. Events such as extreme bad weather, unexpected interruptions of air navigation services, changes in regulatory framework and others: all of those inputs could promote even more delay and its propagation effects. That is why the concept of resilience in ATM system became similarly relevant in the agenda during the same period. Arguably, a resilient ATM system could mitigate the negative effects of excessive demands on insufficient capacity and their respective constraints and bottlenecks.

However, the recent COVID-19 crisis posed a completely different, unexpected, and inverted challenge. Demand for air transport dropped as low as 90% of the previous “normal” in many places. Where the lack of capacity was previously the issue, now the lack of demand threatened the ATM system stability. In the financial perspective, airlines and airports had to deal with an unprecedented decrease in incomes. As a result, air navigation providers collected less fees for their services, due to significantly fewer flights. In the operational perspective, pilots and air traffic controllers practiced less. The problems and obstacles developed into many other dimensions.

Hence, the current scenario is a proper moment to further investigate the concept of resilience.

```
# Problem Statement

# The problem is that, currently, the concept
# of resilience is mostly directed to recovery
# against delay propagation after negative
# disturbances. However, the current scenario
# poses an inverted challenge, of very low
# delays due to low demand against surplus
# capacity. Therefore, there is room for
# enlarging the comprehension of the concept
# of resilience in ATM systems.

# # Purpose Statement
#
# ???The purpose of this research is to
# investigate additional dimensions in which
# resilience could be measured, in addition
# to the current framework of delay analysis.
#
# Research Question
#
# ???How can we enlarge the concept of
# resilience, so that it is applicable to
# scenarios of low traffic?
#
# ???Research Question:
# ???RQ1.What was the impact of the pandemic on ATM resilience?
# ???  RQ1.1 How resilience can be modeled in a low-demand scenario?
# ???  RQ1.2 How resilient were different ATM systems worldwide?
```

This paper approaches the impact of the pandemic as a massive service disruption of the pre-pandemic global connectivity and regional air transport networks. In particular, the project aims to provide data analytical evidence for policy success and transformation of the air transportation system. As an aspirational goal, the industry aims to recover in a "greener" manner. To date, no assessment of this transformational aspects has been conducted.

- data-analytical approach - using open data / freely available (tbd: validated against organisational data)
- ???RQ1.1 = through a qualitative analysis of previous proposed models
- ???RQ1.2 = through a quantitative analysis of open data

The contribution of this paper are

- conceptualisation of the COVID-19 impact on air transportation as a resilience problem;
- assessing the impact on the basis of open data
- identification of patterns and/or measures to describe and quantify/evaluate the level of recovery (or disruption)

II. Background

A. COVID-19 & Air Transportation

B. Resilience

EUROCONTROL (2009): first definition of resilience in ATM context – "Resilience is the intrinsic ability of a system to adjust its functioning prior to, during, or following changes and disturbances, so that it can sustain required operations under both expected and unexpected conditions."

Gluchshenko (2012):
Definitions for Resilience, robustness, disturbance, stress, and perturbation Proposition for a framework of different levels of stress/perturbations Proposition of metrics for resilience (both quantitative and qualitative)

Gluchshenko (2013): repeats the previous ideas and adds a performance-based approach as well as an algorithm to investigate resilience

Project Resilience 2050 (Jun/2012 + 43 months) – includes the previous definitions and other technical tasks. However, it evolves the way to measure resilience. Now, not only the time of deviation and time of recovery is considered. The project measures it as the relative difference of rate of delays correlation, or R = (ax1 – dx1)/dx1 – it has no unit, it's the difference between two pearson correlations.

Koelle (2015): proposes to address resilience as a situation management and state-oriented problem. Through two case studies, argued that "there is a lack of fit of the current operational ANS performance indicators to address impact of disruptions as they are primarily based on actual timestamps or transition times."

C. <if we need to fill space> Crowd-Sourced Data Collection

III. Method/Materials

A mixed-method approach, based on:

a) to answer RQ1.1, a qualitative analysis of previous models to develop acute low-demand as a disturbance
b) to answer RQ1.2, a quantitative analysis of open data, to observe (or not) different levels/stages of stress/recovery, which could indicate different "more" or "less" resilience to the disturbances

A. Open-source Data

This study builds on publicly available data. Opensky Network collects crowdsourced air traffic data from more than 2500 feeders (sensor stations). To support the process of illustrating and studying the impact of the COVID pandemic on air traffic demand, a flight-by-flight dataset is provided on a monthly basis (Olive, Strohmeier, and Lübbe 2021). The data spans the period since 1. January 2019. Fig. 1 shows the number of daily flights tracked by Opensky Network globally.

```
daily_tfc <- read_csv("./data/daily_osn.csv")

daily_tfc %>%
  ggplot(mapping = aes(x = DATE, y = FLTS)) +
```

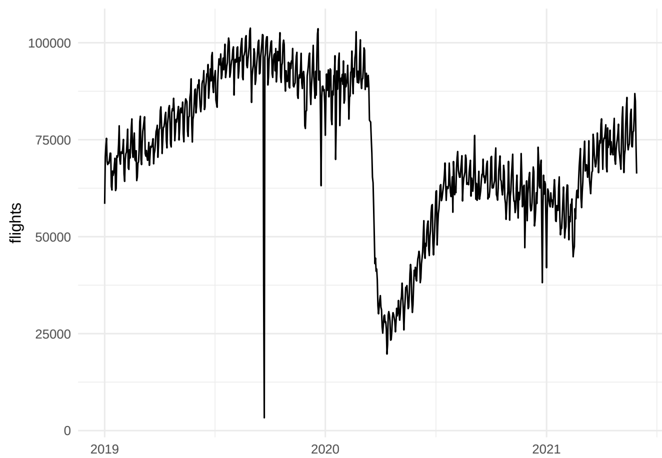


Fig. 1. Daily flights tracked by Opensky Network

```
  geom_line() +
  labs(x = NULL, y = "flights")
```

IV. Results/Discussion

1.1

a) Resilience can be measured as a function of time - the smaller the relationship between time of stress and the time of recovery, more resilient a system is.

1.2 how to use open data to “see” resilience?

1.2.1 Gather and prepare data

```
#Reading raw data

source("./R/list_apt_files.R")

#Here I will assign only one month - "202105"# If you want to include a full year, just assign
year <- "2021"
file_names <- list_apt_files(.year = year)
open_sky <- map_dfr(file_names, read_csv)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   callsign = col_character(),
##   number = col_character(),
##   icao24 = col_character(),
##   registration = col_character(),
##   typecode = col_character(),
##   origin = col_character(),
##   destination = col_character(),
##   firstseen = col_datetime(format = ""),
##   lastseen = col_datetime(format = ""),
##   day = col_datetime(format = ""),
##   latitude_1 = col_double(),
##   longitude_1 = col_double(),
##   altitude_1 = col_double(),
##   latitude_2 = col_double(),
##   longitude_2 = col_double(),
##   altitude_2 = col_double()
## )
##
##
## -- Column specification ------------------------
## cols(
##   callsign = col_character(),
##   number = col_character(),
##   icao24 = col_character(),
##   registration = col_character(),
##   typecode = col_character(),
##   origin = col_character(),
##   destination = col_character(),
##   firstseen = col_datetime(format = ""),
##   lastseen = col_datetime(format = ""),
##   day = col_datetime(format = ""),
##   latitude_1 = col_double(),
##   longitude_1 = col_double(),
##   altitude_1 = col_double(),
##   latitude_2 = col_double(),
##   longitude_2 = col_double(),
##   altitude_2 = col_double()
## )
##
##
## -- Column specification ------------------------
## cols(
##   callsign = col_character(),
##   number = col_character(),
##   icao24 = col_character(),
##   registration = col_character(),
##   typecode = col_character(),
##   origin = col_character(),
##   destination = col_character(),
##   firstseen = col_datetime(format = ""),
##   lastseen = col_datetime(format = ""),
##   day = col_datetime(format = ""),
##   latitude_1 = col_double(),
##   longitude_1 = col_double(),
##   altitude_1 = col_double(),
##   latitude_2 = col_double(),
##   longitude_2 = col_double(),
##   altitude_2 = col_double()
## )
##
##
## -- Column specification ------------------------
## cols(
##   callsign = col_character(),
##   number = col_character(),
##   icao24 = col_character(),
##   registration = col_character(),
##   typecode = col_character(),
##   origin = col_character(),
##   destination = col_character(),
```

```
##   firstseen = col_datetime(format = ""),
##   lastseen = col_datetime(format = ""),
##   day = col_datetime(format = ""),
##   latitude_1 = col_double(),
##   longitude_1 = col_double(),
##   altitude_1 = col_double(),
##   latitude_2 = col_double(),
##   longitude_2 = col_double(),
##   altitude_2 = col_double()
## )
##
##
## -- Column specification --------------------------------------------------------
## cols(
##   callsign = col_character(),
##   number = col_character(),
##   icao24 = col_character(),
##   registration = col_character(),
##   typecode = col_character(),
##   origin = col_character(),
##   destination = col_character(),
##   firstseen = col_datetime(format = ""),
##   lastseen = col_datetime(format = ""),
##   day = col_datetime(format = ""),
##   latitude_1 = col_double(),
##   longitude_1 = col_double(),
##   altitude_1 = col_double(),
##   latitude_2 = col_double(),
##   longitude_2 = col_double(),
##   altitude_2 = col_double()
## )
```

```r
#Selecting relevant variables

fb <- open_sky %>% transmute(ADEP = origin, ADES = destination, TYPE = as.factor(typecode), DA
                             #, ACFT_ID = aircraft_uid
                             )



#Easily "dropping NA's# - this can be further sofisticated

fb1 <- fb %>% drop_na()
fb1
```

```
## # A tibble: 3,338,649 x 5
##    ADEP  ADES  TYPE  DATE       CALL
##    <chr> <chr> <fct> <date>     <chr>
##  1 KJFK  LSGG  B788  2021-01-01 ETH726
##  2 KMIA  KMIA  B763  2021-01-01 LCO1108
##  3 VHHH  71KY  G650  2021-01-01 ABW9515
##  4 OMDM  YSSY  A343  2021-01-01 ASY052
##  5 KLAX  EDDF  B77L  2021-01-01 CSN461
##  6 EBLG  EBLG  B744  2021-01-01 ATG6652
##  7 KORD  EHAM  B77L  2021-01-01 CSN5203
##  8 FAOR  OMDB  B738  2021-01-01 KQA304
##  9 YSSY  OMDB  B77W  2021-01-01 UAE415
## 10 YSSY  WSSS  A332  2021-01-01 ALK302
## # ... with 3,338,639 more rows
```

```r
#Joining to ADEP
fb2 <- left_join(fb1, apt_countries, by = c("ADEP"
#Joining to ADES
fb3 <- left_join(fb2, apt_countries, by = c("ADES"

#Check  NA's
colSums(is.na(fb3))
```

```
##      ADEP ADEP_CTRY      ADES ADES_CTRY      TY
##         0      1271         0      1446
```

```r
# Very few NA's - it's safe to drop and factor now

fb4 <- fb3 %>% drop_na() %>% mutate(ADEP_CTRY = as

#NOTE: Here we can adjust the European countries t
# Currently in european countries: "GB" "AD" "ES"

base_dataset <- fb4 %>% mutate(ADEP_REG = as.facto
                                    ADEP_CTRY == "
                                    ADEP_CTRY %in%
                                    TRUE ~ "Other"
                      mutate(ADES_REG = as.factor(case_wh
                                    ADES_CTRY == "
                                    ADES_CTRY %in%
                                    TRUE ~ "Other"
colSums(is.na(base_dataset))
```

```
##      ADEP ADEP_CTRY  ADEP_REG      ADES ADES_CT
##         0         0         0         0
##      CALL
##         0
```

```r
# No NA's - yaaaayy!!
glimpse(base_dataset)
```

```
## Rows: 3,336,679
## Columns: 9
## $ ADEP      <chr> "KJFK", "KMIA", "VHHH", "OMDM
## $ ADEP_CTRY <fct> US, US, HK, AE, US, BE, US, Z
## $ ADEP_REG  <fct> US, US, Other, Other, US, EU,
## $ ADES      <chr> "LSGG", "KMIA", "71KY", "YSSY
## $ ADES_CTRY <fct> CH, US, US, AU, DE, BE, NL, A
## $ ADES_REG  <fct> EU, US, US, Other, EU, EU, EU
## $ TYPE      <fct> B788, B763, G650, A343, B77L,
## $ DATE      <date> 2021-01-01, 2021-01-01, 2021
## $ CALL      <chr> "ETH726", "LCO1108", "ABW9515
```

```r
summary(base_dataset)
```

```
##      ADEP                ADEP_CTRY        ADEP_REG
##  Length:3336679      US      :2225791   BR   :  3
##  Class :character    AU      : 127008   EU   : 59
##  Mode  :character    DE      : 109611   Other: 47
```

```
##                        GB      :  84635    US
##                        FR      :  64874
##                        CA      :  45627
##                        (Other): 679133
##     ADES_CTRY          ADES_REG              TYPE
##   US      :2223845   BR    :  35937    B738
##   AU      : 127206   EU    : 597541    A320
##   DE      : 109350   Other: 479356     B737
##   GB      :  84605   US    :2223845    A321
##   FR      :  65219                     A319
##   CA      :  45727                     E75L
##   (Other): 680727                      (Other)
##       CALL
##   Length:3336679
##   Class :character
##   Mode  :character
##
##
##
##
```

```
#quick peek
base_dataset
```

```
## # A tibble: 3,336,679 x 9
##     ADEP  ADEP_CTRY ADEP_REG ADES  ADES_CTRY ADES_REG TYPE  DATE       CALL
##     <chr> <fct>     <fct>    <chr> <fct>     <fct>    <fct> <date>     <chr>
##   1 KJFK  US        US       LSGG  CH        EU       B788  2021-01-01 ETH726
##   2 KMIA  US        US       KMIA  US        US       B763  2021-01-01 LCO1108
##   3 VHHH  HK        Other    71KY  US        US       G650  2021-01-01 ABW9515
##   4 OMDM  AE        Other    YSSY  AU        Other    A343  2021-01-01 ASY052
##   5 KLAX  US        US       EDDF  DE        EU       B77L  2021-01-01 CSN461
##   6 EBLG  BE        EU       EBLG  BE        EU       B744  2021-01-01 AIS6652
##   7 KORD  US        US       EHAM  NL        EU       B77L  2021-01-01 CSN5203
##   8 FAOR  ZA        Other    OMDB  AE        Other    B738  2021-01-01 KQA304
##   9 YSSY  AU        Other    OMDB  AE        Other    B77W  2021-01-01 UAE415
##  10 YSSY  AU        Other    WSSS  SG        Other    A332  2021-01-01 ALK302
## # ... with 3,336,669 more rows
```

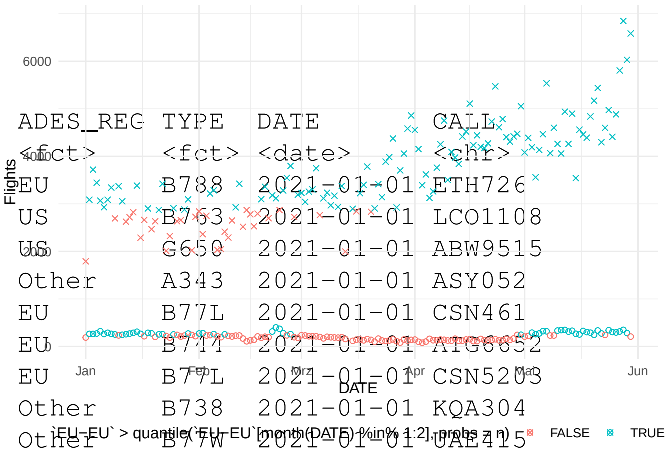
```
#
#
#
#NOW IT'S TIME TO LOOP TO OTHER MONTHS AND BIND THEM ALL!
#
#
#
```

```
#quick first look
temp1 <- base_dataset %>% group_by(DATE, ADEP_REG, ADES_REG) %>% summarize(FLIGHTS = n(), .gro
  filter(ADEP_REG %in% c("BR", "EU"), ADES_REG %in% c("BR", "EU")) %>%
  mutate(ROUTE = paste(ADEP_REG, ADES_REG, sep = "-"), .keep = "unused", .before = "FLIGHTS") %
  pivot_wider(names_from = ROUTE, values_from = FLIGHTS)
temp1
```

```
## # A tibble: 149 x 5
##     DATE        `BR-BR` `BR-EU` `EU-BR` `EU-EU`
```

```
##:2223845
##      <date>        <int>   <int>   <int>   <int>
##   1 2021-01-01       189       8       7    1792
##   2 2021-01-02       265       7      10    3088
##   3 2021-01-03       264      12       6    3721
##   4 2021-01-04 DATE  273       6      11    3446
##   5 2021-01-05 Min.   :2021-01-01  5      11    3068
##   6 2021-01-06 1st Qu.:2021-02-14 10       7    2930
##   7 2021-01-07 Median :2021-03-26  8      11    3088
##   8 2021-01-08 Mean   :2021-03-22  3       1    3343
##   9 2021-01-09 3rd Qu.:2021-04-29  7      11    2695
##  10 2021-01-10 Max.   :2021-05-30  7       7    3371
## # ... with 139 more rows
```

```
#First stab at different levels of traffic

n <- 0.5
temp1 %>% ggplot(aes(x = DATE)) +
  geom_point(aes(y = `EU-EU`, color = `EU-EU` > qua
  geom_point(aes(y = `BR-BR`, color = `BR-BR` > qua
  labs(y  = "Flights") +
  theme(legend.position = "bottom")
```




```
#Normalized by the median of the last 3 months of
temp2 <- temp1 %>% pivot_longer(cols = 2:5, names_
```

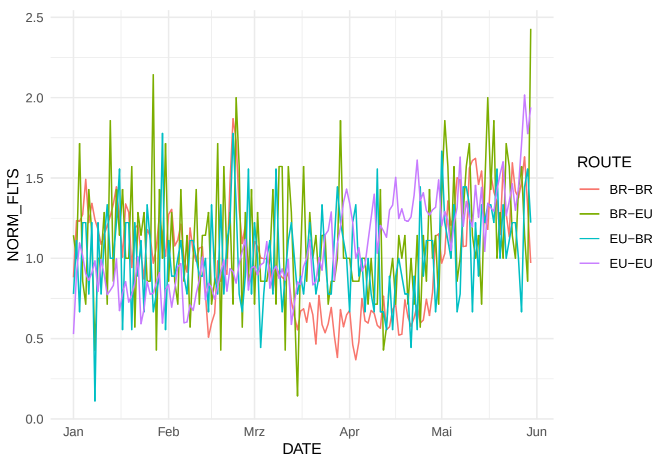
```
## Warning in month(DATE) %in% month(last(DATE)) -
##        ist kein Vielfaches der Länge des kürzere

## Warning in month(DATE) %in% month(last(DATE)) -
##        ist kein Vielfaches der Länge des kürzere

## Warning in month(DATE) %in% month(last(DATE)) -
##        ist kein Vielfaches der Länge des kürzere

## Warning in month(DATE) %in% month(last(DATE)) -
##        ist kein Vielfaches der Länge des kürzere
```

```
temp2
```

```
## # A tibble: 596 x 5
## # Groups:   ROUTE [4]
```

```
##     DATE       ROUTE FLIGHTS MOVING_MEDIAN NORM_FLTS
##     <date>     <chr>   <int>         <dbl>     <dbl>
##   1 2021-01-01 BR-BR     189          214.     0.881
##   2 2021-01-01 BR-EU       8            7      1.14
##   3 2021-01-01 EU-BR       7            9      0.778
##   4 2021-01-01 EU-EU    1792         3397      0.528
##   5 2021-01-02 BR-BR     265          214.     1.24
##   6 2021-01-02 BR-EU       7            7      1
##   7 2021-01-02 EU-BR      10            9      1.11
##   8 2021-01-02 EU-EU    3088         3397      0.909
##   9 2021-01-03 BR-BR     264          214.     1.23
## 10 2021-01-03 BR-EU      12            7      1.71
## # ... with 586 more rows
```

```
temp2 %>% ggplot(aes(x = DATE)) +
  geom_line(aes(y = NORM_FLTS, color = ROUTE)) +
  theme_minimal()
```



1.2.2

V. Conclusion

Acknowledgment

References

Olive, Xavier, Martin Strohmeier, and Jannis Lübbe. 2021. "Crowdsourced air traffic data from The OpenSky Network 2020." Zenodo. https://doi.org/10.5281/zenodo.4893103.

Strohmeier, Martin, Xavier Olive, Jannis Lübbe, Matthias Schäfer, and Vincent Lenders. 2021. "Crowdsourced Air Traffic Data from OpenSky Network 2019-2020." *Earth Systems Science Data* 13: 357–66. https://doi.org/10.5194/essd-13-357-2021.