Some Initial Gaming with Fuel Burn

2022-07-12

# Introduction

This document is a quick hack based on an even quicker web research.

There exists a variety of fuel burn estimation approaches, methods, and sources. However, recent discussions in bi- or multi-regional benchmarking setups did not support the identification of a “quick and dirty” look-up table to accommodate the conversion of operational inefficiencies into fuel burn.

The following is drafted for discussion and exploration purposes.

# Fuel Burn Estimation

Conceptually, there are 2 major approaches to estimate the fuel burn:

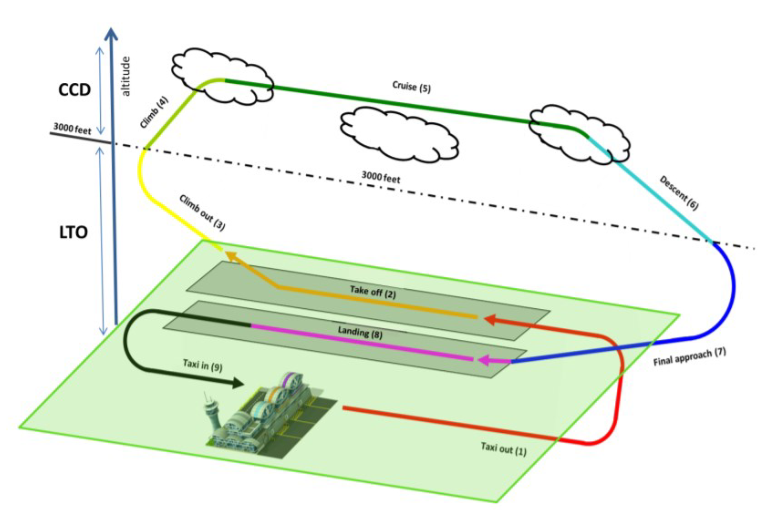
* kinematic energy models - these models are based on addressing thrust/power setting during different flight phases. These models support the estimation of fuel burn (and other emissions) for flight segments. The principal approach is to estimate the fuel burn based on kinematic considerations over the segment. The fundamental input - next to model specific parameters or model processing modules - is a flight trajectory. There are constraints on the openess and accessibility of these models for performance benchmarking purposes.
* compact flight models = these models typically assume a certain population of flights or flight profiles (e.g. aerodrome pairs). The fuel burn (or other emissions) is then provided on the basis of averages over stages of the flight. Also here no ready to plug and play model has been identified.

A third thread of work revolves around engine certification derived estimates. This group of models builds on (partly) openly available data bases from ICAO, i.e. ICAO engine emission data base, ICAO carbon emission calculator. The principal approach is here to map the engine certification data to aircraft types and use this as input to so-called payload-range diagrams (~ mapping of the aircraft weight to its associated range). The range based approach serves as a proxy for higher fidelity based trajectory data. It must be noted that some of these models tap also into complex emission models to map the payload-range to the associated emissions.

The latter approach shows some merit for operational performance purposes as it balances the need for trajectory processing (high accuracy in terms of segmentation) with reasonable assumptions over a sub-population of flights. This could allow to map additional time indicators with estimates for fuel burn/emissions. Some of the work surrounding this area is available openly.

# LTO and CCD

knitr::include\_graphics("./figures/LTO-CCD.png")



Standards for certification of emissions produced by aircraft engines are determined in Volume II of Annex 16 of the International Civil Aviation Organization. The Standard focuses on the measurement of carbon monoxide (CO), unburned hydrocarbons (HC), nitro- gen oxides (NOX), and smoke. Volume II of Annex 16 also sets a regulatory limit on the concentration of the mentioned emission products during the landing and takeoff cycle (LTO).

The LTO cycle comprises the landing, taxiing, and take-off from/up to an altitude of 3000ft. According to the certification process, engines must be tested at various thrust settings representing the operations during the LTO cycle.

The associated data point conventions are

thrust\_mode\_tbl <-   
 tibble::tribble(  
 ~"Operating Mode", ~"Engine Thrust (%)", ~"Operating Time (min)"  
 ,"Taxi-out", 7, 7.0  
 ,"Take-off", 100, 0.7  
 ,"Climb-out", 85, 2.2  
 ,"Approach", 30, 4.0  
 ,"Taxi-in", 7, 19  
 )  
  
thrust\_mode\_tbl %>% flextable::flextable()

| Operating Mode | Engine Thrust (%) | Operating Time (min) |
| --- | --- | --- |
| Taxi-out | 7 | 7.0 |
| Take-off | 100 | 0.7 |
| Climb-out | 85 | 2.2 |
| Approach | 30 | 4.0 |
| Taxi-in | 7 | 19.0 |

Dependent on the work estimates for the complete LTO cycle are based on the associated “standard” times in the operating mode. However, the table enables also to map actual observed movement times to the “standard” thrust setting (as established by ICAO) and derive associated fuel flow/burn estimates.

The aforementioned web search provided the following interesting candidates

* ACERT - Airport carbon and emission reporting tool is an initiative by ACI to support airports in the reporting of Greenhouse Gas Emissions in accordance with the ACI guidance. The latter is also recognised as a means of compliance with other national or international reporting requirements. The ACERT tool contains a pre-processed tab “EF Aircraft” that contains a mapping between aircraft types and fuel flow, i.e. for the taxi phase.

# Fuel Burn Estimation for Performance

## ACERT Taxi-Fuel Estimation

acert <- readxl::read\_xlsx("./data-analytic/ACERT\_6.0\_ACI\_Public\_build2216.xlsx", sheet = "EF Aircraft", skip = 2)  
  
dplyr::glimpse(acert)

## Rows: 176  
## Columns: 9  
## $ `Aircraft type` <chr> "A109", "A119", "A220-100/300", "Airbus 3~  
## $ `#/Eng` <chr> "2", "2", "2", "2", "2", "2", "2", "2", "~  
## $ EngType <chr> "H", "H", "J", "J", "J", "J", "J", "J", "~  
## $ `Typical or assigned engine` <chr> "PW206C", "PT6B-37", "PW1524G", "CF6-50A"~  
## $ `LTO Fuel (kg/LTO\*)` <chr> "55.691999999999993", "91.968000000000018~  
## $ `Taxi Fuel (kg/min taxi)` <chr> NA, NA, "9.6", "19.559999999999999", "24.~  
## $ `APU Fuel\r\n(kg/min)` <chr> NA, NA, "1.78", "4", "4", "1.777777777777~  
## $ `Data Code ICAO/ZRH` <chr> "H021", "H013", "20PW130", "3GE069", "2GE~  
## $ `Data Revision` <chr> "V5, 2017", "V5, 2017", "V6, 2020", "V5, ~

The associated ACERT emission factor table for aircraft provides already a mapping between aircraft type and “typical” engine and the respective fuel burn for a 7% taxi-thrust setting (LTO assumption).

# clean and package acert  
# --> more standard column names  
# --> coerce types  
acert <- acert %>%   
 rename(  
 AC\_TYPE = `Aircraft type`  
 ,NBR\_ENG = `#/Eng`  
 ,ENG\_TYPE= EngType  
 ,TYPICAL\_ENG = `Typical or assigned engine`  
 ,FUEL\_LTO = `LTO Fuel (kg/LTO\*)`   
 ,FUEL\_TAXI = `Taxi Fuel (kg/min taxi)`  
 ) %>%   
 mutate( FUEL\_LTO = as.numeric(FUEL\_LTO)  
 ,FUEL\_TAXI= as.numeric(FUEL\_TAXI)  
 )

For demonstration purposes we quickly map the table to our analytic data:

sbgr\_arrs <- readr::read\_csv(  
 "./data-analytic/BRA-EUR\_SBGR\_2019\_ARR.csv"  
 ,show\_col\_types = FALSE  
 ) %>%   
 mutate(across(.cols = c("AIBT","ALDT"), .fns = ymd\_hms)  
 ,TXIT = difftime(AIBT, ALDT, units = "min") %>% as.numeric) %>%   
 select(ADEP, ADES, FLTID, TYPE, ALDT, AIBT, TXIT)  
sbgr\_arrs

## # A tibble: 147,263 x 7  
## ADEP ADES FLTID TYPE ALDT AIBT TXIT  
## <chr> <chr> <chr> <chr> <dttm> <dttm> <dbl>  
## 1 SBRF SBGR TAM3563 A321 2019-01-01 00:06:32 2019-01-01 00:10:27 3.92  
## 2 SBCT SBGR TAM3003 A320 2019-01-01 00:10:31 2019-01-01 00:19:50 9.32  
## 3 SBFZ SBGR TAM3815 A321 2019-01-01 00:21:27 2019-01-01 00:27:28 6.02  
## 4 SBEG SBGR TAM3168 A321 2019-01-01 00:26:37 2019-01-01 00:39:40 13.0   
## 5 SKBO SBGR ONE851 A320 2019-01-01 00:31:20 2019-01-01 00:35:53 4.55  
## 6 GMMN SBGR RAM215 B788 2019-01-01 00:34:47 2019-01-01 00:41:23 6.6   
## 7 SBPA SBGR TAM4601 A320 2019-01-01 00:42:17 2019-01-01 00:48:14 5.95  
## 8 SBGL SBGR TAM3235 A320 2019-01-01 00:45:26 2019-01-01 00:48:53 3.45  
## 9 SBSV SBGR ONE6055 A320 2019-01-01 00:52:15 2019-01-01 00:59:51 7.6   
## 10 SBPA SBGR TAM4779 A321 2019-01-01 01:10:07 2019-01-01 01:13:30 3.38  
## # ... with 147,253 more rows

To utilise the mapping, we need to recode the aircraft types.  
Example:

acert2 <- acert %>%   
 mutate(TYPE =   
 case\_when(  
 AC\_TYPE == "Airbus 320" ~ "A320"  
 , TRUE ~ NA\_character\_  
 )  
 )  
  
acert2 <- acert2 %>% select(AC\_TYPE, TYPE, FUEL\_TAXI)  
  
acert2 <- acert2 %>% drop\_na()

With the lookup table we can map aircraft type/typical engine with our flight-by-flight tables.

CO2 emissions from aviation fuel are 3.15 grams per gram of fuel

tmp <- sbgr\_arrs %>%   
 inner\_join(acert2 %>% select(-AC\_TYPE)) %>%  
 mutate(TXIT\_FUEL = TXIT \* FUEL\_TAXI  
 ,TXIT\_CO2 = TXIT\_FUEL \* 3.15)

## Joining, by = "TYPE"

tmp[1:20,] %>% select(FLTID, TYPE, TXIT, FUEL\_TAXI, TXIT\_FUEL, TXIT\_CO2) %>% flextable::flextable()

| FLTID | TYPE | TXIT | FUEL\_TAXI | TXIT\_FUEL | TXIT\_CO2 |
| --- | --- | --- | --- | --- | --- |
| TAM3003 | A320 | 9.316667 | 11.28 | 105.092 | 331.0398 |
| ONE851 | A320 | 4.550000 | 11.28 | 51.324 | 161.6706 |
| TAM4601 | A320 | 5.950000 | 11.28 | 67.116 | 211.4154 |
| TAM3235 | A320 | 3.450000 | 11.28 | 38.916 | 122.5854 |
| ONE6055 | A320 | 7.600000 | 11.28 | 85.728 | 270.0432 |
| TAM8041 | A320 | 4.316667 | 11.28 | 48.692 | 153.3798 |
| ONE6057 | A320 | 1.700000 | 11.28 | 19.176 | 60.4044 |
| TAM4745 | A320 | 1.750000 | 11.28 | 19.740 | 62.1810 |
| TAM9105 | A320 | 3.933333 | 11.28 | 44.368 | 139.7592 |
| ONE8525 | A320 | 1.883333 | 11.28 | 21.244 | 66.9186 |
| TAM3441 | A320 | 6.000000 | 11.28 | 67.680 | 213.1920 |
| LAP1303 | A320 | 2.266667 | 11.28 | 25.568 | 80.5392 |
| TAM8049 | A320 | 2.300000 | 11.28 | 25.944 | 81.7236 |
| TAM4609 | A320 | 1.383333 | 11.28 | 15.604 | 49.1526 |
| ONE6395 | A320 | 5.483333 | 11.28 | 61.852 | 194.8338 |
| TPU917 | A320 | 4.616667 | 11.28 | 52.076 | 164.0394 |
| ONE6365 | A320 | 2.033333 | 11.28 | 22.936 | 72.2484 |
| ONE6323 | A320 | 3.266667 | 11.28 | 36.848 | 116.0712 |
| ONE6103 | A320 | 4.416667 | 11.28 | 49.820 | 156.9330 |
| ONE6169 | A320 | 3.083333 | 11.28 | 34.780 | 109.5570 |

Things to do to make this useful

* prepare lookup table ==> clean aircraft type to standard types, make sure to contain 3.15 or kerosene for multiplication