NATIONAL TRANSPORTATION SAFETY BOARD
Vehicle Recorders Division
Washington, D.C. 20594

May 14, 2003

Flight Recorder System

Special Study
by Joseph A. Gregor

A. ACCIDENT

Location:	Bell Harbor, New York
Date:	November 12, 2001
Time:	0916 Eastern Standard Time (EST)
Aircraft:	A300-600, N14053
NTSB Number:	DCA02MA001

B. GROUP - No Group

C. SUMMARY

On November 12, 2001, about 0916 Eastern Standard Time, American Airlines flight 587, an Airbus A300-600, was destroyed when it crashed into a residential area of Belle Harbor, New York, shortly after takeoff from the John F. Kennedy International Airport (JFK), Jamaica, New York. Two pilots, 7 flight attendants, 251 passengers, and 5 persons on the ground were fatally injured. Visual meteorological conditions prevailed and an instrument flight rules (IFR) flight plan had been filed for the flight destined for Santo Domingo, Dominican Republic. The scheduled passenger flight was conducted under 14 Code of Federal Regulations (CFR) Part 121.

A ground test of the system data analog converter (SDAC), digital flight data acquisition unit (DFDAU), flight data recorder (FDR) system was run on an Airbus A300 aircraft at the Airbus factory in Toulouse, France for approximately three weeks starting on 2 September, 2002. The primary purpose for this test was to collect aircraft cockpit control and flight surface control data in support of the human factors portion of the investigation into Flight 587. A secondary goal, and the subject of this study, was to obtain experimental measurements of the data-flow delays through the SDAC, DFDAU, and FDR. The flight test instrumentation group at Patuxent River Naval Air Station (NAWC) was contracted to instrument the A300, collect the required data, and provide the data in reduced form (parameter value vs. time) for analysis by the NTSB Research and Engineering division.

D. DETAILS OF INVESTIGATION

Experimental Setup

A block diagram outlining the system architecture and data flow in the A300 used for this ground test is shown in figure 1. A block diagram outlining the system architecture of the Pax River Flight Test Recorder (FTR) is shown in figure 2. In the cockpit, the control wheel, control column, and rudder pedals were equipped with analog sensors which were fed to analog inputs of the DFDAU. This data was output from the DFDAU via an ARINC 573 bus to the Pax River FTR, which read the digital data directly by synchronizing on the data-frame sync word. Additional sensors mounted on the aircraft external control surfaces – ailerons, rudder, and elevator - were fed to the SDAC. This data was output via an ARINC 429 bus to the digital input to the DFDAU. The 429 bus output was simultaneously sampled by the Pax River FTR. The 429 bus was updated by the SDAC at a rate of 8 samples/second. The Pax River FTU sampled this bus at a rate of 8.54 samples/second to mitigate data loss. Each analog sensor was simultaneously fed to dedicated digital to analog converter (DAC) within the Pax River FTR, and the data recorded as a separate parameter. A series of flight control tests involving manual manipulation of the flight controls in the cockpit (see appendix A), and observation of the response from the aircraft control surfaces, was made and the results recorded digitally in engineering-units for further analysis.

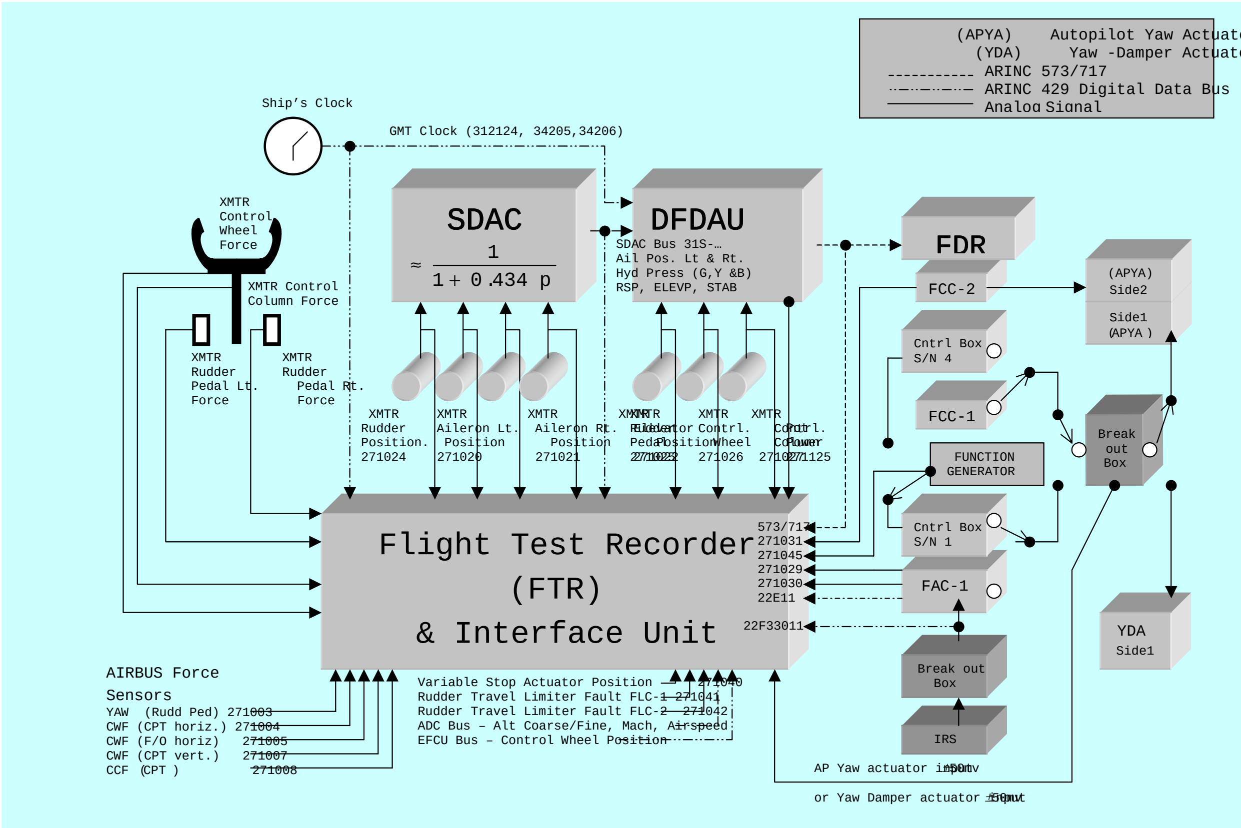

Figure 1. Block diagram of A300 ground test architecture.

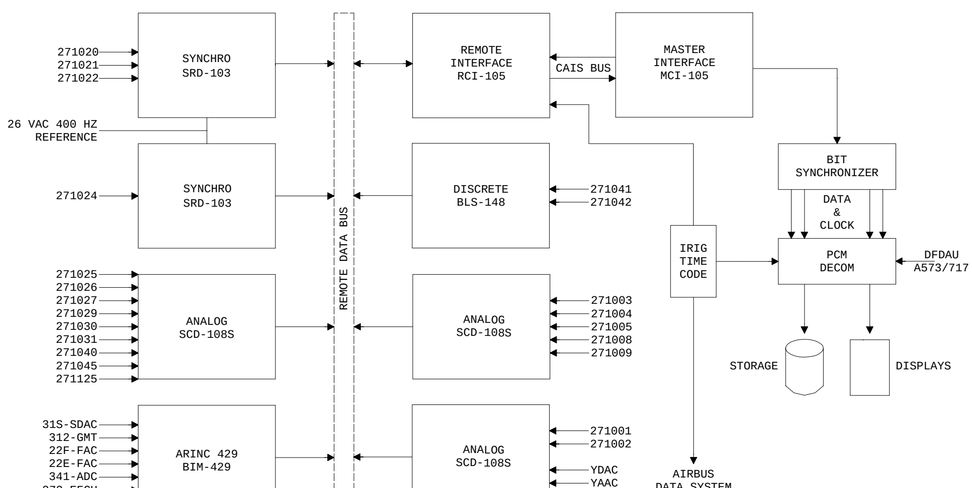

Figure 2. Block diagram of Pax River Flight Test Recorder.

Data Structure

Data from the ground tests was received at the NTSB laboratory in the form of comma delimited variable (.csv) spreadsheet files containing time-stamped parameter values expressed in engineering units. The data from each aircraft sensor was acquired simultaneously via two different data-flow paths:

- a supplemental direct path from the sensor to the Pax River FTR, and
- the normal aircraft systems path from sensor through the SDAC and/or DFDAU, to the FTR.

The time stamps within these files represent the time the respective parameter was written to the memory storage device within the FTR (hard disk). Parameters from the ARINC 573 bus were synchronized with the embedded block sync-word and corrected for offset within the data frame. The time stamps for these parameters represent the time the data became available at the output of the DFDAU, plus a small (known) delay due to the processing overhead for the bit synchronizer and PCM demodulator within the FTR. Data from the ARINC 429 bus was grabbed asynchronously by sampling the bus at 8.54 samples/second - slightly faster than the bus nominal update rate of 8 times/second. The time stamps for these parameters represent the time the data was actually grabbed by the Pax River FTR plus the (known) delay required for analog to digital conversion, internal bus transmission, interface, bit synchronization and PCM demodulation. An additional delay exists in this data flow path, corresponding to the time-difference between the moment when the data became available on the output of the SDAC, and the moment when the FTR actually sampled the 429 bus. This delay can be accounted for statistically in the final overall delay calculation.

Analysis Procedure

The time delay due to processing within the SDAC and DFDAU may be estimated by correlating each pair of parameters representing the output from the same analog sensor; one passing through the SDAC and/or DFDAU, and the other going directly to the FTR. The waveforms representing each pair should be essentially identical in profile, but offset in time; the time shift or delay corresponding to the difference in time required to traverse the two separate data flow paths. The parameters provided by the Pax River FTR are summarized in table 1.

PARAMETER	**SAMPLE RATE** (Sa/s)[1]	**COMMENTS**
AILPOLHSDAC	8.54	Left aileron position from SDAC
AILPORHSDAC	8.54	Right aileron position from SDAC
AILPOS_LH_573	1	Left aileron position from DFDAU
AILPOS_RH_573	1	Right aileron position from DFDAU
AILPOSLHANLG	68.35	Left aileron position from sensor
AILPOSRHANLG	68.35	Right aileron position from sensor
CCP	68.35	Control column position from sensor
CCP_573	2	Control column position from DFDAU
ELEVP_573	2	Elevator position from DFDAU
ELEVPOSANLG	68.35	Elevator position from sensor
ELEVPOSSDAC	8.54	Elevator position from SDAC
GMT_HOURS	8.54	GMT from ship's clock
GMT_MIN	8.54	GMT from ship's clock
GMT_SEC	8.54	GMT from ship's clock
RDRPOSANLG	68.35	Rudder position from sensor
RPP	68.35	Rudder pedal position from sensor
RPP_573	2	Rudder pedal position from DFDAU
RSP_573	2	Rudder position from DFDAU
RSPSDAC	8.54	Rudder position from SDAC

Table 1. Parameter names and sample rates used for the parameters recorded by the Pax River FTR.

Data corresponding to rudder position, aileron position (left), elevator position, and rudder pedal position was extracted and correlated using MATLAB[2] to obtain the total delay between the two data flow paths for each parameter. The various components of the total delay through these data flow paths can represented by a mathematical equation. Using the experimental measurements for differential delay, these equations may then be solved for the unknown delay through the SDAC and/or DFDAU.

The delay through the data flow path from analog sensor directly to the ADC within the Pax River FTR is represented by the following measured parameters in the Pax River naming convention: AILPOSLHANLG, ELEVPOSANLG, RDRPOSANLG, RPP, CCP (e.g. aileron position, elevator position, ruder position, rudder pedal position, and

[1] Samples/second
[2] Numerical analysis and programming environment licensed by The MathWorks.

control column position, respectively). For these parameters, the difference between the time the data appears at the sensor and the time stamp appended by the Pax River FTR is given by:

$$\Delta_{A} = \tau_{ADC} + \tau_{\Delta} \qquad \text{Eq. [1]}$$

where τ_{ADC} is the time required for the analog to digital conversion process within the FTR; and τ_{Δ} is a delay representing the time between output from the analog to digital converter (ADC) internal to the FTR and generation of the recorded time stamp.

The delay through the data flow path from analog sensor to the ARINC 429 input of the Pax River FTR applies to the following measured parameters: AILPOSLHSDAC, ELEVPOSSDAC, RSPSDAC (i.e. left aileron position, elevator position, and rudder surface position, respectively). For these parameters, the difference between the time the data appears at the sensor, and the time stamp appended by the FTR for 429 bus input, is given by:

$$\Delta_{429} = \tau_{SDAC} + \tau_{429} + \tau_{\Delta} + \tau_{\Gamma} \qquad \text{Eq. [2]}$$

where τ_{SDAC} is the time delay through the SDAC; τ_{429} is the time required for the Pax River FTR to read and encode the ARINC 429 bus; τ_{Δ} is a delay representing the time between output from the FTR's internal ADC and generation of the recorded time stamp; and τ_{Γ} represents the delay between the time when the data first became available at the output of the SDAC, and the time the Pax River FTR grabbed a snapshot of the data from the 429 bus.

The delay through the data flow path from analog sensor through the SDAC and/or DFDAU, and to the ARINC 573 input of the Pax River FTR applies to the following measured parameters: AILPOS_LH_573, ELEVP_573, RPP_573, RSP_573 (i.e. aileron position, elevator position, rudder pedal position, and rudder surface position, respectively). For these parameters, the difference between the time the data appears at the sensor and the time stamp appended by the FTR for 573 bus input is given by:

$$\Delta_{573} = \tau_{SDAC} + \tau_{DFDAU} + \tau_{BIT\_PCM} + \tau_{\lambda} \qquad \text{Eq. [3]}$$

where τ_{SDAC} is the time delay through the SDAC; τ_{DFDAU} is the time delay through the DFDAU for digital inputs; τ_{BIT_PCM} is the time delay introduced by the bit comparator and PCM demodulator within the Pax River FTR; and τ_{λ} represents the delay between the time the data first became available at the output of the SDAC, and the time the DFDAU grabs a snapshot of the data from the 429 bus.

For rudder pedal position, the difference between the time the data appears at the sensor and the time stamp appended by the FTR for 573 bus input is given by:

$$\Delta'_{573} = \tau'_{DFDAU} + \tau_{BIT\_PCM} \qquad \text{Eq. [4]}$$

where τ'_{DFDAU} is the time delay through the DFDAU for analog inputs; and τ_{BIT_PCM} is the time delay introduced by the bit comparator and PCM demodulator within the Pax River FTR.

A mathematical cross-correlation operation was performed between seven pairs of parameters. The correlation procedure was repeated using all six data sets provided by the instrumentation group at Pax River and the results summarized in table 2. Entries for correlations which failed due to corrupted data appear in strike-through text.

Correlation [ms]	Parameter	TEST-1.01	TEST-2.01	TEST-3.01	TEST-4.01	TEST-5.01	TEST-6.01
$\Delta_{429} - \Delta_A$	Rudder Pos.	424.5	424.5	395.2	365.9	351.3	351.3
$\Delta_{573} - \Delta_A$	Rudder Pos.	512.3	468.4	453.7	468.4	409.8	424.5
$\Delta_{429} - \Delta_A$	Aileron Pos.	439.1	424.5	365.9	365.9	365.9	365.9
$\Delta_{573} - \Delta_A$	Aileron Pos.	497.7	439.1	~~4961.9~~	512.3	~~5035.1~~	380.6
$\Delta_{429} - \Delta_A$	Elevator Pos.	439.1	~~87.8~~	395.2	~~131.7~~	365.9	365.9
$\Delta_{573} - \Delta_A$	Elevator Pos.	512.3	~~-541.6~~	~~1258.8~~	~~190.3~~	424.5	424.5
$\Delta'_{573} - \Delta_A$	Rudder Pedal Pos.	14.6	29.3	0	29.3	0	-14.6

Table 2. Parameter names and sample rates used for the parameters recorded by the Pax River FTR.

Using equations [1] – [4], together with the cross correlations of measured data summarized in table 2, a closed-form solution was found for the delay through the SDAC and DFDAU:

$$\boldsymbol{\tau}_{\mathbf{SDAC}} = \left(\Delta_{\mathbf{429}} - \Delta_{\mathbf{A}}\right) - \left\langle \boldsymbol{\tau}_{\Gamma} \right\rangle \qquad \text{Eq. [5]}$$

Analog input:

$$\boldsymbol{\tau}'_{\mathbf{DFDAU}} = \left(\Delta'_{\mathbf{573}} - \Delta_{\mathbf{A}}\right) + \left\langle \boldsymbol{\tau}_{\Delta} \right\rangle \qquad \text{Eq. [6]}$$

Digital input:

$$\boldsymbol{\tau}_{\mathbf{DFDAU}} = \left(\Delta_{\mathbf{573}} - \Delta_{\mathbf{A}}\right) - \left\langle \boldsymbol{\tau}_{\mathbf{SDAC}} \right\rangle + \left\langle \boldsymbol{\tau}_{\boldsymbol{\lambda}} \right\rangle - \left\langle \boldsymbol{\tau}_{\Delta} \right\rangle \qquad \text{Eq. [7]}$$

where $\left\langle \boldsymbol{\tau}_{\Gamma} \right\rangle$ is the expected value of the delay between the time the data first becomes available at the output to the SDAC, and the time the Pax River FTR grabs the data from the 429 bus; $\left\langle \boldsymbol{\tau}_{\boldsymbol{\lambda}} \right\rangle$ is the expected value of the delay between the time the data first becomes available at the output to the SDAC, and the time the DFDAU grabs the data from the 429 bus; $\left\langle \boldsymbol{\tau}_{\Delta} \right\rangle$ is the expected value of the internal FTR delay representing the time between output from the internal FTR ADC and generation of the recorded time stamp; and $\left\langle \boldsymbol{\tau}_{\mathbf{SDAC}} \right\rangle$ is the expected value of the delay through the SDAC itself.

The delay due to the asynchronous nature of the communication between the SDAC, the DFDAU, and the Pax River FTR can be modeled as a continuous random variable. If no relationship exists between the time the SDAC outputs the data, and the time the FTR and DFDAU acquire the data, then the incremental delay could be any value from 0 to $^{1}/_{(\text{sample rate})}$ seconds, with equal probability. Making this assumption, the expected values for $\left\langle \boldsymbol{\tau}_{\Gamma} \right\rangle$ and $\left\langle \boldsymbol{\tau}_{\boldsymbol{\lambda}} \right\rangle$ are then given by

$$\left\langle \tau_{\Gamma} \right\rangle = \left\langle \tau_{\lambda} \right\rangle \cong \int_{0}^{1/8} \tau_{\Gamma} f_{\Gamma}\left(\tau_{\Gamma}\right) d\tau_{\Gamma} \cong 62.5\ \text{ms} \qquad \text{Eq. [8]}$$

where $f_\Gamma(\tau_\Gamma)$ is the probability density function (PDF), representing the probability that τ_Γ will equal a specific value at a specific time. The PDF for all asynchronous delay variables was assumed to be a constant valued function between the minimum and maximum possible delay times, and zero elsewhere. $\langle \boldsymbol{\tau}_{\mathbf{SDAC}} \rangle$ is obtained by taking the average of equation [5] calculated over all available cases. Data received separately from Pax River indicate that $\langle \boldsymbol{\tau}_{\Delta} \rangle \cong \mathbf{62.5\,ms}$.

Numerical Results

The data from table 2, together with Equations [5] – [8], was used to determine the mean delay times for data passing through the SDAC and DFDAU. The expected value for the measurement error in this experiment is given by:

$$\langle \boldsymbol{\sigma}_{\tau} \rangle \cong \int_0^{1/8} (\boldsymbol{\tau} - \boldsymbol{\mu}_{\tau})^2 \mathbf{f}_{\tau}(\boldsymbol{\tau}) \mathbf{d}\boldsymbol{\tau} \cong \mathbf{62.4\,ms} \qquad \text{Eq. [9]}$$

where $\boldsymbol{\mu}_{\tau} = \langle \boldsymbol{\tau} \rangle$ for each source of delay. Equation [9] may be used to determine the error bars for the measurement. Assuming that the individual delays are uncorrelated and combine as one net delay with gaussian statistics, the total error will be the square root of the sum of the squares of the individual errors. The mean delays through the SDAC and DFDAU are given by the expected value of the random variables calculated in Eqs. [5] – [7] above. These delays, along with the mean error for the measurement itself, are shown in table 3 below.

Parameter	**Comment**	**Mean Delay**	**Mean Error**
$\langle \boldsymbol{\tau}_{\mathbf{SDAC}} \rangle$		326 ± 25 ms	$\sigma \approx 62$ ms
$\langle \boldsymbol{\tau}_{\mathbf{DFDAU}} \rangle$	Digital Input	130 ± 39 ms	$\sqrt{3}\sigma \approx 108$ ms
$\langle \boldsymbol{\tau}'_{\mathbf{DFDAU}} \rangle$	Analog Input	72 ± 20 ms	$\sigma \approx 62$ ms

Table 3. Mean delay times through the SDAC and DFDAU as determined using Pax River ground test data.

Note that the standard deviation in the mean delay as calculated from experimental measurement is less than the mean error. This is physical, and may indicate some degree of incidental correlation between the variables due to the small data set and short timeframe over which the data was collected.

Joseph A. Gregor
Electrical Engineer

APPENDIX A

Test Conditions and Procedures

1. Simulation of Preflight Flight Control Check;
   - **a.** **Conditions** (Airspeed= 0 knots, Yaw Damper "ON", All Hydraulics "ON")
   - **b.** **Procedures –** Using captain's rudder pedals, perform preflight rudder control check pausing at full pedal deflection and null position for 5 seconds, repeat procedure for Control Wheel and Column.

2. Maximum Rate Flight Control Inputs with Pause at Full Deflection;
   - **a.** **Conditions** (Airspeed= 0 knots, Yaw Damper "ON", All Hydraulics "ON")
   - **b.** **Procedures –** Using captain's rudder pedals, move rudder pedals to stops (left and right) at maximum rate pausing at rudder pedal stops for 5 seconds, repeat procedure for Control Wheel and Column.

3. Maximum Rate Flight Control Inputs with "NO" Pause at Full Deflection;
   - **a.** **Conditions** (Airspeed= 0 knots, Yaw Damper "ON", All Hydraulics "ON")
   - **b.** **Procedures –** Using captain's rudder pedals, move rudder pedals to stops (left and right) at maximum rate, repeat procedure for Control Wheel and Column.

4. Repeat Test 2 with Airspeed at 240 knots;

5. Repeat Test 3 with Airspeed at 240 knots;

6. Maximum SDAC Activity
   - **a.** **Conditions -** (Airspeed=240 knots, Yaw Damper "ON", All Hydraulics "ON")
   - **b.** **Procedures –** Simultaneously move rudder pedals, control wheel and control column through full range of travel at maximum rate.

APPENDIX B

Log files for the MATLAB data runs employed to correlate the parameters and determine the delay times for the various data-flow paths represented in table 2.

DFDRTEST-1.01

[Rdr_SDAC_Corr Rdr_SDAC_Delay RDRPOSANLG_cs RSPSDAC_cs RDRPOSANLG_t_cs RSPSDAC_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSPSDAC', RDRPOSANLG, RSPSDAC, RDRPOSANLG_t, RSPSDAC_t);

SampRate1 = 68.35944760488357
SampRate2 = 8.53422626342275

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSPSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSPSDAC lags RDRPOSANLG by 424.5 ms

pause
[Rdr_573_Corr Rdr_573_Delay RDRPOSANLG_cs RSP_573_cs RDRPOSANLG_t_cs RSP_573_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSP.573', RDRPOSANLG, RSP_573, RDRPOSANLG_t, RSP_573_t);

SampRate1 = 68.35944760488357
SampRate2 = 2.02289551446634

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSP.573 lags RDRPOSANLG by 512.3 ms

pause
[Ail_SDAC_Corr Ail_SDAC_Delay AILPOSLHANLG_cs AILPOLHSDAC_cs AILPOSLHANLG_t_cs AILPOLHSDAC_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOLHSDAC', AILPOSLHANLG, AILPOLHSDAC, AILPOSLHANLG_t, AILPOLHSDAC_t);

SampRate1 = 68.35944760487769
SampRate2 = 8.53919465230149

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOLHSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOLHSDAC lags AILPOSLHANLG by 439.1 ms

pause

[Ail_573_Corr Ail_573_Delay AILPOSLHANLG_cs AILPOS_LH_573_cs AILPOSLHANLG_t_cs AILPOS_LH_573_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOS.LH.573', AILPOSLHANLG, AILPOS_LH_573, AILPOSLHANLG_t, AILPOS_LH_573_t);

SampRate1 = 68.35944760487769

SampRate2 = 1.01019012854166

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOS.LH.573 assumed to be 1.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOS.LH.573 lags AILPOSLHANLG by 497.7 ms

pause
[Elev_SDAC_Corr Elev_SDAC_Delay ELEVPOSANLG_cs ELEVPOSSDAC_cs ELEVPOSANLG_t_cs ELEVPOSSDAC_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVPOSSDAC', ELEVPOSANLG, ELEVPOSSDAC, ELEVPOSANLG_t, ELEVPOSSDAC_t);

SampRate1 = 68.35944760487769
SampRate2 = 8.53716509188753

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVPOSSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVPOSSDAC lags ELEVPOSANLG by 439.1 ms

pause
[Elev_573_Corr Elev_573_Delay ELEVPOSANLG_cs ELEVP_573_cs ELEVPOSANLG_t_cs ELEVP_573_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVP.573', ELEVPOSANLG, ELEVP_573, ELEVPOSANLG_t, ELEVP_573_t);

SampRate1 = 68.35944760487769

SampRate2 = 2.02043645169353

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVP.573 lags ELEVPOSANLG by 512.3 ms

pause
[RPP_573_Corr RPP_573_Delay RPP_cs RPP_573_cs RPP_t_cs RPP_573_t_cs] = CSV_Param_Corr_v2('RPP', 'RPP.573', RPP, RPP_573, RPP_t, RPP_573_t);

SampRate1 = 68.36535287851322

SampRate2 = 2.02683699590947

Sample rate for RPP assumed to be 68.32 Sa/s
Sample rate for RPP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RPP.573 lags RPP by 14.6 ms

==

DFDRTEST-2.01

[Rdr_SDAC_Corr Rdr_SDAC_Delay RDRPOSANLG_cs RSPSDAC_cs RDRPOSANLG_t_cs RSPSDAC_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSPSDAC', RDRPOSANLG, RSPSDAC, RDRPOSANLG_t, RSPSDAC_t);

SampRate1 = 68.38243364460836

SampRate2 = 8.56651999494815

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSPSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSPSDAC lags RDRPOSANLG by 424.5 ms

%pause
[Rdr_573_Corr Rdr_573_Delay RDRPOSANLG_cs RSP_573_cs RDRPOSANLG_t_cs RSP_573_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSP.573', RDRPOSANLG, RSP_573, RDRPOSANLG_t, RSP_573_t);

SampRate1 = 68.38243364460836

SampRate2 = 2.09492520203501

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSP.573 lags RDRPOSANLG by 468.4 ms

%pause
[Ail_SDAC_Corr Ail_SDAC_Delay AILPOSLHANLG_cs AILPOLHSDAC_cs AILPOSLHANLG_t_cs AILPOLHSDAC_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOLHSDAC', AILPOSLHANLG, AILPOLHSDAC, AILPOSLHANLG_t, AILPOLHSDAC_t);

SampRate1 = 68.38242679169051
SampRate2 = 8.55998698513765

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOLHSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOLHSDAC lags AILPOSLHANLG by 424.5 ms

%pause
[Ail_573_Corr Ail_573_Delay AILPOSLHANLG_cs AILPOS_LH_573_cs AILPOSLHANLG_t_cs AILPOS_LH_573_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOS.LH.573', AILPOSLHANLG, AILPOS_LH_573, AILPOSLHANLG_t, AILPOS_LH_573_t);

SampRate1 = 68.38242679169051
SampRate2 = 1.05458792344754

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOS.LH.573 assumed to be 1.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOS.LH.573 lags AILPOSLHANLG by 439.1 ms

%pause
[Elev_SDAC_Corr Elev_SDAC_Delay ELEVPOSANLG_cs ELEVPOSSDAC_cs ELEVPOSANLG_t_cs ELEVPOSSDAC_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVPOSSDAC', ELEVPOSANLG, ELEVPOSSDAC, ELEVPOSANLG_t, ELEVPOSSDAC_t);

SampRate1 = 68.38242679169051

SampRate2 = 8.46295880217941

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVPOSSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVPOSSDAC lags ELEVPOSANLG by 87.8 ms

%pause
[Elev_573_Corr Elev_573_Delay ELEVPOSANLG_cs ELEVP_573_cs ELEVPOSANLG_t_cs ELEVP_573_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVP.573', ELEVPOSANLG, ELEVP_573, ELEVPOSANLG_t, ELEVP_573_t);

SampRate1 = 68.38242679169051

SampRate2 = 2.11017374140046

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVP.573 lags ELEVPOSANLG by -541.6 ms

%pause
[RPP_573_Corr RPP_573_Delay RPP_cs RPP_573_cs RPP_t_cs RPP_573_t_cs] = CSV_Param_Corr_v2('RPP', 'RPP.573', RPP, RPP_573, RPP_t, RPP_573_t);

SampRate1 = 68.38243521643791
SampRate2 = 2.11198262319650

Sample rate for RPP assumed to be 68.32 Sa/s
Sample rate for RPP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RPP.573 lags RPP by 29.3 ms

==

DFDRTEST-3.01

[Rdr_SDAC_Corr Rdr_SDAC_Delay RDRPOSANLG_cs RSPSDAC_cs RDRPOSANLG_t_cs RSPSDAC_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSPSDAC', RDRPOSANLG, RSPSDAC, RDRPOSANLG_t, RSPSDAC_t);

SampRate1 = 68.42492757435916
SampRate2 = 8.50783294415472

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSPSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSPSDAC lags RDRPOSANLG by 395.2 ms

%pause
[Rdr_573_Corr Rdr_573_Delay RDRPOSANLG_cs RSP_573_cs RDRPOSANLG_t_cs RSP_573_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSP.573', RDRPOSANLG, RSP_573, RDRPOSANLG_t, RSP_573_t);

SampRate1 = 68.42492757435916
SampRate2 = 2.40456625580726

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSP.573 lags RDRPOSANLG by 453.7 ms

%pause
[Ail_SDAC_Corr Ail_SDAC_Delay AILPOSLHANLG_cs AILPOLHSDAC_cs AILPOSLHANLG_t_cs AILPOLHSDAC_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOLHSDAC', AILPOSLHANLG, AILPOLHSDAC, AILPOSLHANLG_t, AILPOLHSDAC_t);

SampRate1 = 68.42492757435916
SampRate2 = 8.49745721826662

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOLHSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOLHSDAC lags AILPOSLHANLG by 365.9 ms

%pause
[Ail_573_Corr Ail_573_Delay AILPOSLHANLG_cs AILPOS_LH_573_cs AILPOSLHANLG_t_cs AILPOS_LH_573_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOS.LH.573', AILPOSLHANLG, AILPOS_LH_573, AILPOSLHANLG_t, AILPOS_LH_573_t);

SampRate1 = 68.42492757435916

SampRate2 = 1.23965456612256

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOS.LH.573 assumed to be 1.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOS.LH.573 lags AILPOSLHANLG by 4961.9 ms

%pause
[Elev_SDAC_Corr Elev_SDAC_Delay ELEVPOSANLG_cs ELEVPOSSDAC_cs ELEVPOSANLG_t_cs ELEVPOSSDAC_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVPOSSDAC', ELEVPOSANLG, ELEVPOSSDAC, ELEVPOSANLG_t, ELEVPOSSDAC_t);

SampRate1 = 68.42492757435916
SampRate2 = 8.60639474839166

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVPOSSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVPOSSDAC lags ELEVPOSANLG by 395.2 ms

%pause
[Elev_573_Corr Elev_573_Delay ELEVPOSANLG_cs ELEVP_573_cs ELEVPOSANLG_t_cs ELEVP_573_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVP.573', ELEVPOSANLG, ELEVP_573, ELEVPOSANLG_t, ELEVP_573_t);

SampRate1 = 68.42492757435916
SampRate2 = 2.45814882679692

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVP.573 lags ELEVPOSANLG by 1258.8 ms

%pause
[RPP_573_Corr RPP_573_Delay RPP_cs RPP_573_cs RPP_t_cs RPP_573_t_cs] = CSV_Param_Corr_v2('RPP', 'RPP.573', RPP, RPP_573, RPP_t, RPP_573_t);

SampRate1 = 68.42492757429396
SampRate2 = 2.44371859870883

Sample rate for RPP assumed to be 68.32 Sa/s
Sample rate for RPP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RPP.573 lags RPP by 0.0 ms

===

DFDRTEST-4.01

[Rdr_SDAC_Corr Rdr_SDAC_Delay RDRPOSANLG_cs RSPSDAC_cs RDRPOSANLG_t_cs RSPSDAC_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSPSDAC', RDRPOSANLG, RSPSDAC, RDRPOSANLG_t, RSPSDAC_t);

SampRate1 = 68.38878286654831
SampRate2 = 8.53205896146981

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSPSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.
Assumed sample rate for the delay calculation is 68.320 Sa/s

RSPSDAC lags RDRPOSANLG by 365.9 ms

%pause
[Rdr_573_Corr Rdr_573_Delay RDRPOSANLG_cs RSP_573_cs RDRPOSANLG_t_cs RSP_573_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSP.573', RDRPOSANLG, RSP_573, RDRPOSANLG_t, RSP_573_t);

SampRate1 = 68.38878286654831
SampRate2 = 2.11393119203707

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSP.573 lags RDRPOSANLG by 468.4 ms

%pause
[Ail_SDAC_Corr Ail_SDAC_Delay AILPOSLHANLG_cs AILPOLHSDAC_cs AILPOSLHANLG_t_cs AILPOLHSDAC_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOLHSDAC', AILPOSLHANLG, AILPOLHSDAC, AILPOSLHANLG_t, AILPOLHSDAC_t);

SampRate1 = 68.38877139658780
SampRate2 = 8.52374773433604

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOLHSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOLHSDAC lags AILPOSLHANLG by 365.9 ms

%pause
[Ail_573_Corr Ail_573_Delay AILPOSLHANLG_cs AILPOS_LH_573_cs AILPOSLHANLG_t_cs AILPOS_LH_573_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOS.LH.573', AILPOSLHANLG, AILPOS_LH_573, AILPOSLHANLG_t, AILPOS_LH_573_t);

SampRate1 = 68.38877139658780
SampRate2 = 1.08057415412330

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOS.LH.573 assumed to be 1.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOS.LH.573 lags AILPOSLHANLG by 512.3 ms

%pause
[Elev_SDAC_Corr Elev_SDAC_Delay ELEVPOSANLG_cs ELEVPOSSDAC_cs ELEVPOSANLG_t_cs ELEVPOSSDAC_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVPOSSDAC', ELEVPOSANLG, ELEVPOSSDAC, ELEVPOSANLG_t, ELEVPOSSDAC_t);

SampRate1 = 68.38877139661699
SampRate2 = 8.57607920631918

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVPOSSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVPOSSDAC lags ELEVPOSANLG by 131.7 ms

%pause
[Elev_573_Corr Elev_573_Delay ELEVPOSANLG_cs ELEVP_573_cs ELEVPOSANLG_t_cs ELEVP_573_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVP.573', ELEVPOSANLG, ELEVP_573, ELEVPOSANLG_t, ELEVP_573_t);

SampRate1 = 68.38877139661699
SampRate2 = 2.13154367690566

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVP.573 lags ELEVPOSANLG by 190.3 ms

%pause
[RPP_573_Corr RPP_573_Delay RPP_cs RPP_573_cs RPP_t_cs RPP_573_t_cs] = CSV_Param_Corr_v2('RPP', 'RPP.573', RPP, RPP_573, RPP_t, RPP_573_t);

SampRate1 = 68.38878487300966
SampRate2 = 2.13384463137089

Sample rate for RPP assumed to be 68.32 Sa/s
Sample rate for RPP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RPP.573 lags RPP by 29.3 ms

===

DFDRTEST-5.01

[Rdr_SDAC_Corr Rdr_SDAC_Delay RDRPOSANLG_cs RSPSDAC_cs RDRPOSANLG_t_cs RSPSDAC_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSPSDAC', RDRPOSANLG, RSPSDAC, RDRPOSANLG_t, RSPSDAC_t);

SampRate1 = 68.42849439842327
SampRate2 = 8.50603053361028

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSPSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSPSDAC lags RDRPOSANLG by 351.3 ms

%pause
[Rdr_573_Corr Rdr_573_Delay RDRPOSANLG_cs RSP_573_cs RDRPOSANLG_t_cs RSP_573_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSP.573', RDRPOSANLG, RSP_573, RDRPOSANLG_t, RSP_573_t);

SampRate1 = 68.42849439842327
SampRate2 = 2.27975986739050

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSP.573 lags RDRPOSANLG by 409.8 ms

%pause
[Ail_SDAC_Corr Ail_SDAC_Delay AILPOSLHANLG_cs AILPOLHSDAC_cs AILPOSLHANLG_t_cs AILPOLHSDAC_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOLHSDAC', AILPOSLHANLG, AILPOLHSDAC, AILPOSLHANLG_t, AILPOLHSDAC_t);

SampRate1 = 68.42842262829419
SampRate2 = 8.55566289936887

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOLHSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOLHSDAC lags AILPOSLHANLG by 365.9 ms

%pause
[Ail_573_Corr Ail_573_Delay AILPOSLHANLG_cs AILPOS_LH_573_cs AILPOSLHANLG_t_cs AILPOS_LH_573_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOS.LH.573', AILPOSLHANLG, AILPOS_LH_573, AILPOSLHANLG_t, AILPOS_LH_573_t);

SampRate1 = 68.42842262829419
SampRate2 = 1.17220798687146

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOS.LH.573 assumed to be 1.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOS.LH.573 lags AILPOSLHANLG by 5035.1 ms

%pause
[Elev_SDAC_Corr Elev_SDAC_Delay ELEVPOSANLG_cs ELEVPOSSDAC_cs ELEVPOSANLG_t_cs ELEVPOSSDAC_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVPOSSDAC', ELEVPOSANLG, ELEVPOSSDAC, ELEVPOSANLG_t, ELEVPOSSDAC_t);

SampRate1 = 68.42849439842327
SampRate2 = 8.54045498961822

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVPOSSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVPOSSDAC lags ELEVPOSANLG by 365.9 ms

%pause
[Elev_573_Corr Elev_573_Delay ELEVPOSANLG_cs ELEVP_573_cs ELEVPOSANLG_t_cs ELEVP_573_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVP.573', ELEVPOSANLG, ELEVP_573, ELEVPOSANLG_t, ELEVP_573_t);

SampRate1 = 68.42849439842327
SampRate2 = 2.33126167238541

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVP.573 lags ELEVPOSANLG by 424.5 ms

%pause
[RPP_573_Corr RPP_573_Delay RPP_cs RPP_573_cs RPP_t_cs RPP_573_t_cs] = CSV_Param_Corr_v2('RPP', 'RPP.573', RPP, RPP_573, RPP_t, RPP_573_t);

SampRate1 = 68.42848967340392
SampRate2 = 2.31738895151879

Sample rate for RPP assumed to be 68.32 Sa/s
Sample rate for RPP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RPP.573 lags RPP by 0.0 ms

===

DFDRTEST-6.01

DFDRTEST_Analysis_Script
[Rdr_SDAC_Corr Rdr_SDAC_Delay RDRPOSANLG_cs RSPSDAC_cs RDRPOSANLG_t_cs RSPSDAC_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSPSDAC', RDRPOSANLG, RSPSDAC, RDRPOSANLG_t, RSPSDAC_t);

SampRate1 = 68.38878687944191
SampRate2 = 8.53205921248887

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSPSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSPSDAC lags RDRPOSANLG by 351.3 ms

pause
[Rdr_573_Corr Rdr_573_Delay RDRPOSANLG_cs RSP_573_cs RDRPOSANLG_t_cs RSP_573_t_cs] = CSV_Param_Corr_v2('RDRPOSANLG', 'RSP.573', RDRPOSANLG, RSP_573, RDRPOSANLG_t, RSP_573_t);

SampRate1 = 68.38878687944191
SampRate2 = 2.12506674037720

Sample rate for RDRPOSANLG assumed to be 68.32 Sa/s
Sample rate for RSP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RSP.573 lags RDRPOSANLG by 424.5 ms

pause
[Ail_SDAC_Corr Ail_SDAC_Delay AILPOSLHANLG_cs AILPOLHSDAC_cs AILPOSLHANLG_t_cs AILPOLHSDAC_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOLHSDAC', AILPOSLHANLG, AILPOLHSDAC, AILPOSLHANLG_t, AILPOLHSDAC_t);

SampRate1 = 68.38877540778843
SampRate2 = 8.52374798486629

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOLHSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOLHSDAC lags AILPOSLHANLG by 365.9 ms

pause
[Ail_573_Corr Ail_573_Delay AILPOSLHANLG_cs AILPOS_LH_573_cs AILPOSLHANLG_t_cs AILPOS_LH_573_t_cs] = CSV_Param_Corr_v2('AILPOSLHANLG', 'AILPOS.LH.573', AILPOSLHANLG, AILPOS_LH_573, AILPOSLHANLG_t, AILPOS_LH_573_t);

SampRate1 = 68.38877540778843
SampRate2 = 1.07182386277124

Sample rate for AILPOSLHANLG assumed to be 68.32 Sa/s
Sample rate for AILPOS.LH.573 assumed to be 1.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

AILPOS.LH.573 lags AILPOSLHANLG by 380.6 ms

pause
[Elev_SDAC_Corr Elev_SDAC_Delay ELEVPOSANLG_cs ELEVPOSSDAC_cs ELEVPOSANLG_t_cs ELEVPOSSDAC_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVPOSSDAC', ELEVPOSANLG, ELEVPOSSDAC, ELEVPOSANLG_t, ELEVPOSSDAC_t);

SampRate1 = 68.38877540775924
SampRate2 = 8.54670932307087

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVPOSSDAC assumed to be 8.54 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVPOSSDAC lags ELEVPOSANLG by 365.9 ms

pause
[Elev_573_Corr Elev_573_Delay ELEVPOSANLG_cs ELEVP_573_cs ELEVPOSANLG_t_cs ELEVP_573_t_cs] = CSV_Param_Corr_v2('ELEVPOSANLG', 'ELEVP.573', ELEVPOSANLG, ELEVP_573, ELEVPOSANLG_t, ELEVP_573_t);

SampRate1 = 68.38877540775924
SampRate2 = 2.14495153482064

Sample rate for ELEVPOSANLG assumed to be 68.32 Sa/s
Sample rate for ELEVP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

ELEVP.573 lags ELEVPOSANLG by 424.5 ms

pause
[RPP_573_Corr RPP_573_Delay RPP_cs RPP_573_cs RPP_t_cs RPP_573_t_cs] = CSV_Param_Corr_v2('RPP', 'RPP.573', RPP, RPP_573, RPP_t, RPP_573_t);

SampRate1 = 68.38878687944191
SampRate2 = 2.14731591352787

Sample rate for RPP assumed to be 68.32 Sa/s
Sample rate for RPP.573 assumed to be 2.00 Sa/s

The lower sample-rate parameter has been cubic spline interpolated for the correlation.

Assumed sample rate for the delay calculation is 68.320 Sa/s

RPP.573 lags RPP by -14.6 ms

echo off

APPENDIX C

MATLAB M-files used to reduce the raw data and perform the parameter correlation for the data collected using the Pax River flight test unit.

```
function [Param, Param_t] = CSV_Param_Extract(filename, N, T, t0, P)
% Algorithm used to extract parameter data stored in a .csv file.  This data
% pertains to A-300 ground tests performed in Toulouse, France in September
2002.
% The data is contained in a .csv file produced by technicians from Pax River
NAS;
% who performed the instrumentation and data collection for these tests.
%
% Inputs:
%  filename = name of csv from which to obtain parameter data
%         N = column number (zero based) of the parameter in the .csv file.
%         T = vector representing time [s] from midnight for all possible data
%               acquisition windows. Only a subset of these will apply to any
%               individual parameter.
%        t0 = guess at earliest valid acquisition time for parameter
%         P = plot number to use for display of results
%
% Outputs:
%       Param   = vector representing extracted parameter
%       Param_t = vector representing time [s] from midnight for extracted
parameter
%
% Before running this file, 'Time_extraction.m' must be run to obtain the
generic
% time-stamp information, as well as a header containing an ordered listing of
% the parameters included in the .csv file.  This information is saved as
MATLAB
% workspace variables.  The header is used to determine the value of N
required to
% obtain the desired parameter.  The number of chunks needed to load in the
entire
% .csv file is variable and may be determined via experimentation.
% 10/17/02  jag     modified

% Acquire data in 64kB chunks to workaround memory problem
fprintf(1,'Reading data...\n')
Data1 = csvread(filename,2,N,[2,N,8192,N]); %65535
fprintf(1,'16.7%% done...\n')
Data2 = csvread(filename,65536,N,[65536,N,131071,N]);
fprintf(1,'33.3%% done...\n')
Data3 = csvread(filename,131072,N,[131072,N,196607,N]);
fprintf(1,'50%% done...\n')
Data4 = csvread(filename,196608,N,[196608,N,262143,N]);
fprintf(1,'66.7%% done...\n')
Data5 = csvread(filename,262144,N,[262144,N,327679,N]);
fprintf(1,'83.3%% done...\n')
Data6 = csvread(filename,327680,N,[327680,N,393215,N]);
fprintf(1,'100%% done.\n\n')

% Create a single column vector from the data
Data = Data1'; %[Data1' Data2' Data3' Data4' Data5' Data6']';

clear Data1 %Data2 Data3 Data4 Data5 Data6

% Find time tag info for each non-zero data point and discard the rest
Param = Data(find(Data));
```

```
Param_t = T(find(Data));
clear Data

% Remove invalid datapoints and create workspace variables
Param = Param(find(Param_t > t0));
Param_t = Param_t(find(Param_t > t0));

% Plot the data
figure(P)
ph1 = plot(Param_t, Param);
title('Parameter from .csv file')
xlabel('time [s] from midnight')
zoom on
```

```
function    [header, h, m, s, t, t0, ts] = CSV_Time_Extract(filename)
% This m-file is used to extract time tags from a .csv file containing data
for the
% AA587 A-300 ground tests in Toulouse, France, in Fall 2002.
%
% This m-file creates the following variables for use by the
FDR_Param_Extract.m and
% FDR_Param_Corr.m routines:
%       ts  = timestamp containing text denoting time in h:m:s when a data
acquisition
%               'grab' was made.  Any individual parameter will contain
entries in the
%               .csv file only for a subset of these times.
%       h   = hours measured from midnight
%       m   = minutes measured from midnight
%       s   = seconds measured from midnight
%       t0  = guess at earliest valid datapoint
%       t   = time [s] of data point acquisition, since midnight
%
% 9/23/02   jag     created
% 10/17/02  jag     modified

% Extract time-stamp from column one (1) of the .csv file and calculate the
elapsed
% time in seconds from the morning of the test.
[A ts C] = textread(filename,'%s %15s %s',8192); %  393210
clear A C
ts = ts(3:length(ts));
for n = 1:length(ts)
    [h(n),m(n),s(n)] = strread(ts{n},'%f %f %f','delimiter',':');
end
t0 = 3600*h(1) + 60*m(1);   % guess at earliest valid datapoint
t = 3600*h + 60*m + s;      % actual elapsed time for datapoint

% Read header information into cell array (copy into excel spreadsheet to
% determine the column numbers of desired parameters
header = textread(filename,'%s',6,'delimiter',',');
```

```
function [Corr, Delay, Param1_rs, Param2_rs, Param1_t_rs, Param2_t_rs] =
CSV_Param_Corr_v2(Param1_name, Param2_name, Param1, Param2, Param1_t,
Param2_t)
% Algorithm for correlating two equal sized FDR parameter data vectors.
% 10/21/02  jag     created
```

```
% 10/24/02  jag     added code to enable cubic spline interpolation of data
for correlation
%                   of parameters with disparate sample rates
% 11/13/02  jag     modified code to enable cubic spline interpolation of data
for correlation
%                   of parameters with disparate sample rates
%
% Inputs:
%       Param1_name, Param2_name    =   names of parameters to be correlated
%       Param1, Param2              =   parameter data to be correlated
%       Param1_t, Param2_t          =   time corresponding to parameter data
%       SampRate1, SampRate2        =   sample rate for parameter data
%
% Outputs:
%       Corr                        =   correlation results
%       Delay                       =   delay [ms] between Param2 and Param1
%       Param1_rs, Param2_rs        =   resampled parameter data
%       Param1_t_rs, Param2_t_rs    =   resampled time corresponding to
parameter data


% Deduce an approx sample rate from the time vector for the parameter.
SampRate1 = length(Param1)/(Param1_t(length(Param1))-Param1_t(1))
SampRate2 = length(Param2)/(Param2_t(length(Param2))-Param1_t(2))

% Pick precise sample rate from above deduction and knowledge of actual
% sample rates used by the Pax River data acquisition system. Sample
% rates used in this experiment were: 2, 4, 4.27, 8.54, and 68.35.
if (SampRate1 < 1.5)
    SampRate1 = 1;
elseif (SampRate1 < 3)
    SampRate1 = 2;
elseif (SampRate1 < 4.135)
    SampRate1 = 4;
elseif (SampRate1 < 6.405)
    SampRate1 = 4.27;
elseif (SampRate1 < 38.43)
    SampRate1 = 8.54;
elseif (SampRate1 < 100)
    SampRate1 = 68.32;
end

if (SampRate2 < 1.5)
    SampRate2 = 1;
elseif (SampRate2 < 3)
    SampRate2 = 2;
elseif (SampRate2 < 4.135)
    SampRate2 = 4;
elseif (SampRate2 < 6.405)
    SampRate2 = 4.27;
elseif (SampRate2 < 38.43)
    SampRate2 = 8.54;
elseif (SampRate2 < 100)
    SampRate2 = 68.32;
end

fprintf(1,'\nSample rate for %s assumed to be %2.2f Sa/s\n', Param1_name,
SampRate1)
fprintf(1,'\nSample rate for %s assumed to be %2.2f Sa/s\n', Param2_name,
SampRate2)

% Determine which parameter was sampled at the lower rate and upsample
% to match sample rates for both parameters.  Upsample ratio is equal
% to P/Q. The option exists to pick standard upsampling techniques or
% cubic spline interpolation to obtain processable data.
```

```
if (SampRate1 > SampRate2)
    % *** FOR UPSAMPLING ***
        %Q = 100;
        %P = round(SampRate1*Q/SampRate2);
        %Param2 = resample(Param2,P,Q);
        %Param2_t = interp1(Param2_t,[1:(Q/P):length(Param2_t)]);
        %Param2 = Param2(1:length(Param2_t));
    % *** FOR INTERPOLATING ***
        Param2 = interp1(Param2_t,Param2,Param1_t,'cubic');
        Param2_t = Param1_t;
    SampRate = SampRate1;
elseif (SampRate1 <= SampRate2)
    % *** FOR UPSAMPLING ***
        %Q = 100;
        %P = round(SampRate2*Q/SampRate1);
        %Param1 = resample(Param1,P,Q);
        %Param1_t = interp1(Param1_t,[1:(Q/P):length(Param1_t)]);
        %Param1 = Param1(1:length(Param1_t));
    % *** FOR INTERPOLATING ***
        Param1 = interp1(Param1_t,Param1,Param2_t,'cubic');
        Param1_t = Param2_t;
    SampRate = SampRate2;
end

% *** FOR UPSAMPLED DATA ***
%fprintf(1,'\n\nThe lower sample-rate parameter has been upsampled by\n')
%fprintf(1,'a factor of %2.3f for the correlation.\n', P/Q)
%fprintf(1,'\nThe sample rate used for both Parameters is now %2.3f Sa/s\n',
SampRate)

% *** FOR INTERPOLATED DATA ***
fprintf(1,'\n\nThe lower sample-rate parameter has been cubic spline\n')
fprintf(1,'interpolated for the correlation.\n')
fprintf(1,'\nAssumed sample rate for the delay calculation is %2.3f Sa/s\n',
SampRate)

% Determine which vector is longer and adjust both to equal lengths
Param1_len = length(Param1);
Param2_len = length(Param2);
if (Param1_len > Param2_len)
     Param1 = Param1([1:length(Param2)]);
     Param1_t = Param1_t(1:length(Param1));
else
     Param2 = Param2([1:length(Param1)]);
     Param2_t = Param2_t(1:length(Param2));
end
n = length(Param1);

% Return resampled data
Param1_rs = Param1;
Param1_t_rs = Param1_t;
Param2_rs = Param2;
Param2_t_rs = Param2_t;

% Plot the raw parameter data
figure
ph = plot(Param1,'b-');
hold on
plot(Param2,'r-')
title('Data to be Correlated')
legend(Param1_name, Param2_name)
xlabel('samples')
%print
saveas(ph,[Param1_name, '_raw_', Param2_name, '.fig'])
```

```
% Perform the correlation[c_ww,lags]
[Corr, lags] = xcorr(Param1, Param2);

% Plot the correlation
figure
ph = plot(lags,Corr,'b-');
title(['Correlation between ', Param1_name, ' and ', Param2_name])
xlabel('lags')
%print
saveas(ph,[Param1_name, '_corr_', Param2_name, '.fig'])

% Find position of the correlation maxima
[C, I] = max(Corr);

% Calculate the delay in milliseconds between Param1 and Param2
Delay = 1000*((n-I)/SampRate + (Param1_t(1)-Param2_t(1)));
fprintf(1,'\n\n%s lags %s by %3.1f ms\n\n', Param2_name, Param1_name, Delay)

% Reset figures for next run
%figure(1)
```

```
% Automate Parameter extraction process from Pax River data files DFDRTEST-
1.01.csv thru DFDRTEST-6.01.csv.

[AILPOLHSDAC, AILPOLHSDAC_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 1, t,
t0, 1);
[AILPOS_LH_573, AILPOS_LH_573_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 3,
t, t0, 2);
[AILPOSLHANLG, AILPOSLHANLG_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 5, t,
t0, 3);
[CCP, CCP_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 9, t, t0, 4);
[CCP_573, CCP_573_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 10, t, t0, 5);
[ELEVP_573, ELEVP_573_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 14, t, t0,
6);
[ELEVPOSANLG, ELEVPOSANLG_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 15, t,
t0, 7);
[ELEVPOSSDAC, ELEVPOSSDAC_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 16, t,
t0, 8);
[GMT_HOURS, GMT_HOURS_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 17, t, t0,
9);
[GMT_MIN, GMT_MIN_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 18, t, t0, 10);
[GMT_SEC, GMT_SEC_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 19, t, t0, 11);
[RDRPOSANLG, RDRPOSANLG_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 21, t, t0,
12);
[RPP, RPP_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 24, t, t0, 13);
[RPP_573, RPP_573_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 25, t, t0, 14);
[RSP_573, RSP_573_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 26, t, t0, 15);
[RSPSDAC, RSPSDAC_t] = CSV_Param_Extract('DFDRTEST-6.01.csv', 27, t, t0, 16);
```

```
%Run parameter correlation for relevant pairs. Display results to console to
save as log file.

echo on
[Rdr_SDAC_Corr Rdr_SDAC_Delay RDRPOSANLG_rs RSPSDAC_rs RDRPOSANLG_t_rs
RSPSDAC_t_rs] = CSV_Param_Corr_v1('RDRPOSANLG', 'RSPSDAC', RDRPOSANLG,
RSPSDAC, RDRPOSANLG_t, RSPSDAC_t);
```

```
%pause

[Rdr_573_Corr Rdr_573_Delay RDRPOSANLG_rs RSP_573_rs RDRPOSANLG_t_rs
RSP_573_t_rs] = CSV_Param_Corr_v1('RDRPOSANLG', 'RSP.573', RDRPOSANLG,
RSP_573, RDRPOSANLG_t, RSP_573_t);

%pause

[Ail_SDAC_Corr Ail_SDAC_Delay AILPOSLHANLG_rs AILPOLHSDAC_rs AILPOSLHANLG_t_rs
AILPOLHSDAC_t_rs] = CSV_Param_Corr_v1('AILPOSLHANLG', 'AILPOLHSDAC',
AILPOSLHANLG, AILPOLHSDAC, AILPOSLHANLG_t, AILPOLHSDAC_t);

%pause

[Ail_573_Corr Ail_573_Delay AILPOSLHANLG_rs AILPOS_LH_573_rs AILPOSLHANLG_t_rs
AILPOS_LH_573_t_rs] = CSV_Param_Corr_v1('AILPOSLHANLG', 'AILPOS.LH.573',
AILPOSLHANLG, AILPOS_LH_573, AILPOSLHANLG_t, AILPOS_LH_573_t);

%pause

[Elev_SDAC_Corr Elev_SDAC_Delay ELEVPOSANLG_rs ELEVPOSSDAC_rs ELEVPOSANLG_t_rs
ELEVPOSSDAC_t_rs] = CSV_Param_Corr_v1('ELEVPOSANLG', 'ELEVPOSSDAC',
ELEVPOSANLG, ELEVPOSSDAC, ELEVPOSANLG_t, ELEVPOSSDAC_t);

%pause

[Elev_573_Corr Elev_573_Delay ELEVPOSANLG_rs ELEVP_573_rs ELEVPOSANLG_t_rs
ELEVP_573_t_rs] = CSV_Param_Corr_v1('ELEVPOSANLG', 'ELEVP.573', ELEVPOSANLG,
ELEVP_573, ELEVPOSANLG_t, ELEVP_573_t);

%pause

[RPP_573_Corr RPP_573_Delay RPP_rs RPP_573_rs RPP_t_rs RPP_573_t_rs] =
CSV_Param_Corr_v1('RPP', 'RPP.573', RPP, RPP_573, RPP_t, RPP_573_t);

echo off
```

REFERENCES

1. K. Shanmugan and A. Breipohl, *Random Signals Detection, Estimation and data Analysis*, Wiley, New York (1988)