

SYSTEM IDENTIFICATION OF AIRCRAFT LONGITUDINAL MOTION UTILIZING FLIGHT SIMULATOR DATA

Konstantin Metodiev

*Space Research and Technology Institute – Bulgarian Academy of Sciences
e-mail: komet@space.bas.bg*

Keywords: *FlightGear, System Identification, JSBSim, Pilatus PC-9M*

Abstract

In the presented article, an exemplary problem of identifying the parameters of an aircraft's motion within the vertical plane of symmetry is solved. The data was collected by FlightGear simulator for a Pilatus PC-9M aircraft flight case. After submitting a sequence of commands to pitch elevator and engine thrust, the simulator records five parameters of longitudinal motion at a certain frequency. The collected data are further used to build a state-space model of the flight through least-squares estimation. The obtained numerical results are compared with the experimental ones and are overlaid graphically. Charts depicting elevator angle to trim are also derived. A source code developed in GNU Octave was applied, and the problem was solved.

Introduction

The proposed study aims to develop a mathematical model of a linear time-invariant dynamic system by measuring input and output signals. The obtained model can be further used to examine system response to a broader range of input “stimuli.” In this line of thought, a relatively easy-to-implement identification algorithm has been applied to a flight simulator model of Pilatus PC-9M in order to study the aircraft’s longitudinal motion subject to control inputs of elevator angle and throttle percentage. A straightforward choice of aircraft properties to look into includes but is not limited to dynamic stability. Another important property is stick-fixed stability, which yields the elevator angular deflection to trim the aircraft at certain velocities and mass center locations.

The insufficiency of both analytical and experimental data with regard to PC-9M flying and maneuvering characteristics is somewhat significant and understandable. Presumably, one reason might be the military objectives the aircraft is meant to fulfill. The paper goes a short way towards filling that gap provided the model borrowed from FlightGear flight simulator, [1] is plausible enough for the obtained results to be credible and true.

System identification

The objective of system identification is to build up a mathematical model of a linear time-invariant system on the basis of observations. The model could be derived in both time and frequency domains and might be further used to look into the system response to different inputs. In the current study, preference is given to time domain analysis for simplicity reasons. In order to keep away from underlying complex theory, the employed system identification algorithm is solely described.

Consider the following state-space model of aircraft motion:

$$\dot{x} = Ax + Bu$$

$$(1) \quad \int_{t_0}^{t_1} \dot{x} dt = x(t_1) - x(t_0) = A \int_{t_0}^{t_1} x dt + B \int_{t_0}^{t_1} u dt$$

State variables x are gathered during the numerical experiment from time $t_0 = 0$ to t_1 . Control inputs u have been assigned in advance. In eq. (1), matrices A and B are unknown. Sufficient measurements are available for all system states for the augmented matrix $\|A \ B\|$ to be estimated through the least squares approach as demonstrated in the thesis [2], for instance.

$$Y = \bar{A} \cdot X = \|A \ B\| \cdot X$$

$$(2) \quad Y = \|x(\Delta t) \ x(2\Delta t) \ \dots \ x(n\Delta t)\| \quad X = \left\| \begin{array}{cccc} \int_0^{\Delta t} x dt & \int_0^{2\Delta t} x dt & \dots & \int_0^{n\Delta t} x dt \\ \int_0^{\Delta t} u dt & \int_0^{2\Delta t} u dt & \dots & \int_0^{n\Delta t} u dt \end{array} \right\|$$

$$\bar{A} = YX^T (XX^T)^{-1}$$

In Fig. 1, a multistep control input to the system is depicted. Pulses with alternating widths, according to rule 3-2-1-1, are passed to the elevator. The throttle receives two-sided pulses with equal widths. The sampling frequency is 20 Hz and the total integration time is 20 sec. The weather conditions are set to the simplest possible. The aircraft is brought to a steady horizontal flight using the FlightGear native autopilot. Only then can the measurement begin. The control inputs are stored in a text file and passed to the simulator utilizing Yet Another Terminal, [3], a serial monitoring utility. A delay of 1000 ms is added to each line. The connection between the simulator and the serial monitor utility is established by UDP protocol.

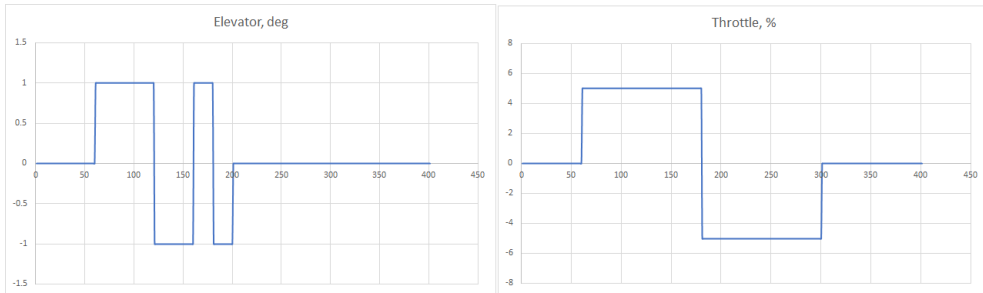


Fig. 1. Control inputs: elevator (left) and throttle

Elevator angle to trim

The PC-9M model has been produced using the Aeromatic++ utility, which comes with every JSBSim distribution [4]. A *.param file containing some basic data is required in advance for the utility to generate project files. The outcome is elaborate *.xml files containing data about airplane and engine performance, mass and balance, flight dynamics, control systems, etc. For example, the lift coefficient data shown in Fig. 2 might also be found in the main configuration PC-9M.xml file (<function name="aero/force/Lift_alpha">). In Fig. 2, a small fraction of the lift coefficient chart is shown according to angle of attack variation, deg. The lift coefficient data have been suggested by Holfman and Culp [5] and implemented in the configuration PC-9M.xml file in the JSBSim/aircraft directory. The curve slope was determined to be $5.218235 \text{ rad}^{-1}$. Savov and Marinov quote similar experimental results in paper [6] as follows: $5.01 \text{ rad}^{-1} @ M = 0.4$ and $6.2 \text{ rad}^{-1} @ M = 0.6$, which agrees well with the aforementioned data.

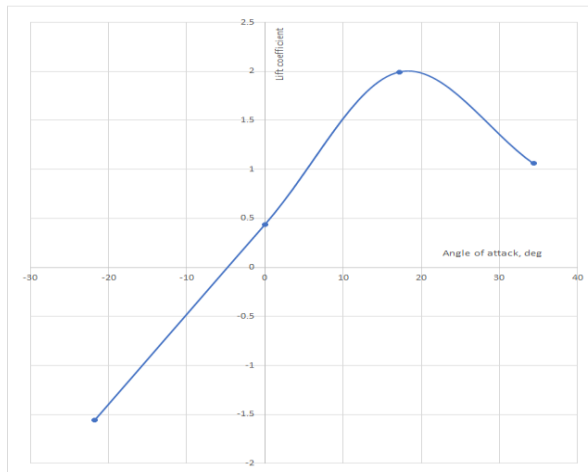


Fig. 2. Lift coefficient vs. angle of attack (deg), PC-9M.xml model

An input *.xml script is passed to the JSBSim executable as a command line option. Among other important lines, the script contains the so-called events for the program to execute. An exemplary event to point out is basic aircraft trimming carried out after setting the property “simulation / do_simple_trim” to a specified step value, exponential approach, or ramp. For example, consider the following line `<set name="simulation/do_simple_trim" value="1"/>`. JSBSim works out the elevator angle to trim the aircraft (stick fixed) at a given altitude and airspeed. Exemplary scripts are available in the JSBSim installation directory. A good tutorial on the matter still exists and might be found in the link [7]. An extensive description of xml schema used by JSBSim is available in the link [8].

It is somewhat important to mention that JSBSim succeeded in solving the assigned task for velocity values of 180 kts or less.

FlightGear photo realism

A project called FlightGear PhotoScenery, uploaded in link [9], allows the user to overlay existing scenery textures with realistic satellite orthophotos whenever available at the aircraft current longitude and latitude. A Python script downloads the requested scenery tile from specific ArcGIS servers [10]. In the FlightGear splash screen, tiles location on the drive must be specified in the relevant text box (Add-ons/Additional scenery folders). Photo realistic ability is invoked in the flight simulator environment by the View/Rendering Options/Satellite Photoscenery dialog. In Fig. 3, PC-9M flying over LB41 (ICAO ID) airstrip is shown in order to make a comparison between realistic and default textures. In the left half, credible environmental features, such as the SuperMoto circuit “Dolna Mitropolia – Pleven,” are clearly distinguishable.



Fig. 3. PC-9M flying over LB41. The left half depicts a photo realistic scenery.

Results

In Fig. 4, results obtained after system identification completion are shown. The identified data are computed using a script developed in the GNU Octave environment [11]. The state vector is $\|V, \alpha, q, \theta, H\|$, i.e., total velocity, angle of attack, pitch rate, pitch angle, and altitude. The control input is $\|\delta e, \delta th\|$, i.e., elevator and throttle, according to Fig. 1. The script used to work out the numerical solution might be found in Appendix 1 alongside an extract of the input data file, Appendix 2. Initial conditions could be read in the first line of the file containing gathered data. An extract of the input file (raw data) can be seen in Appendix 2.

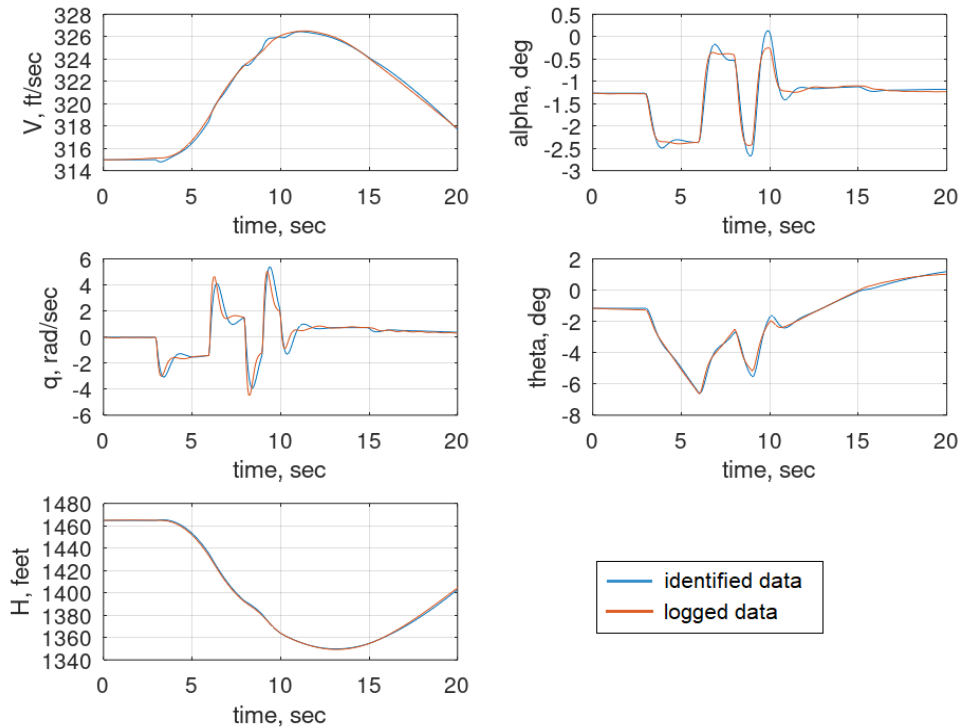


Fig. 4. Recorded and identified data

The number of samples acquired is ~ 400 , i.e., the sampling frequency is 20 Hz and total integration time is 20 sec, as it was already mentioned. It yields an integration interval long enough for the identified data to match the collected ones with sufficient precision.

In Fig. 5, a screenshot taken from the GNU Octave console window shows elements of the augmented matrix $\overline{A} = \|A \ B\|$ according to eq. (2).

```

>> sys
sys.a =
      x1      x2      x3      x4      x5
x1 -0.06688 -9.227 -29.86 -35.75 -0.005529
x2 9.589e-05 -1.588 1.021 0.02475 8.999e-05
x3 -0.004134 -11.76 -2.246 0.1361 -0.0006117
x4 0.001439 -0.008806 1.139 0.04919 0.0001737
x5 0.2025 -289.2 32.65 329 0.01836

sys.b =
      u1      u2
x1 -99.88 1.465
x2 -0.08787 0.005515
x3 -13.95 -0.5265
x4 0.2332 0.01632
x5 124.3 9.932

sys.c =
      x1 x2 x3 x4 x5
y1 1 0 0 0 0

sys.d =
      u1 u2
y1 0 0

Continuous-time model.
>> |

```

Fig. 5. Results reported by GNU Octave

In Fig. 6, results reported by JSBSim after doing a simple trim are shown. Curves are drawn for various values of flight speed and center of gravity location.

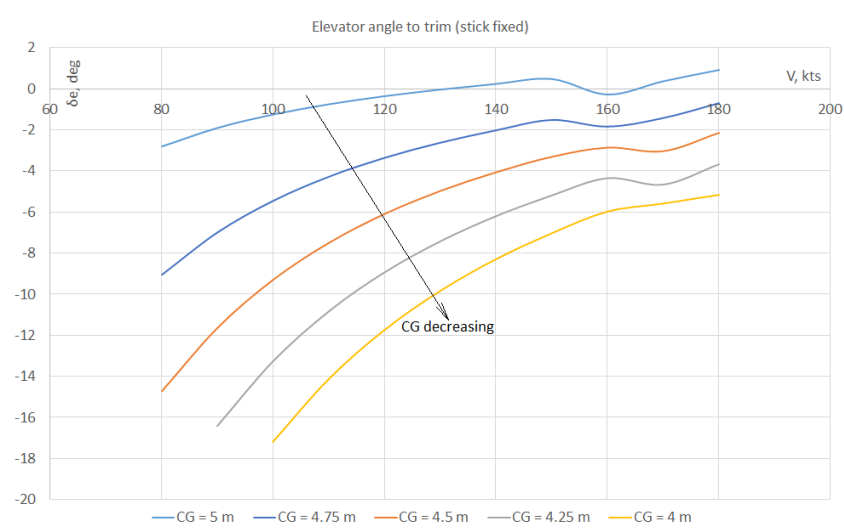


Fig. 6. Elevator angle to trim (stick fixed)

Conclusion

The identified model precision might be improved further by tuning both the integration step and interval (data gathering time). It takes numerous experiments to guess the right settings, though. The “trial and error” approach is a last resort for no evident rule exists with regard to what the best choice of settings might be. Additional experiments are to be carried out with derived model to verify its feasibility including test flights onboard the real machine.

References

1. <https://www.flightgear.org/>
2. Cetin, E., System Identification and Control of a Fixed Wing Aircraft by Using Flight Data Obtained From X-Plane Flight Simulator, M.Sc. thesis, Middle East Technical University, 2018
3. <https://sourceforge.net/projects/y-a-terminal/>
4. <https://github.com/JSBSim-Team/jsbsim/releases>
5. <http://mirrors.ibiblio.org/flightgear/ftp/Aircraft/PC-9M.zip>
6. Savov, V., A. Marinov, Aerodynamic Characteristics Calculation of Pilatus-9M Aircraft during Longitudinal and Lateral Motion, Proceedings of Scientific Conference “40 Years since a Man Stepped on the Moon and 30 Years since the First Bulgarian Cosmonaut Flight,” p.p. 66-71, vol. 1, Air Force Academy, Dolna Mitropolia, 2009, in Bulgarian
7. <http://www.holycows.net/JSBSim/index.htm>
8. <http://jsbsim.sf.net/JSBSimScript.xsd>
9. <https://github.com/nathanielwarner/flightgear-photoscenery>
10. https://services.arcgisonline.com/arcgis/rest/services/World_Imagery/MapServer/
11. <https://octave.org/>

Appendix 1. Source code in GNU Octave for system identification

```
% >> pkg load control
clear;
t = 0:.05:20.;
[r, c] = size(t);
xfile = dlmread('in.csv', ',', 1, 0); % CSV file with data
x = zeros(7,c); % Fill in data
for i=1:7
    x(i,:) = xfile(:,i);
end
% Elevator and throttle inputs
x(6,:) = x(6, :)*pi/180.; x(7,:) = x(7, :)/100.;
x0 = zeros(1, 5); x0 = x(:,1); % Vector of initial conditions
for i=1:5 % V (ft/s), alpha (rad), q (rad/s), theta (rad), H asl (ft)
    x(i,:) = x(i, :) - x(i,1);
end
```

```

X = zeros(7,c);
% Trapezoidal numerical integration
for j = 1:7
    for i = 1:c
        X(j,i) = trapz(t(1:i),x(j,1:i));
    end
end

Y = zeros(5,c); Y = x(1:5,:);

a = Y*X'*inv(X*X');

A = zeros(5,5); A(:, :) = a(:, [1:5]);
B = zeros(5,2); B(:, :) = a(:, [6:7]);
C = [1., 0., 0., 0., 0.]; D = [0., 0.];

sys = ss(A,B,C,D);
[y,t,z] = lsim(sys,x([6,7],:)',t);

% Revert initial conditions
for i=1:5
    if i == 2 || i == 3 || i == 4
        z(:,i) = z(:,i)*180./pi;
        x(i,:) = x(i,)*180./pi;
    end
    z(:,i) = z(:,i)+x0(i);
    x(i,:) = x(i,)+x0(i);
end

```

Appendix 2. A few lines extracted from the input file

	A	B	C	D	E	F	G
1	V_{Total} (ft/s)	Alpha (rad)	Q (rad/s)	Theta (rad)	Altitude ASL (ft)	Elevator, deg	Throttle, %
2	314.978696	-0.022108261	-0.000324947	-0.020274211	1464.838392	0	0
3	314.9776457	-0.022108982	-0.000369167	-0.020290665	1464.852321	0	0
4	314.9766597	-0.022111421	-0.000404946	-0.020309048	1464.866002	0	0
5	314.9758003	-0.022115363	-0.000444505	-0.020329195	1464.879432	0	0
6	314.9751434	-0.022121083	-0.000495619	-0.020351454	1464.892604	0	0
7	314.9747671	-0.022129122	-0.000562202	-0.020376484	1464.905511	0	0
8	314.9747416	-0.022140088	-0.000644654	-0.020405068	1464.918141	0	0
9	314.9750814	-0.022154479	-0.000739169	-0.020437945	1464.93048	0	0
10	314.9757056	-0.022172445	-0.000837242	-0.020475539	1464.942509	0	0

ИДЕНТИФИКАЦИЯ НА ПАРАМЕТРИТЕ НА НАДЛЪЖНОТО ДВИЖЕНИЕ НА САМОЛЕТ ПО ДАННИ ОТ ФЛАЙТ СИМУЛАТОР

К. Методиев

Резюме

В настоящата статия е решена примерна задача за идентификация на параметрите на движение на самолет във вертикалната равнина на симетрия. Данните са събрани от симулатор FlightGear за случай на полет на самолет Pilatus PC-9M. След подаване на последователност от команди към кормилото за височина и тягата на двигателя, симулаторът регистрира с определена честота пет параметъра на надлъжното движение. Събраните данни се използват за построяване на модел на полета в пространство на състоянията чрез оценка по метода на най-малките квадрати. Получените числени резултати са сверени с експерименталните и са показани в графичен вид. Балансировъчните криви на кормилото за височина са допълнително определени. Приложен е сорс-код на GNU Octave, с помощта на който е решена поставената задача.