1. Introduction

1.1. In general
The development of aircraft and aircraft technology have recently been under the influence of contradictory requirements - on one hand, the aircraft should provide great efficiency and performance along with the least adverse impact on the environment (low levels of exhaust gases emission, noise and waste), and on the other hand, the reliability and safety would have to be increased. These improvements should be accompanied by low production and maintenance costs.

The parameter within this field, which can be improved by every airport is the technology of passenger handling, especially with regard to safety of the subsequent flight, which includes good balancing and technical equipping of the aircraft. Roughly, the technology at an airport can be divided into the technical part, dealing with equipping and servicing the aircraft and the part dealing with passenger and cargo handling. These activities are in fact, at the moment, separated, and are being carried out separately by single operation services at the airport. The recordings of these procedures are unified only in the official form used for balancing the aircraft which is filled in directly prior to the aircraft take-off, and which contains only those data that refer to changes in cargo, people and fuel, whereas the technical details regarding servicing and other operations performed on the aircraft are kept by the technical services.

This leads to the conclusion that this method of keeping the register (except in special cases) does not provide safe aircraft handling, since there are no data on the technical condition of the aircraft arriving into the airport, and therefore no guarantee for its technical condition on take off.

Criteria and methodology of determining the general flowchart for the calculation and evaluation of the aircraft balance

Janko Rados
University of Zagreb
Faculty of Transport and Traffic Engineering

Mihaela Mise
Ericsson department of software design, Zagreb

Frane Jelusi
University of Zagreb
Faculty of Transport and Traffic Engineering

This paper deals with the criteria and requirements in developing general autonomous software related to handling the aircraft at the airport. It gives an overview of almost all the influencing factors which are relevant to "processing" an aircraft both upon arrival as well as on departure. It gives a brief description of the conventional balancing method which is being used, of the latest advancement in the field, and it offers a concrete suggestion for improving the reliability of criteria and results in aircraft handling.

The main idea of this initial work is to unify all the necessary activities and to register them by one computer, from landing until take-off, including the computer communication with other airports and companies. Currently, the programs of certain air companies are being used and they have produced individual software in co-operation with the manufacturers only for certain types of aircraft which are currently employed by them.

Since the range of aircraft types landing at airports is growing, there is the need to find a universal program which can calculate the balance chart for each aircraft, based, of course, on the manufacturer-supplied design data.

The performance card, in the form of a servicing card and other technical documentation is kept exclusively by the air company which owns the aircraft and provides the transportation service, in agreement with the manufacturer, according to set international regulations defined by law. In case of an accident the air company is responsible for failures.

The aim of this paper is to unify all the activities involving the aircraft during its stay at the airport and their registration by the computer network connected not only within one airport but to all other airports being in contact. The paper deals for the moment only with the new method of aircraft balancing. In this way, all the data about a certain aircraft would be available to every airport of its landing, with all the changes during its operation. The data on the number of passengers, weight and distribution of cargo, and condition of the fuel tanks would be known before landing. Information would be available about the passengers in transit, those getting off, about baggage that has to proceed to new destination, and which baggage will reduce the load of the aircraft. Figure (1) presents the schematic function of such a system.
First, the tendency of the paper is to find out the autonomous universal program (algorithm) for aircraft balancing, which would form a part of the comprehensive program set as the objective of this work.

2. Method of algorithm design

2.1 The proposal for the algorithm of proper aircraft balancing software

The main characteristic of the conventional programming is that the main control loop is contained in the application itself. For instance, an editor reads a character, performs the given activities, reads then the next character, etc. Upon receiving a character which represents the user’s request for ending the task, the programme terminates. Figure 2 presents the flowchart of the conventional approach.

![Figure 2 Flowchart of the conventional approach](image)

Unlike the conventional approach of interactive programming, this control structure is changed in notification-based systems. The main control loop is contained in the Notifier, and not in the application. Notifier reads the events and calls various procedures which the application had previously associated with them using the Notifier. This control procedure is presented in Figure 3. Notifier operates as in fact as a control entity within the user process.

**Program flowchart**

Due to the way of using the aircraft balancing software, where constant user-computer interaction is necessary, the notification-based approach is the most appropriate solution. From the point of view of the user this means that the application (program) starts with the initial window. Upon starting the application, the control of the process is taken over by the Notifier. It registers all the events, e.g. clicking of the mouse, or moving the cursor using the keyboard. After registering the event (selection of the option) the procedure (function) is called which has been previously associated with this event by the Notifier.

The program would consist of the main program and a range of functions. All the boxes and the objects in them would be created in the main program, and the functions would be called depending on the user. The application can be represented as a range of overlapping windows and sub-windows which appear on the screen according to the user’s request. The flowchart of using the program is presented in Figure 4. Since drawing the flowchart in one piece would greatly exceed the format, it will be drawn in several phases.

![Figure 3 Notifier operates “as a control entity”](image)

![Figure 4 Range of the overplaning](image)
After having carried out the primary selection, the selected option is processed. In order to be further able to open the appropriate windows for the given type of aircraft, the global variable (e.g. x) is introduced as an identifier. If narrow-body a/c is selected, this variable will obtain value 1 (x=1), and when wide-body a/c is selected the variable is assigned value 2. The part of the flowchart related to narrow-body aircraft will be further presented in detail (Figure 5). The procedure is somewhat different for the wide-body aircraft, but the preliminary solution suggested for the narrow-body aircraft can also be applied to the wide-body aircraft.

Figure 5 Step of selecting options

By starting the program the following options appear on the display:

- NARROW BODY A/C
- WIDE BODY A/C
- END.

By clicking on the e.g. menu button NARROW BODY the following menu would be opened (Figure 6)

- FLIGHT INFORMATION (the flight data can be loaded and printed)
- PAX&LOAD INFORMATION (the data on passengers, luggage, cargo and post can be entered, printed and calculated)
- OPERATING DATA (operating data can be entered, printed and calculated)
- DRAW GRAPH (graphs and lines for the actual flight can be drawn)

Figure 6 Aircraft balancing

Figure 7 Flight information

Flight information
In this part of the program the flight-related data are entered using the keyboard:

- abbreviation of the take-off airport (ORIGINATOR)
- abbreviation of the destination airport (ADDRESS)
- flight number (OU)
- aircraft registration code (REG)
- aircraft type (TYPE)
- aircraft code number (CODE)
- cabin configuration (CONFIG)
- version of the crew (CREW VERSION)
- date (DATE)

Data that do not have to be entered using the keyboard but are loaded from the database are:

- Cry Operating Index
- Take-off Fuel
- Trip Fuel

The possible form of the display is given in Figure 7, and its flowchart is presented in Figure 7 and 8.

Flight information - narrow body A/C

<table>
<thead>
<tr>
<th>Flight information - narrow body A/C</th>
<th>Flight information - narrow body A/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Address:</td>
<td>2. Originator:</td>
</tr>
<tr>
<td>3. Date:</td>
<td>4. QV:</td>
</tr>
<tr>
<td>5. Type:</td>
<td>6. Code:</td>
</tr>
<tr>
<td>7. Reg:</td>
<td>8. Config:</td>
</tr>
<tr>
<td>8. Crew version:</td>
<td></td>
</tr>
<tr>
<td>9. DOI:</td>
<td></td>
</tr>
<tr>
<td>10. TOF:</td>
<td></td>
</tr>
<tr>
<td>11. TIF:</td>
<td></td>
</tr>
<tr>
<td>Menu</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td></td>
</tr>
<tr>
<td>Pax &amp; load info</td>
<td></td>
</tr>
<tr>
<td>Operating data</td>
<td></td>
</tr>
<tr>
<td>Draw graph</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7 Flight information
Operating data
The flowchart for the OPERATING DATA window is somewhat more complex than the previous flowchart for FLIGHT INFORMATION since it contains the part for the parameters calculation. The proposal for the display is presented in Figure 9.

Operating data
1. DOW:
2. MZFW:
3. MTOW:
4. MLW:
5. TOF:
6. TIF:
7. CV:
8. DOI:
9. IV:

Menu D

Figure 9 Operating data
This sub-window contains 9 spaces for entering the parameters including: Dry Operating Weight (DOW), Maximum Zero Fuel Weight (MZFW), Maximum Take-off Weight (MTOW), Maximum Landing Weight (MLW), Take-off Fuel (TOF), Trip Fuel (TIF), Crew Version (CV), Dry Operating Index (DOI), and change of index due to fuel (UI). These values are loaded from the database already on entering this window. If a value needs to be changed, the cursor is positioned on the value, it is deleted, the new value keyed in, and Enter pressed. The values that need to be calculated are the Actual Zero Fuel Weight (AZFW), Actual Take-off Weight (ATOW), Actual Landing Weight (ALW) and Actual Total Load (ATL). The flowchart for this option is presented in Figure 10, 11 and 12.
By pressing the buttons for printing the required option, the function is performed which was assigned to this button, e.g. AZFW printing option analyses first whether the values necessary for calculating AZFW have been entered. If this is not the case, the user is notified by a message on the window that certain data are missing. If all the data have been entered, AZFW can be calculated according to the formula and printed in the appropriate space. The same procedure can also be performed for all the other values that need to be calculated. For simplification, the variables in this flowchart are designated by the standard abbreviations. The MOW variable is the weight obtained by summing up the dry operating weight and the total fuel amount. The allowed landing weight is obtained as the minimum of the values of MOW, MALW and MTOW. The minimum value is obtained by first assuming that one of the values is minimal, e.g. MOW. Then, it is checked whether the difference between the assumed ALTOW (MOW) and MALW is greater than zero. If yes, then the minimum value is MALW. Otherwise, ALTOW and continues to be MOW. Finally, it is checked whether the actual value ALTOW is greater than MTOW. If so, MTOW is considered as ALTOW. From this window it is possible to enter any of the windows offered by the MENU options.

PAX & LOAD INFORMATION

By pressing the PAX&LOAD INFORMATION button, we can open the page for entering data regarding the aircraft type selected primarily in the first window (narrow-body or wide-body aircraft). The possible display of the window for narrow-body aircraft is presented in Figure 13.

![Figure 13 PAX-LOAD input information](image)

The window consists of two parts. One is for entering the data on passengers, and the data on cargo are entered into the other part. The passenger data include:

- the number of adult passengers
- the number of children
- the number of infants
- their distribution according to classes
- their distribution according to compartments

All the values are entered into the table by first positioning the cursor in the appropriate field and by pressing Enter after the value has been keyed in. Thus the value is input and assigned to the appropriate variable. By pressing the button TOTAL, all the values on the number of passengers are calculated, and by pressing the button PAX the value of the total number of passengers is shown. In the table containing the number of passengers distributed into compartments, a warning appears if the number of entered passengers exceeds the maximum. The division into D1 and D2 denotes the number of passengers for destination 1 (D1) and destination 2 (D2). The data required for filling the tables related to the cargo distribution are:

- weight of the cargo in transit
- weight of the baggage
- weight of the cargo
- weight of the post
- distribution of cargo according to the belly compartments.

All these values are entered for the destination 1 and destination 2. The weight of the baggage shows if B key is pressed and if the number of passengers has been entered. By pressing B, the function which multiplies the number of passengers with the assumed weight of baggage per passenger (14 kg) is called. The total cargo weight appears by pressing TOTAL. This window includes also the distribution of cargo regarding belly compartments. The total weight of the cargo regarding belly compartments appears by pressing the button TOTAL in the table DISTRIBUTION WEIGHT.

Fig. 14 shows the flowchart for information on passengers and cargo.
and end point of the straight line. An important part in writing the programming code consists in determining the distance in pixels for e.g. 100 kg of cargo. After having determined the distance d in pixels between two distributions, every subsequent line has x co-ordinate by d greater than the previous one. For easier reading, the lines can be of various thickness and colors. Apart from the two mentioned charts, the window also contains a table of cargo distribution for the three cases. The weights entered in PAX&LOAD INFORMATION are automatically entered into the table under ACT1 upon opening the window. If we want the computer to suggest an optimal cargo distribution, in which the position of the center of gravity would be most optimal, we have to select button OPT.

There is a possibility for re-distributing the cargo into belly compartments in the ACT2 part of the table. The lines (which refer to the actual flight) are drawn into the graph by pressing the key DRAW GRAPH. This enables drawing the lines for the optimal cargo distribution. By selecting the draw graph option for e.g. optimal cargo distribution, lines are drawn in both diagrams and the values of weights and numbers of passengers are entered in the table next to the graphs. The program allows lines to be of different colors for every single case (OPT, ACT1, ACT2). Button CLEAR can be used to delete the lines for the actual flight. The procedure assigned to this button opens a menu with the list of cases that need to be deleted (both lines and values entered in the table).

After having selected one of the variants which fulfills the conditions, we can print the Load Sheet. The procedure implemented in this button checks first whether all the weights are within the allowed limits and whether the loaded aircraft index is within the operational limits. Then the list is updated, and it has to be in accordance with the EDP (Electronic Data Processing) standards proposed by IATA.

The flowchart of using this window is presented in Figure 16, 17a and 17b.
Figure 16 Accordance EDP with standard proposes by IATA

Figure 17a Block diagram Step 12-14

The step of the flowchart describing the opening of the window and drawing the diagram may be "extended" to several phases. The sequence of phases is the following:

- drawing the straight lines representing the frames of sub-windows for drawing the Balance Chart and diagram W=f(I)
- drawing the straight lines of the co-ordinate network of diagram W=f(I) and the horizontal lines of the Balance Chart
- drawing short vertical lines of distribution in the Balance Chart
- drawing the lines of the constant amount of the Mean Aerodynamic Chord (MAC) in percentages
- drawing the lines of operational limits
- entering numerical values
- entering the textual data into the diagrams

In order to draw a line (in fact a segment of a line) with the appropriate function, it is necessary to enter the co-ordinates of the starting and end points. The precision of such a diagram depends on the accuracy of defining these co-ordinates. The step of the diagram which describes how the optimal lugga-
Figure 18a Optimal luggage distribution
Figure 18b Optimal luggage distribution
The logic of determining the optimal distribution can be roughly divided into two phases: determining the optimal values of weights for the belly compartments 1 and 4, and determining of the belly compartment contents. The optimal weights are determined by defining the difference between the DOI and the optimal amount of the loaded aircraft index. After that, the value of weight is determined that needs to be loaded into belly compartment 1 in order to reach the balance. If there is more cargo available, the undistributed cargo is loaded into both compartments. Of course, these values have to be checked constantly in order to remain lower than the maximum. In the second phase the belly compartment contents is selected by assuming that the weight in belly compartment 1 is cargo in transit Tr1, and the weight in belly compartment 4 is the rest of the cargo. Then the deviation from the optimal value is checked. If there is a shortage in the belly compartment 1, the whole cargo amount is added. The deviation from the optimal value is checked again. If there is still shortage in the belly compartment 1, the baggage is added, and if there is an excess then the cargo is re-distributed to belly compartments 1 and 4. The same procedure (in the case of shortage in belly compartment 1) is carried out for the post as well. A different sequence of adding weight may be selected, e.g. in case of a shortage in the belly compartment 1, the baggage or post rather than cargo may have been added to the cargo in transit.

A more detailed flowchart of drawing the lines referring to the actual flight is presented in Figure 19. After having drawn all the changes of index into the diagram due to loading of cargo and passengers, the vertical lines of constant loaded aircraft indexes with and without fuel take-off weight are drawn in the diagram W=f(l).

Symbols used in the flowchart are the following:

A1 - number of pixels for the index unit value
B - number of pixels from the center to the center of the parts of the upper diagram for neighboring compartments
A2 - number of pixels for the index unit value for compartment 1
W1 - weight of cargo in compartment 1
Xn - X co-ordinate of a point
Yn - Y co-ordinate of a point
Let X7 (according to the logic of flowchart in the previous Figure) be the value of co-ordinate X in pixels for LI. In order to determine the actual numerical value of LI, we have to multiply it by the number of pixels for the index unit value. The formula would be:

\[ LI = X7 + A1 \]

The numerical value for LI is increased by the value of index change DI due to loading the fuel into the tanks. The program can be developed so that it can read DI from the database whose elements are the values of the total and trip fuel for every destination (TIF and TOF). In order to obtain X co-ordinate of the line of the constant index value in pixels, DI needs to be divided by the number of pixels for the unit index value and X7 (co-ordinate of the line LI for AZFW) added to this value. Finally, the lines of constant weight are drawn in the graph (Figure 20).

\[ y_i = a[i] \times x_i - b[i] \]

The X co-ordinate represents the loaded index, and y co-ordinate the aircraft weights. The flowchart for this step in the programme is presented in Figure 21 a and b.

The input data for the realisation of this step are:

- elements of fields a[i] and b[i] defining the straight lines
- co-ordinates of the intersection points of loaded index and the actual landing weight (LI, AT/OW)
- co-ordinates of the intersection points of the loaded index zero fuel and the actual zero fuel loaded weight

Determining the %MAC at TOW starts by defining the area containing the point. The area is defined by two bordering straight lines. First, the concrete value for LI is inserted in the straight line equations, starting from the equation of the straight line which represents the upper limit. If this value exceeds the amount of AT/OW for the straight lines of positive direction coefficient (a>0), it means that we can determine the integer %MAC. In our example the line index is increased by 8 since the line with index 0 represents the straight line 8 (%MAC at TOW is 8%). Similarly, for the value AT/OW smaller than the calculated amount for the straight lines of negative direction coefficient (a<0), it means that we can determine the integer %MAC.

For more accurate determining %MAC, we have to determine the decimal amount added to the integer (remainder), so that it can be written:

\[ \%MAC = i + 8 + \text{remainder} \]

It is known now that the point is between the straight lines i and i-1. The remainder can be calculated if we determine:
- the distance between the intersection points of the straight lines between which there is the point, and the straight line representing AT/OW
  \[ (x \text{ [i-1]} - x\text{[i]}) \]
- distance between the intersection point of the straight line i and the straight line that represents AT/OW and point (LI, AT/OW)
  \[ (LI-x\text{[i]}) \]

The decimal amount is the ratio of these two distances:

\[ \text{remainder} = (LI-x\text{[i]})/(x \text{ [i-1]} - x\text{[i]}) \]
The values $x[i]$ and $x[i-1]$ are obtained from the appropriate equations of straight lines (1) if we insert AT/OW for $y_i$, obtaining the following equations:

(4) \[ x[i] = (AT/OW - b[i])a[i] \]

(5) \[ x[i-1] = (AT/OW - b[i-1])a[i-1] \]

If equations (4) and (5) are inserted in the equation (3), the final expression for $\%MAC$ at TOW is obtained:

(6) \[ \%MAC = i + 8 + ((LI * a[i] - AT/OW - b[i]) * a[i-1]) / (AT/OW - b[i-1])a[i-1] \]

The second part of the flowchart presents the same procedure but for the point of the loaded aircraft, with zero fuel wing tanks. If the point is outside the allowed limits (which in the programme equals the counter value less than zero), the message appears which recommends re-distribution of cargo. This would end the balancing procedure.

3 Conclusion

Because of the operating conditions under which the handling services perform aircraft balancing, studious attitude towards work is not possible. The services responsible for balancing the aircraft are required to perform the procedure in the shortest of time with sufficient precision, which is neither the best method nor as nearly optimal as could be obtained by the application of information software. They are also required to enter the last
minute changes which additionally reduces the safety and reliability of data and results.
Introducing the computer-aided balancing is the most sophisticated approach which satisfies the needs for speed and precision. For the computer to be able to balance the aircraft, there has to be certain software available. Today, some off-the-shelf software tools are already available, some of which are being used by the Croatian airports.
This paper emphasises the development of an independent flowchart model for proper software. Further evolution involves writing of a program code based on the proposed flowchart. The programming language C is suggested as well as the Open Windows graphical interface to support the interactive applications based on the windows-supported graphics. The ready-made functions for the graphics have been written in the programming language C as well as the tools necessary for developing applications under Open Windows. Such user friendly tools is the XView (X Windows-System-based Visually Integrated Environment for Workstation) which allows the use of graphical applications performed in windows. XView provides the user with a simple, efficient and reliable communication.
The suggested flowchart is the basis for developing the concrete application so that a possibility is left to select the operational system, graphic interface and the programming language itself.

REFERENCES
International Air Transport Association: (1966) Airport Handling Manual. 16th Edition Effective,
CLARK R., MCCORMACK. American Flyers, Donald D. Harrington – CEO. Private pilot manual
DONALD J.CLAUSING. (1988), Moderne Flug-Navigation, Motorbuch Verlag Stuttgart