

Generating Flight Operations Quality Assurance (FOQA) Data from the X-Plane Simulation

Anvardh Nandur (M.Sc.)

Center for Air Transportation Systems Research (CATSR)
at George Mason University (GMU)
Fairfax, VA, USA
hkourdal@masonlive.gmu.edu

Lance S.herry (Ph.D)

Center for Air Transportation Systems Research (CATSR)
at George Mason University (GMU)
Fairfax, VA, USA
lsherry@gmu.edu

Abstract— Analysis of Flight Operations Quality Assurance (FOQA) data from millions of flights from at least 10 U.S. airlines has enabled researchers, using data mining and machine learning algorithms, to identify abnormal scenarios that maybe precursors to accidents and incidents. Innovation of these algorithms is hindered by restricted access to the FOQA data to comply with confidentiality, proprietary and security policies. This paper describes a method to generate larges sets of FOQA-like data for development and testing of machine learning and data mining algorithms. This data is not restricted and can be made publicly available. The data is generated using a C++ plug-in to the X-Plane Simulation. The plug-in manipulates the simulation set-up configuration, the pilot commands, and coordinates a Monte Carlo shell to run multiple runs. An example for generating data for CAT III ILS approach for KSFO Runway 28L and 28R is described. Implications of this capability, limitations and future work are discussed.

Keywords—Data mining; Anomaly detection,; Big Data Analytics; Flight Operations Quality Assurance (FOQA), Flight Data Recorder (FDR)

I. INTRODUCTION (HEADING 1)

Flight Operations Quality Assurance (FOQA) data, also known as Flight Data Recorder (FDR) data, is detailed recorded data from the aircraft avionics bus. Typical FOQA parameters consist of both continuous and discrete (categorical) data from the avionics, propulsion system, control surfaces, landing gear, the cockpit switch positions, and other critical systems. These sets can have up to 500 parameters and are sampled at 1 Hz.

This data is used to identify deviations from normal operations. The aim is to pick out potential problems and correct them before they lead to accidents. For example, cluster analysis identified specific mountainous airports that experienced excessive Terrain Awareness and Warning System (TAWS) alerts [1], and anomaly detection algorithms have identified speed, altitude and airspace violations, as well as mode oscillations [2][3][4].

The Distributed National FOQA (Flight Operations Quality Assurance) Archive (DNFA) contains data from flight data recorders of over two million flights that have been submitted by more than 10 participating carriers. All the data is de-

identified and protected by data confidentiality, proprietary data clauses and security policies.

Due to the data confidentiality, proprietary and security policies, this data is not publicly available to all researchers developing innovative machine learning or data mining algorithms. These researchers can submit their algorithms to the DNFA be run on the data if their research meets specific criteria (e.g. relevance, contribution, does not violate confidentiality clauses) and if resources are available. The results of the algorithms are passed back to the researchers. The problem is having large enough data sets to develop and test the machine learning/data mining algorithms to the point where they are tested and ready to request access or be used on the DNFA data.

This paper describes a method to generate larges sets of FOQA-like data for development and testing of machine learning/data mining algorithms. This data is not restricted and can be made publicly available. The data is generated using the X-Plane Simulation [5] using the X-plane Software Development Kit (SDKs). External plugins were developed to manipulate the simulation set-up configuration, the pilot commands, and a Monte Carlo shell. The procedure described in this paper for the ADG plugin is capable of generating FOQA-like data for ILS approaches into KSFO airport with Auto Land configuration (AFDS mode LAND3). These algorithms can be easily expanded to include other phases of flight and other anomalous conditions. The X-Plane simulation was able to generate 485 flights in approximately 72 hours, running on a T7500 machine with Intel® Xenon® CPU, with 6 GB memory on a windows 64 bit Operating System.

This paper is organized as follows. Section 2 describes the simulation set-up for data collection. Section 3 describes the X-Plane Plug-in procedure for pilot actions during the approach. Section 4 describes the data collected for the approach procedure. Section 5 concludes with a discussion of the limitations of the method and future work.

2 AUTOMATIC DATA GENERATION (ADG) FROM X-PLANE

X-Plane is a comprehensive flight simulator for personal computers that offers realistic visuals of the flight-deck automation and out-the-window, as well as accurate

automation behavior and realistic flying qualities. X-Plane scenery package cover the Earth from 74° north to 60° south latitude and includes 33,000 airports. X-Plane includes weather modeling, simulated failure conditions, and instructor controlled simulator configuration.

One of the features of X-Plane is an open architecture API that allows modification of aircraft and aircraft system data using a plug-in.

The X-Plane Plugin SDK allows third party programmers to create plugins for X-Plane. A plug-in is an executable code that runs inside X-Plane, extending what X-Plane does. Plugins are modular, which allows the developers to extend the simulator without touching the source code of the simulator. Plugins are programs written in C, C++ or another binary-compatible language. The plug-in described in this paper uses the X-Plane 10.3x and the latest SDK (2.1.x) and was built on Windows 7 (64-bit) using MS Visual Studio 2013 Professional.

The ADG plug-in for X-Plane (Figure 1) includes four high level procedures: (1) InitDataRefs, (2) Allocate Resources, (3) Initialize Aircraft and (4) Register Flight Loop Callback.

The InitDataRefs procedure registers all the X-Plane data. The optional AllocateResources procedure reserves memory for the data objects and files as needed.

The Initialize Aircraft procedure sets the conditions for the start of the new run. These conditions include aircraft state (e.g. Latitude, Longitude, Altitude, ...) as well as aircraft system parameters (e.g. Target Altitude, Flap Position, Autopilot ON, ...).

The Register Flight Loop Callback records the “FOQA” data and also executes actions that would normally be performed by a pilot such as selecting automation modes and targets, and setting aircraft control surfaces (e.g. flaps, landing gear).

Algorithm 1 Automatic Data Generation Algorithm
(for Approach)

```

1: procedure PluginStart
2:   InitDataRefs()           P Reading Data from
Xplane and for modifying XPlane Model
3:   Allocate Resources
4:   InitializeAircraft
5:   RegisterFlightLoopCallback
(pluginRuntimeCallback, 2.0, NULL)
6: end procedure
7: procedure InitializeAircraft
8:   (Lat, Long) ← Rand({(Lati, Longi)})
9:   CurrentAltitude, TargetAltitude ← 1800 ft
10:  CurrentSpeed, TargetSpeed ← 190 kts
11:  Heading ← 210°
12:  Flaps ← 5°
13:  tear, Throttle ← 0

```

```

14:  AP 1, AP 2, AT ← 1 //Buttons set to ON
state
15:  FD1, FD2 ← 0 // Switches set to ON state
16:  V NAV ← 0 // P Button set to OFF state
17:  FlightCount ← FlightCount + 1 // P
Increment the count once initialized
18: endprocedure
19: procedure pluginRunTimeCallback
20: Record Parameters           P Record
performance data from X plane Data Refs
21: while (FMAPitchMode != FLARE) do
22: if LocalizerArmed = False AND
ApproachArmed = False then
23:  LocalizerButton ← 1 P Button set to ON
state
24:  LocalizerArmed = True
25: endif
26: if RollMode = LOC then
27:  ApproachButton ← 1 // P Button set to ON
state
28: endif
29: if PitchMode = ttS AND Flaps < 15° then
30:  TargetSpeed ← 170 kts
31:  Command (FlapsDown) // P extend from
5° to 15°
32: endif
33: if tear = 0 AND RollMode = LOC AND
PitchMode = ttS
34: AND Flaps > 15° then
35:  tear ← 1 // Extend landing gear
36:  TargetSpeed ← 150 kts
37: endif
38: if AFDSMode = AP then
39:  TargetSpeed ← 138kts
40: endif
41: end while
42: InitializeAircraft // P Reinitialize aircraft if it
reaches FLARE mode
43: end procedure
44: procedure pluginStop
45:   UnregisterFlightLoopCallback(pluginRunTimeCallback, NULL)
46: Release Resources
47: end procedure

```

Fig. 1. ADG Plug-in Algorithm

Anomalies in the flight data can be introduced in two ways: (1) programmed random events and (2) manual over-ride. The ADG plug-in can be programmed to introduce mode changes or control commands (e.g. thrust) with rare probabilities when a set of conditions exist or when a sequence of events has occurred. For example, Nanduri [4] programmed the sequence of events in the Asiana Air 214 accident.

Anomalies can also be introduced by manual over-ride of the ADGPlugin. Manual interventions include performing abnormal actions like making aircraft pitch up and slow down by pulling the control column, toggling a switch at an inappropriate time during the flight, increasing the thrust or decreasing the thrust abnormally for short durations.

II. GENERATED “FOQA” DATA

Though for adgPlugin hundreds of parameters are accessed and programmed, for assessing the performance of each flight we have recorded following 21 important parameters once every 2 seconds. Of these continuous parameters are Latitude, Longitude, Airspeed, Vertical Speed, Flap Position, Gear, Pitch, Altitude, Thrust, N1, LD, Pitch Target, Roll Target. The discrete parameters are AT, VNAV, Heading Mode, Pitch Mode, AP1, AP2, FD1, FD2. Time is considered as the reference during all our analysis. The details of these parameters is given in Table 1.

TABLE I. FOQA DATA6

Parameter (units)	Description
Continuous parameters	
Latitude (deg)	Current latitude position of aircraft
Longitude (deg)	Current longitude position of aircraft
Airspeed (knots)	Current speed
Vertical Speed (ft/min)	Rate at which aircraft is descending or ascending
Flap Position (deg)	Position of the flaps control surfaces on the aircraft wings
Gear (real)	Position of landing gear. 0 if fully retracted (up), 1 if completely extended (down) and anything in between during transition
Pitch (deg)	Current pitch of the aircraft
Altitude (feet)	The height of aircraft above sea level as given by altimeter Thrust (pounds) Forward force provided by aircraft

	engines
N1 (real)	The rotational speed of the low pressure spool also called fan speed of an engine given as percentage of rated speed
LD(real)	Lift to Drag ratio
Pitch Target	Target Pitch (along Lateral axis of aircraft)
Roll Target	Target Roll (along Longitudinal axis of aircraft) calculated
<i>Discrete Parameters</i>	
AT, AP1, AP2	Status of Auto Throttle, Auto pilot Buttons in cockpit. 1 for ON, 0 for OFF
VNAV	Status of VNAV button in cockpit. 1 for ON, 0 for OFF Heading Mode FMA heading mode used by auto pilot
Pitch Mode	FMA pitch mode used by auto pilot
FD1, FD2	Positions of Flight Director switches in cockpit. 0 for ON, 1 for OFF

III. FOQA DATA FOR B777-200 ER APPROACH KSFO RUNWAY 24L/R

FOQA data was generated for a Boeing 777-200 ER aircraft in XPlane for 500 approaches into San Francisco (KSFO) airport. A C++ plugin called adgPlugin (Approach Data Generator Plugin) has been developed which automates the approach phase of the flight into any desired airport. The plug-in enables collection of all the data generated in X-Plane. This example demonstrates the collection of 20 aircraft state and aircraft system/automation state parameters for the approach phase of flight. The demonstration included 500 flights: 485 normal flights and 15 anomalous flights.

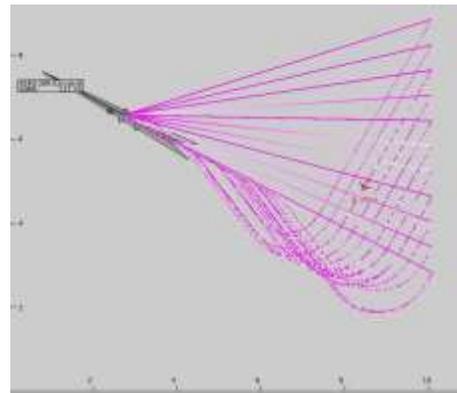


Fig. 2. Lateral Flight Paths for multiple approaches into KSFO Runway 28R ILS, CAT III approach (ILS Frequency 111.7 Hz) by adgPlugin on Boeing 777-200 ER

The ADG plug-in includes four sources of stochastic behavior:

1. Initial point of Lateral Flight Path: From a predefined set of latitude longitude pairs the plugin chooses a pair for each new approach and initializes the aircraft at that position at a fixed altitude as shown in Figure 2. Because of this some of the flights show variations in their lateral paths while approaching the specified runway.

2. Fuel and weight: We begin the simulation with fueled aircraft and during each approach the fuel is burned gradually. This version of the ADG Plugin does not refuel after each flight. This makes the aircraft lighter and lighter during and after each approach. Because of this inevitable variations in the data.

3. Wind and turbulence: The simulation enables to specify predefined wind speed or random amounts of wind speed during the flight. The simulation was configured with a random amount of winds (between 0-8 kts) for some of the flights. Because of this change in wind patterns there are variations in the data.

4. KSFO has 4 different runways available and our approaches include only two of them, either 28L or 28R. Approximately half of the flights approached runway 28L and while remaining approached runway 28R.

Figure 3 shows how various continuous and discrete parameters vary with respect to altitude and time. Each unit on horizontal axis represents 2 seconds of time. Secondary vertical axis shows altitude in feet. It can be observed that at the beginning of the approach though the aircraft is programmed to hold and maintain altitude of 1800 ft, it climbs to 2000 ft followed by a rapid descent to 1600 ft and then reaches 1800 ft to maintain that altitude. This behavior of the simulation was attributed to the pitch up condition during flare at the end of previous flight. Because of the positive pitch at end of the flight, the simulation continues to maintain the attitude even if repositioned at initial approach point. Thus it climbs to 2000 ft before its descent because of lack of thrust. Since autothrottle and autopilot are engaged, the aircraft is brought back to designated target altitude of 1800 ft. Since this behavior was commonly observed in all flights, we have assumed it as normal behavior for the purpose of this study.

Figure 4 depicts how the altitude and target airspeed vary in a typical flight as the aircraft approaches the runway. Primary vertical axis has altitude in feet and secondary vertical axis shows speed in knots.

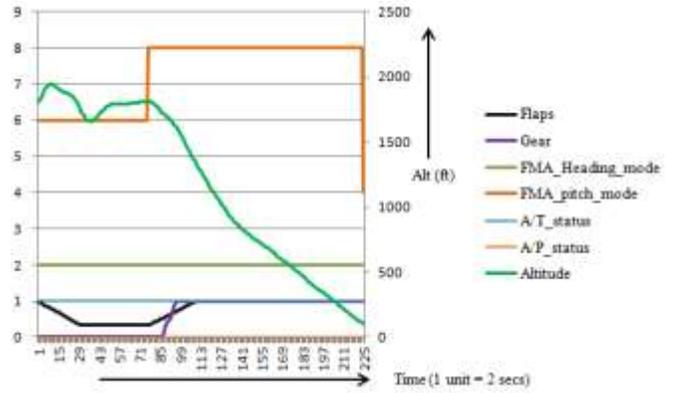


Fig. 3. Characteristics of Normal Flight Parameters

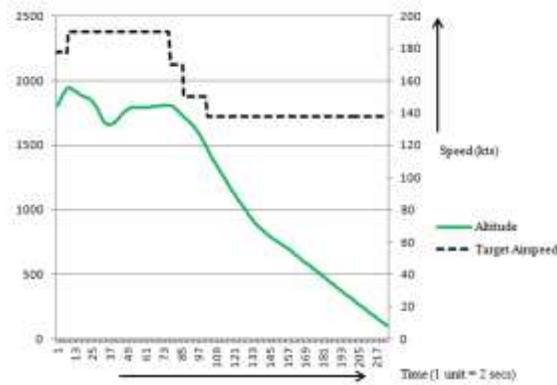


Fig. 4. Characteristics of Normal Actual Altitude and Target Airspeed

A. Anomalies in the Flight Data

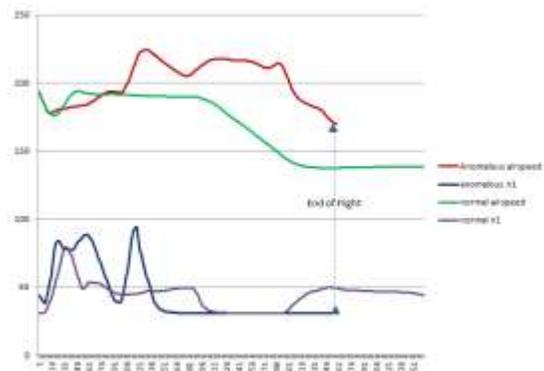
Li et al. [4] identified 11 flight anomalies in the approach and landing flight phases.

1) Very High Airspeed Approach (Rushed and Unstable Approach)

This is a very high speed ILS approach which is a type of unstable approach as shown in Figure 5. Because of high energy state of the aircraft, the engines were always idle which resulted in significantly low n1 values (anomalous n1) throughout the later part of the approach. Also because of high speed, the approach took relatively less time than normal approaches.

Fig. 5. Very High Airspeed Approach (Rushed and Unstable Approach)

2) Landing Runway Configuration Change



These types of anomalies have been observed in FOQA data by previous algorithms because of change in destination runway during final approach. We have considered two cases of this type, wherein the first case as detailed in Figure 6, the landing runway is changed from 28R to 28L after the aircraft crosses the ILS outer marker. We rely on deviations in latitude-longitude and target roll parameters to detect the runway configuration changes.

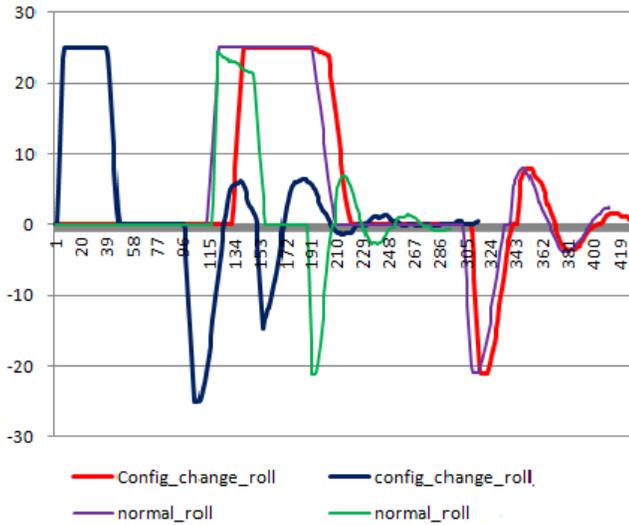


Fig. 6. Landing Runway Configuration Change: Case 1

As shown in Figure 7, in this case the landing runway is changed from 28R to 28L similar to case 1, but when aircraft is very close to the destination. It has to be noted that, both these anomalies are very subtle and are not considered severe by the Exceedance detection algorithm[3].

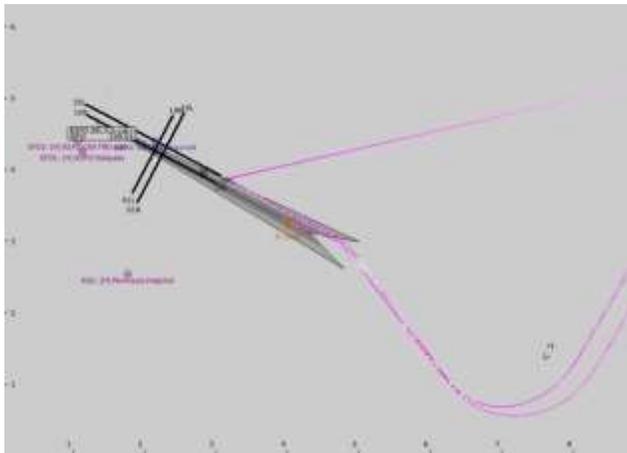


Fig. 7. Landing Runway Configuration Change: Case 2

3) *Auto Land without Full Flaps (Unusual Auto Land Configuration)*

Due to poor visibility and low ceiling altitude, visual landing may not be possible and automation is delegated to perform the landing of aircraft. There are strict requirements on both ground and airborne instruments for executing autoland operation. Generally this operation is performed with fully extended flaps and both auto pilots engaged. All the normal approaches executed by adgPlugin have full flaps configured with both autopilots engaged, during the auto landing mode. This anomalous case has flaps set to a configuration where flaps are not fully extended when the aircraft is in auto land mode as shown in Figure 8. The AFDS mode LAND3 is the autoland mode for all the flights considered in this study.

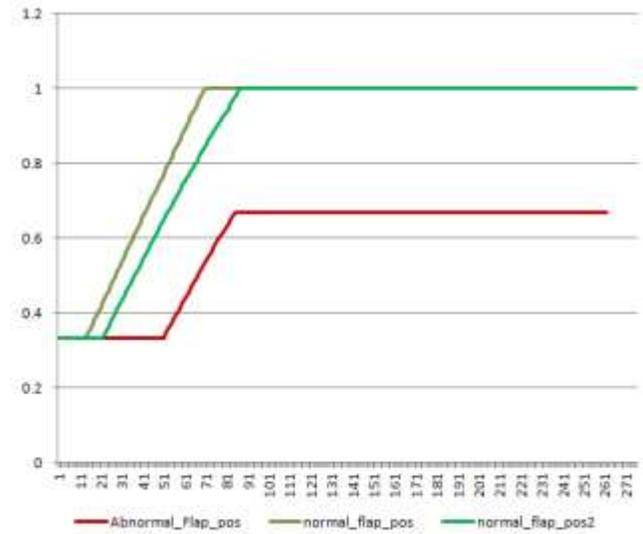


Fig. 8. Auto Land without Full Flaps (Unusual Auto Land Configuration)

4) *Auto Land with Single Auto Pilot (Unusual Auto Land Configuration)*

In this abnormal case, while flaps are set to full during autoland, through out the flight, only single AP is engaged which constitutes an unusual auto land configuration.

5) *High Energy Approach (Too High or Too Fast or both)*

This is an example of high energy flight because the airspeed was too high as shown in Figure 9. Once glideslope is captured in order to achieve rapid deceleration to target speed, the pitch was increased momentarily (as shown by anomalous pitch). This also resulted in abnormal vertical speeds (not shown). Thus the anomaly was result of multiple continuous parameters in this case.

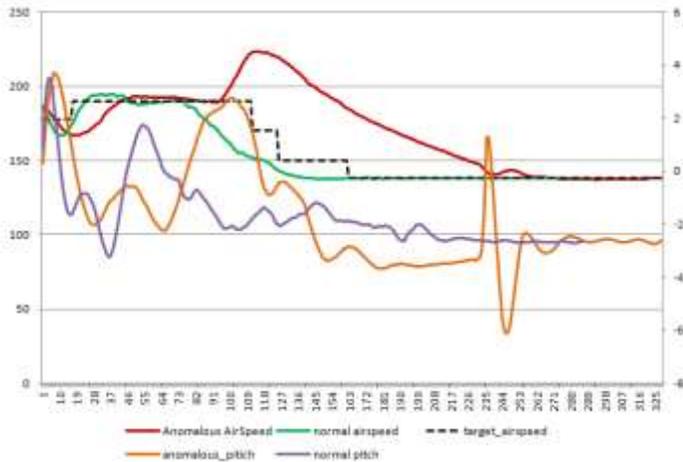


Fig. 9. High Energy Approach (Too High or Too Fast or both)

6) Recycling FDIR

As shown in Figure 10, the flight director switches are toggled (switched off and switched on) momentarily. Though this should ideally result in disconnect of auto pilots the simulation did not disconnect the automation. Nevertheless, the momentary toggle of FDIR switches is recorded in the discrete parameter data.

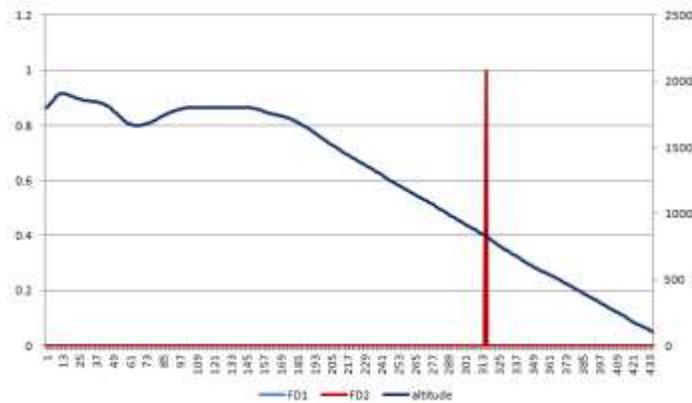


Fig. 10. Recycling FDIR

7) Influence of Wind

As shown in Figure 11, there is a significant turbulence throughout this flight. The rapid fluctuations in the continuous parameter (airspeed) is recorded in the data for this anomalous flight. Though most of the flights are subjected to wind and turbulence this case is abnormal as the influence of wind is significantly higher.

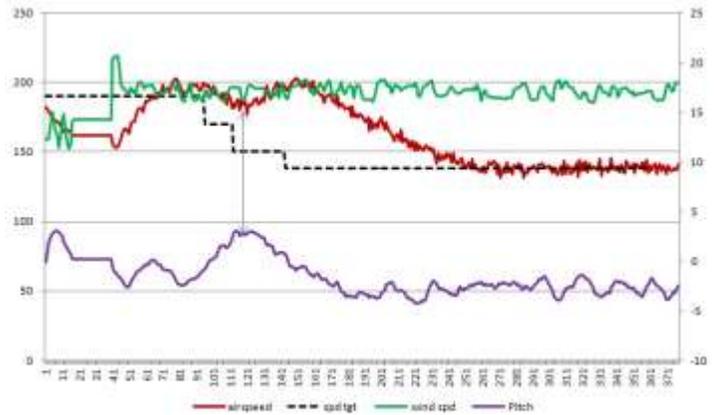


Fig. 11. Influence of Wind

8) High Pitch Rate for Short Duration

This flight is anomalous because of slight abnormalities in pitch just before landing as shown in Figure 12. Since the pitch is abnormal for only short durations, this anomaly is hard to detect.

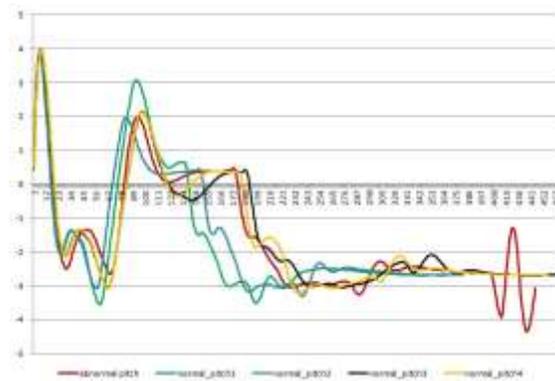


Fig. 12. Characteristics of High Pitch Rate During Landing

9) High Airspeed for Short Duration

This anomaly is related to high airspeed for very short duration. As shown in Figure 13, the airspeed was high for two short periods. The increase in airspeed was the result of anomalous pitch angle as shown in the Figure, but these are immediately rectified by appropriate actions. Since the deviations occur for short durations, these kinds of anomalies are difficult to be detected by the algorithms.

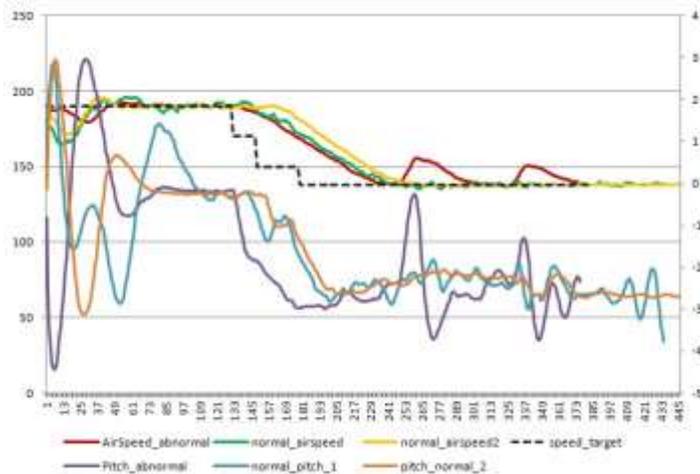


Fig. 13. High-Airspeed for Short Durations

10) *GS missed from Below, Captured from Above with VS mode*

In this case, the aircraft missed to capture the 3° glide slope path from below which is the case for normal flights in this study. The FMA pitch mode had to be changed to Vertical Speed in order for the aircraft to descend and capture the glideslope path from above. Once it is captured the Pitch mode automatically changes to GS from VS. This anomaly records the abnormalities in discrete parameter (FMA pitch mode) and also in a continuous parameter (Vertical Speed) as shown in Figure 14. Though the test set does not include this case and we do not present results for this anomaly, the proposed algorithms were able to detect this anomaly.

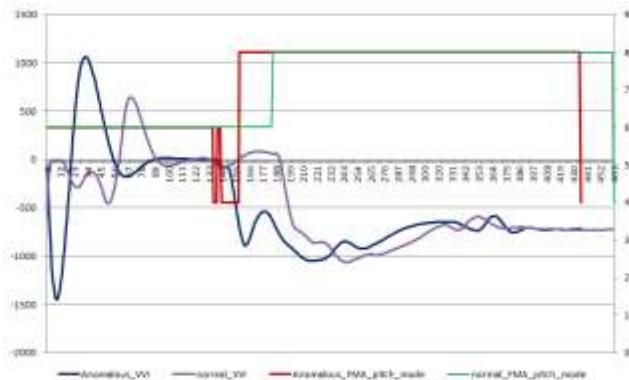


Fig. 14. G/S missed from Below, captured from Above with VS

11) *Low Energy Approach*

This anomalous flight is the another case of unstable approach but because of low energy state. As seen in Figure 15, the airspeed during the end of the approach is way less than the normal flights. Low energy unstable approaches are one of the major contributors of Controlled Flight into Stall/ Controlled Flight into Terrain accidents observed in the past.

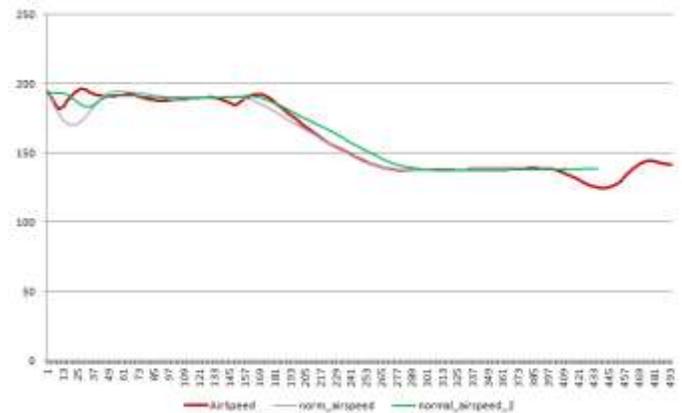


Fig. 15. Characteristics of Low Energy Approach

IV. CONCLUSIONS AND FUTURE WORK

This paper describes a plug-in to the X-Plane simulation that can be used to generate FOQA-like data that can be used for development and testing of data mining and machine learning algorithms. When these algorithms are proven, they can be used for analysis of restricted access data in the Distributed National FOQA (Flight Operations Quality Assurance) Archive (DNFA).

The X-Plane simulation was able to generate 485 flights in approximately 72 hours, running on a T7500 machine with Intel® Xenon® CPU, with 6 GB memory on a windows 64 bit Operating System.

The procedure described in this paper for the ADG plugin is capable of generating FOQA-like data for ILS approaches into KSFO airport with Auto Land configuration (AFDS mode LAND3). These algorithms can be easily expanded to include other phases of flight and other anomalous conditions.

Due to restrictions in the amount of control over the underlying aerodynamic model for Boeing 777-200 ER aircraft model, there are some limitations in the capabilities. For example, the positive pitch angle of the aircraft after reinitializing into air results in an altitude error before it get automatically corrected. Due to the unavailability of data refs for FMS buttons, needed to rely on auto/manual tuning of ILS frequency of the destination runway for the aircraft to follow the designated approach. Once data refs for Boeing-777 FMS become available in future, automating more complex approaches like manual/visual approaches would be straightforward.

A. Future Work

Next steps include the design of an intuitive Graphical User Interface for the users of the ADG plug-in to design complex vertical profiles to simulate various Standard Operating Procedures. For example, users can specify gear and flap extension points, altitude and speed constraints, speed brakes and so on. Also, the GUI could provide a menu to choose the airports and runways for the plugin to record data for various

approaches. Also a user-interface for selecting data to be recorded and the desired update rate.

ACKNOWLEDGMENT

The authors acknowledge the technical contributions from Michael Feary (NASA), Rohit Deshmukh (SJSU), Robert Mauro (Decision Research Inc.), Julia Trippe (University of Oregon), Immanuel Barshi (NASA), Houda Kourdali (CATSR/GMU), Zhenming Wang (CATSR/GMU). This research was funded by internal research funds through the George Mason University Research Foundation.

REFERENCES

- [1] ASIAs (2012) Overview of Aviation Safety InfoShare. EOFDM, January 2012.
- [2] Das, S., B. Matthews, and R. Lawrence (2011) "Fleet level anomaly detection of aviation safety data," in *Prognostics and Health Management (PHM), 2011 IEEE Conference on*, June 2011, pp. 1–10.
- [3] Das, S., B. L. Matthews, A. N. Srivastava, and N. C. Oza. (2010) Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In *Proceedings of SIGKDD'10*, pages 47–56, 2010.

- [4] Li, L., M. Gariel, R. Hansman, and R. Palacios (2011) "Anomaly detection in onboard-recorded flight data using cluster analysis," in *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*, Oct 2011, pp. 4A4–1–4A4–11.
- [5] Laminar Research (2015) X-Plane developed by Laminar Research <http://www.xplane.com>

EMAIL ADDRESSES

snanduri@masonlive.gmu.edu

lsherry@gmu.edu

2016 Integrated Communications Navigation and Surveillance (ICNS) Conference

April 19 - 21, 2016