Development and Evaluation of Onboard Model-Based Gas Path Diagnostic Methods

Jeffrey Armstrong
ASRC Aerospace Corporation

Propulsion Controls and Diagnostics Workshop
December 8, 2009

Goals

A structured, standardized data set is needed for the development and evaluation of onboard gas-path diagnostic strategies.

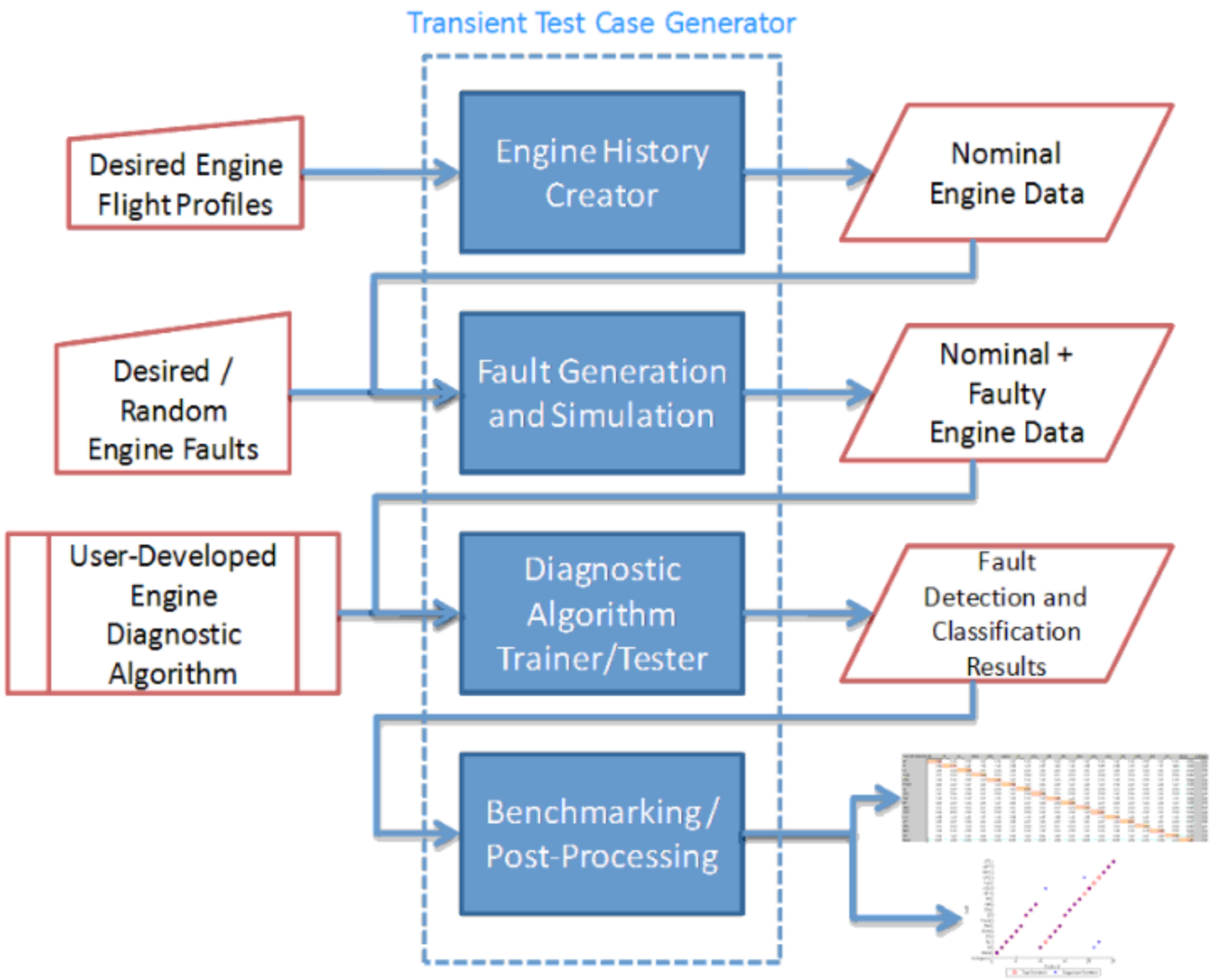
- Construct engine "histories" containing multiple realistic transient flight profiles
- Insert gas path faults into specified flight profiles
- Feed engine histories or single flight profiles into detection/classification algorithms
- Quantify algorithms' success metrics

Implementation

Transient Test Case Generator

- MATLAB ® / Simulink ® tool suite
- Based on Commercial Modular Aero Propulsion System Simulation, *C-MAPSS*
- Introduces sensor noise on 8 control sensors
- Includes engine health trending
- Incorporates three engine fault categories
- Calculates diagnostic evaluation metrics

Implementation



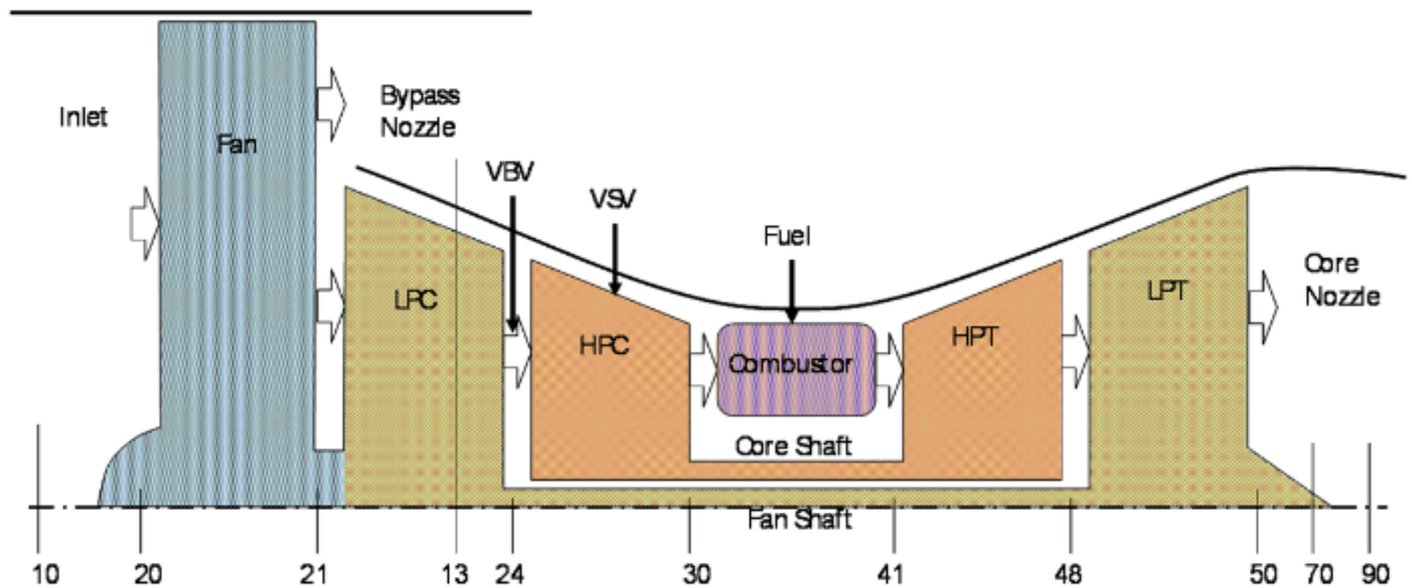
Implementation: Interface

Transient Test Case Generator is a collection of high-level MATLAB routines for generating engine data.

- Command-line driven
- Create single profiles or complete histories
- Insert faults into nominal profiles
- Evaluate diagnostic algorithms

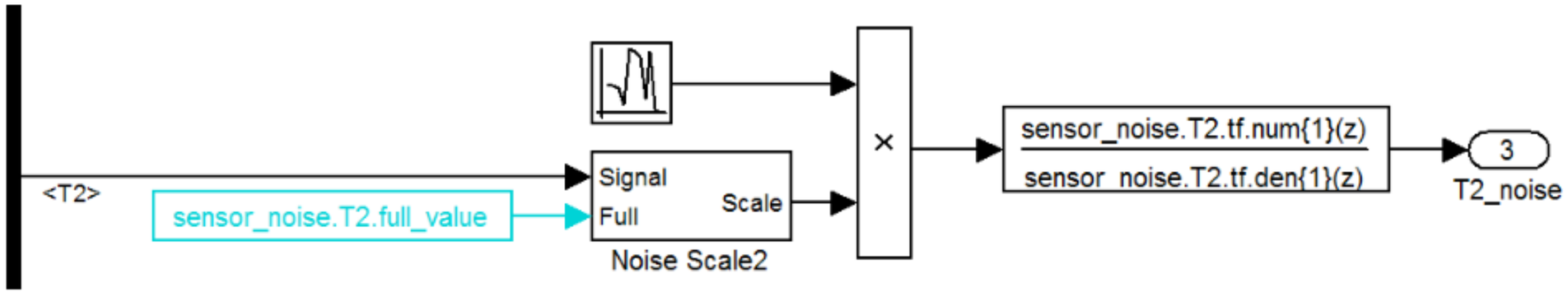
Implementation: C-MAPSS

- 90,000 lbf turbofan simulation
  - Implemented in Simulink
- Fan speed controlled
- Three actuators
  - Fuel flow (Wf)
  - Variable Bleed Valve (VBV)
  - Variable Stator Vanes (VSV)

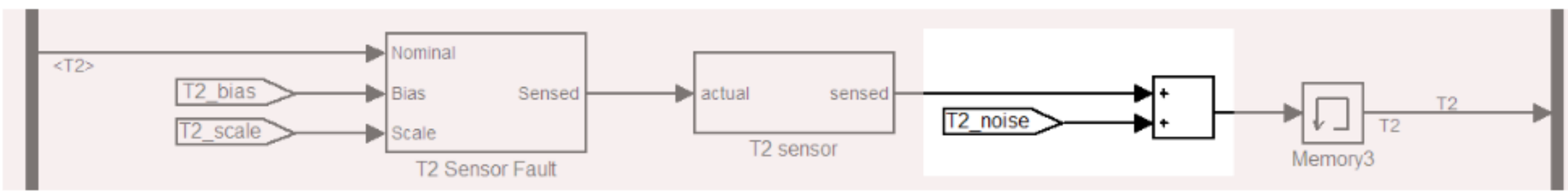


Implementation: Noise

- Tuned against actual flight data

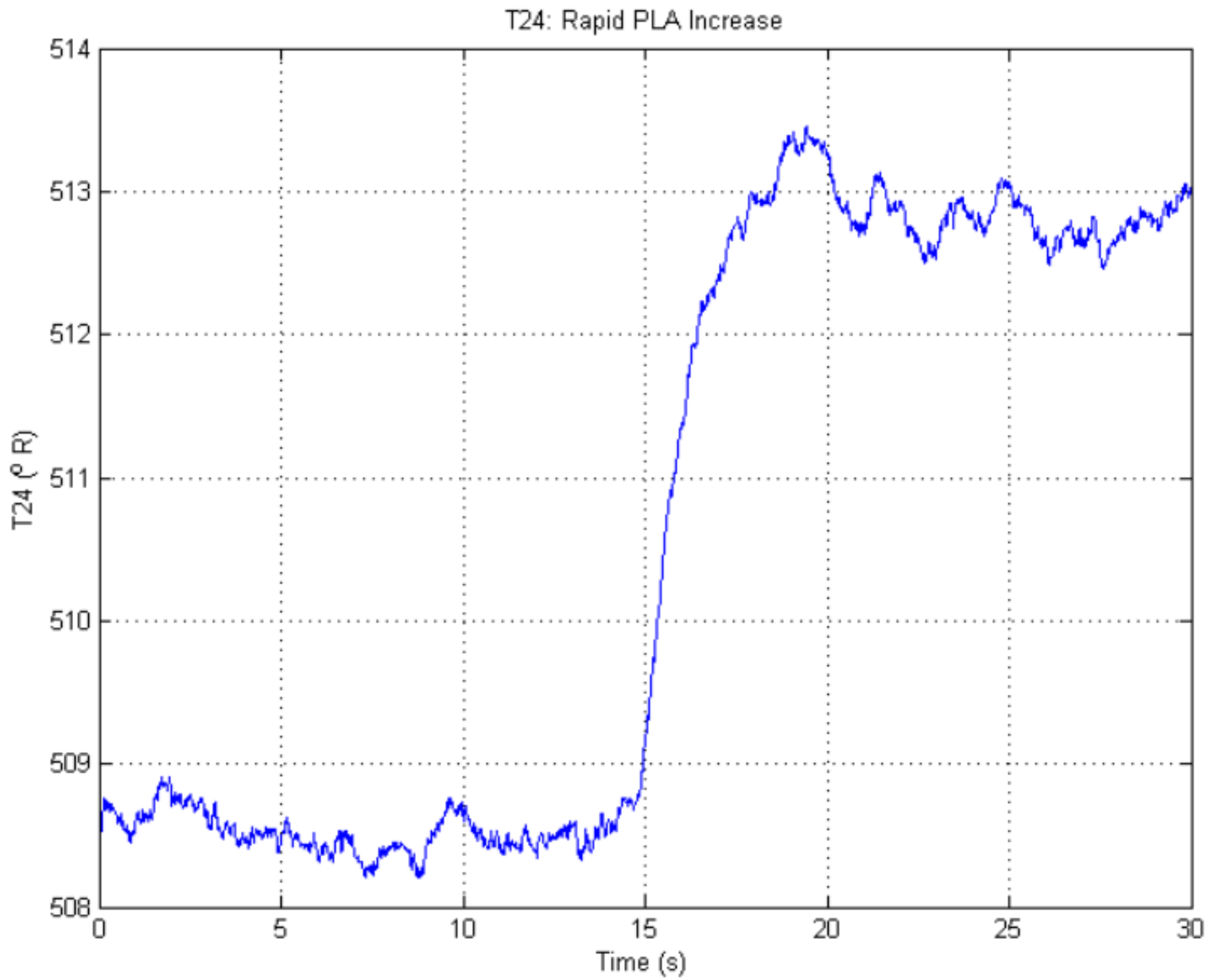


- Implemented on all C-MAPSS control sensors

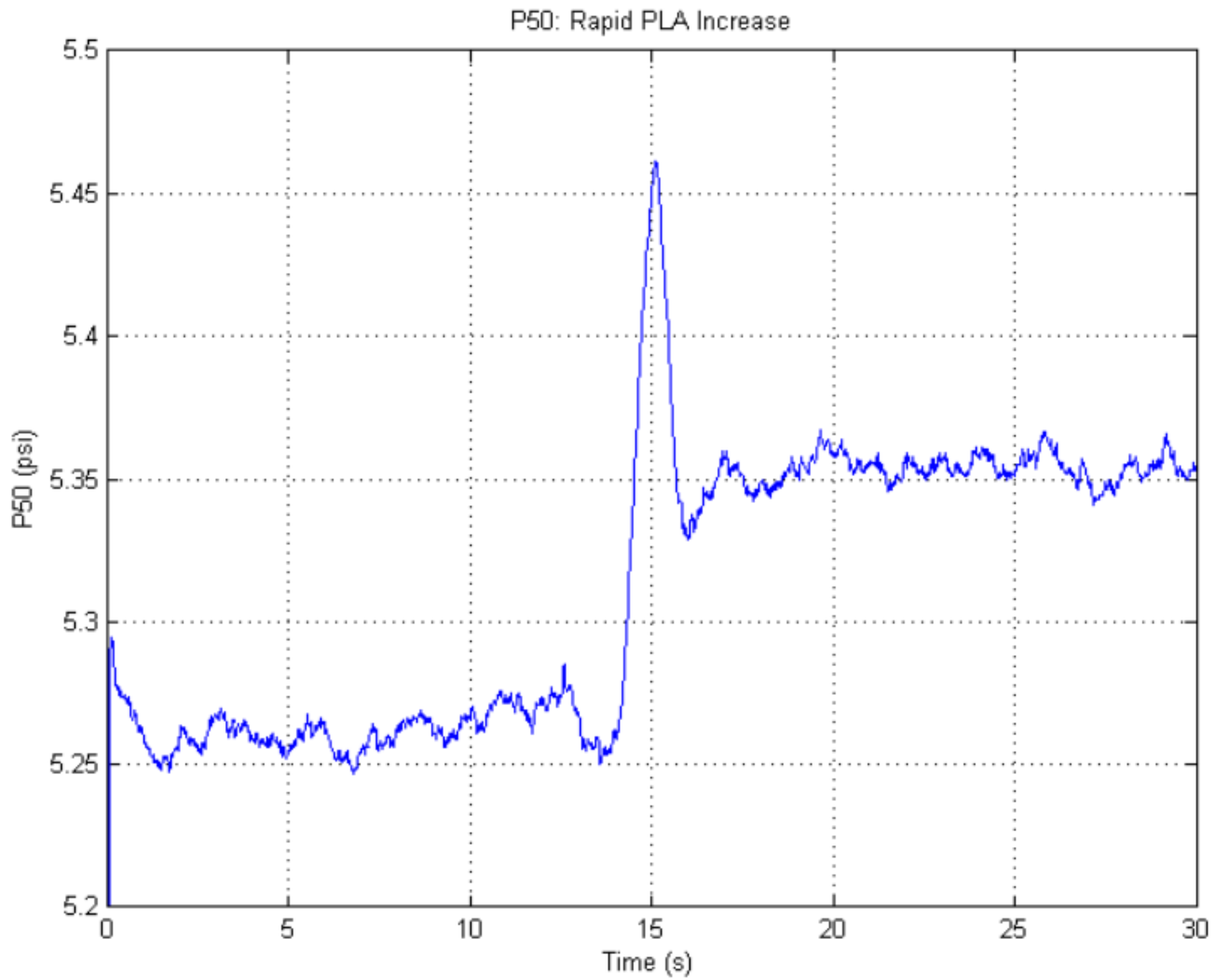


- Within control loop

Implementation: Noise (T24)

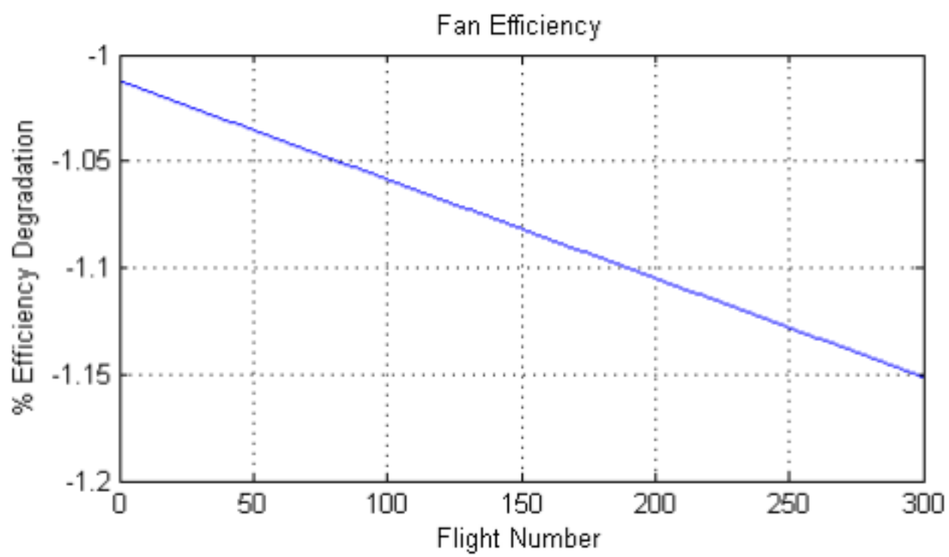


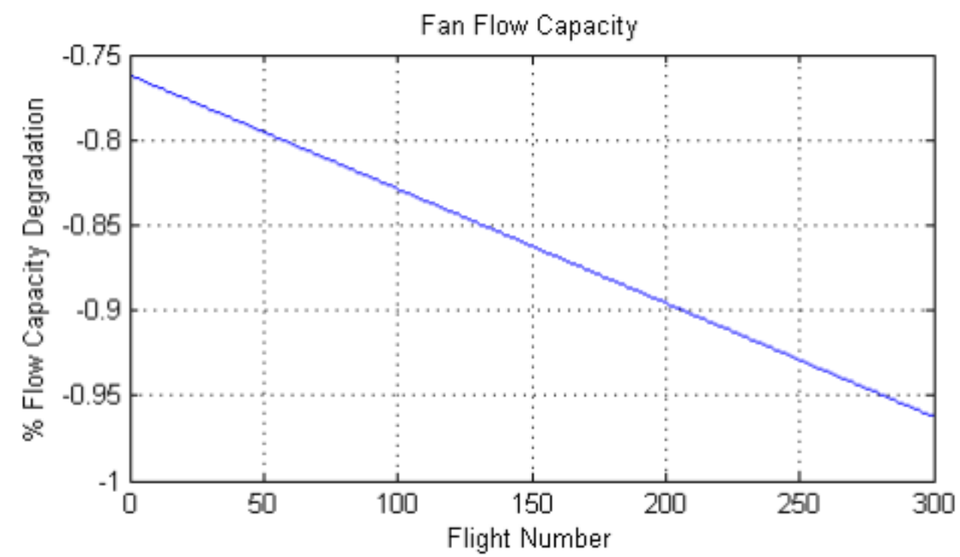
Implementation: Noise (P50)



Implementation: Health Trends

- Engine deterioration based on ProDiMES[1]
- Constant within flight (nominally)
- Gradual deterioration across flights





[1] Simon, D. L., et al, *Benchmarking Gas Path Diagnostic Methods: A Public Approach*, NASA TM 2008-215271, July 2008.

Engine Faults

- Sensor Faults
- Actuator Faults
- Turbomachinery Component Faults

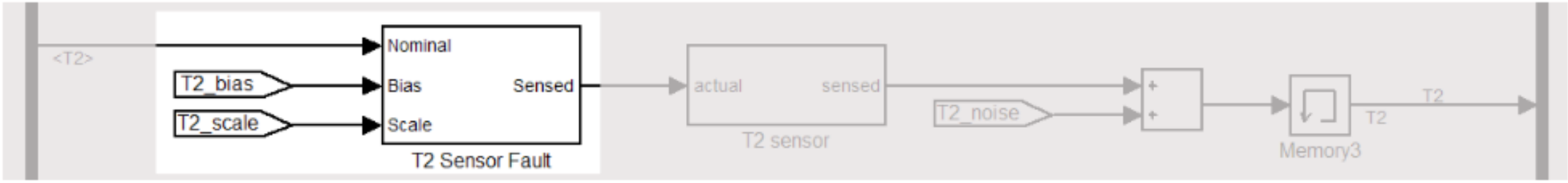
All faults are isolated to a single profile

Sensor Faults

- Implemented on all 8 C-MAPSS control sensors

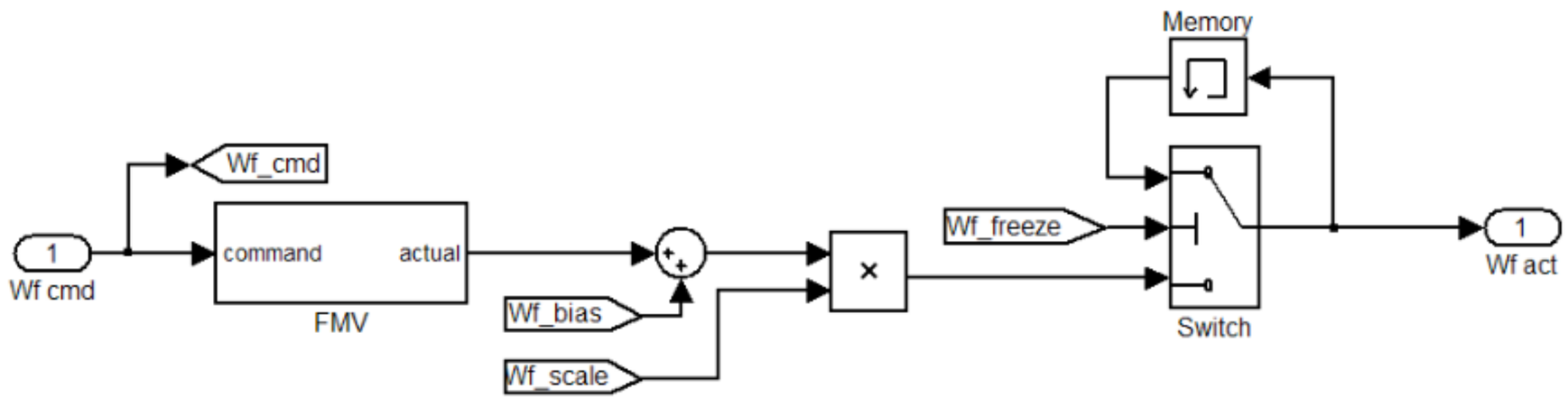
Rotational	Nf	Nc	
Pressure	P2	P50	Ps30
Temperature	T2	T24	T48

- Bias (absolute and relative), Drift, Failure



Actuator Faults

- Fuel Flow, Variable Bleed Valve, Variable Stator Vanes
- Bias (absolute and relative), Lockup



Turbomachinery Component Faults

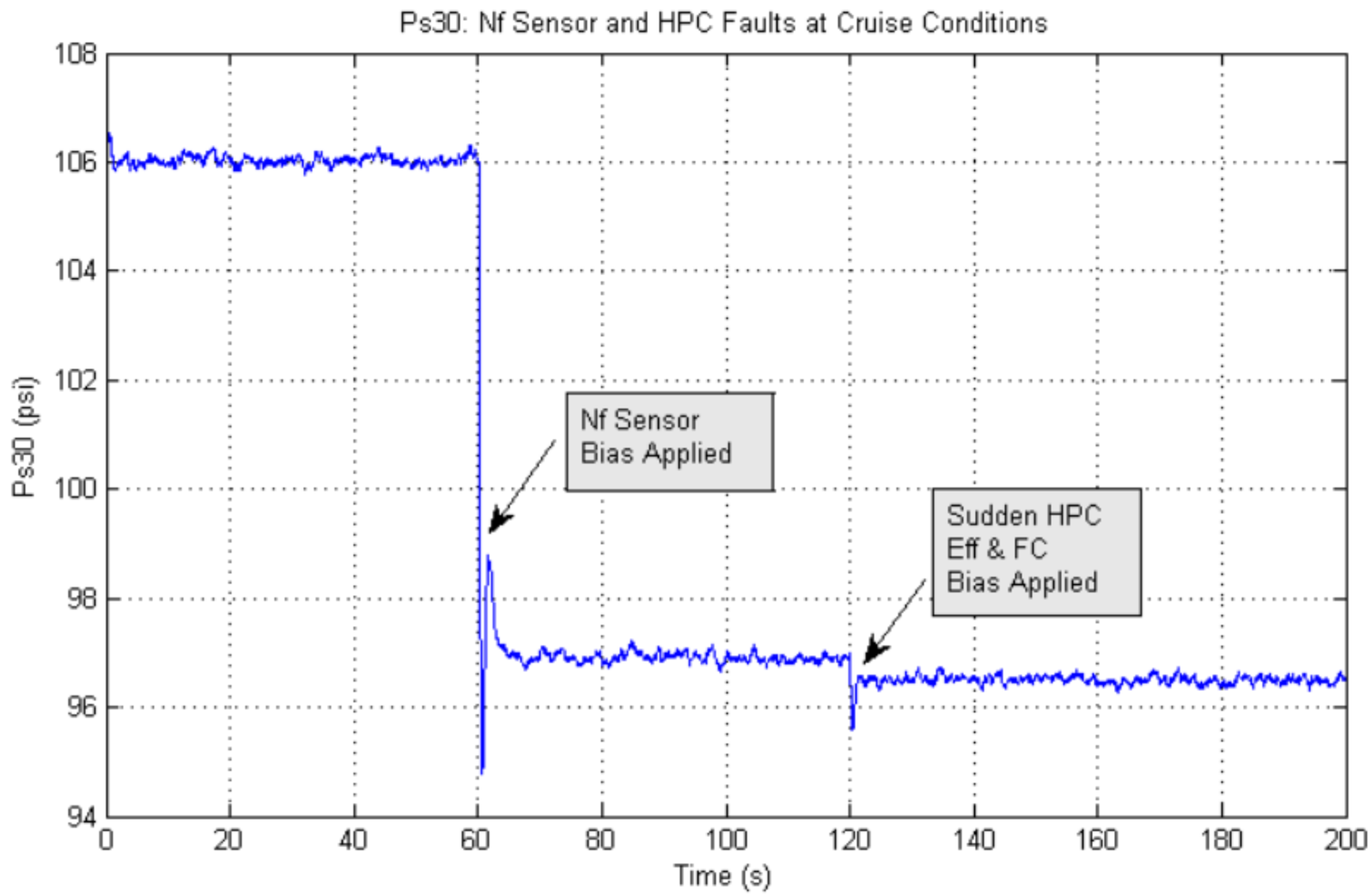
- Implemented through 10 health parameters

  Fan LPC HPC HPT LPT

- Efficiency and flow capacity modifiers for each component
- Bias only

Multiple Faults

- Ability to introduce multiple faults per profile



Diagnostic Algorithm Evaluation

- Single Profile Detection/Isolation
- Engine History Detection/Isolation
  - Including possible training
- Benchmarking

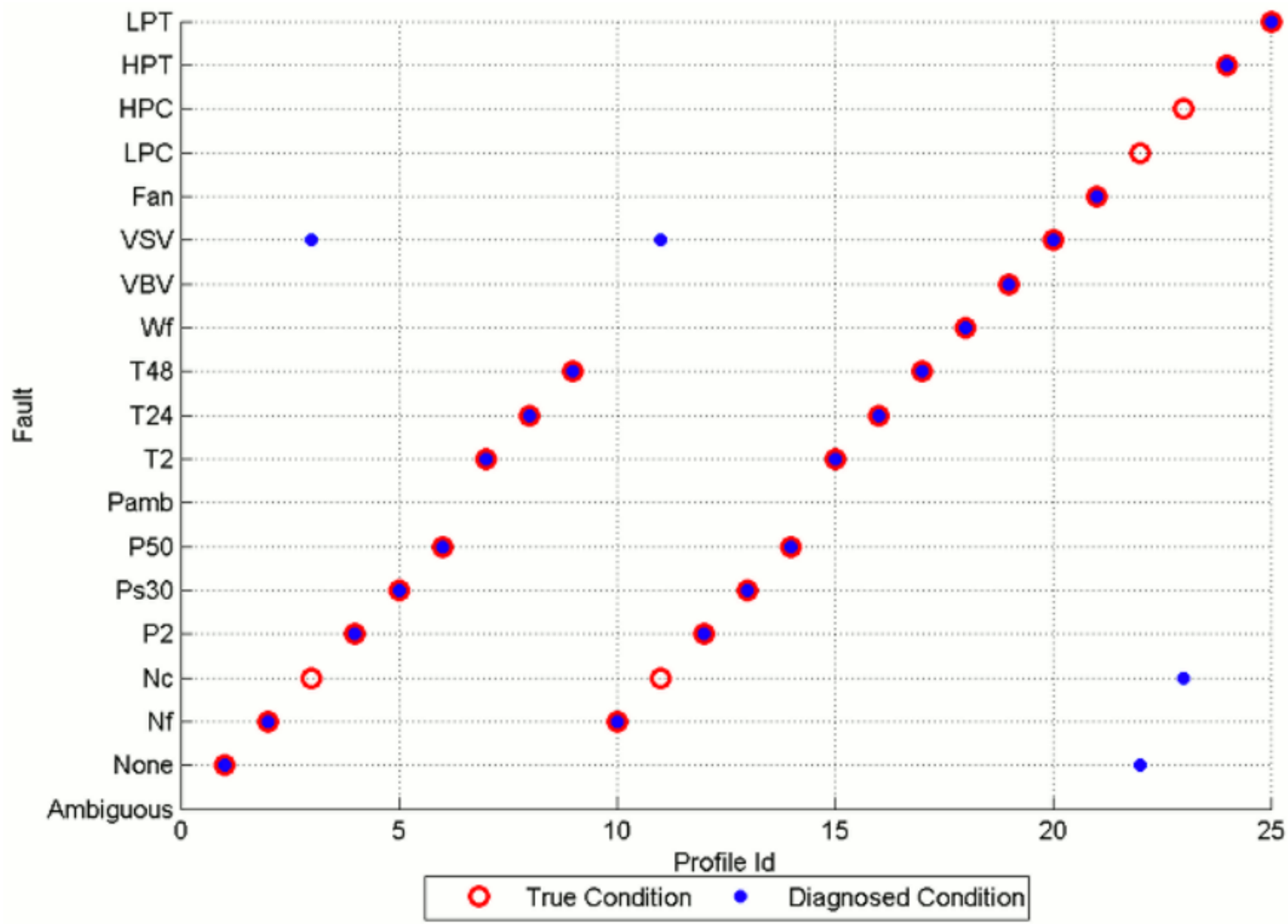
Command-line metrics

```
>> [correct_detection, false_positive, correct_classification, misclassification] = ...
        test_profile(@fault_detection, p)

correct_detection =

     1

false_positive =

     0

correct_classification =

   Empty cell array: 0-by-1

misclassification =

    'HPT'    [10.0200]
```

```
>> s = test_history(@fault_detection,get_meta('sample',1),'sample',0,'PassHealth')

s =
       false_positive: {0x1 cell}
    correct_detection: {24x1 cell}
     missed_detection: {1x0 cell}
       classification: [1x25 struct]
           fault_data: {25x2 cell}
```

Diagnostic Algorithm Evaluation

Graphical metrics



Benchmarking

A small subset of tools are included to benchmark a detection/isolation algorithm under controlled conditions.

- Generates profiles containing each possible fault
  - Multiple operating points (altitude, mach, power)
  - Quasi-steady conditions
- All profiles fed through detection/isolation algorithm
- Consolidated results can be processed into a *Confusion Matrix*

Example Diagnostic Algorithm

Piecewise Linear Detection & Isolation

- Based on a piecewise linear approximation of C-MAPSS
- Difference between piecewise linear model and sensors

$$\Delta \vec{y} = \vec{\hat{y}}_{PWL} - \vec{y}_{sensed}$$

- Detection: Threshold-based
  - 10 seconds of violations registers as fault
  - Triggers isolation algorithm

Example Diagnostic Algorithm

- Underdetermined problem
  - 8 sensors to detect 17 possible fault sources
- Isolation utilizes a linear "influence" matrix, *H*
  - Based on sensor residuals, test all fault hypotheses

$$\Delta \hat{x}_i = f(H, \Delta \vec{y}, \ldots)_i \longrightarrow \hat{\Delta \vec{y}} = \begin{bmatrix} H_{1,i} \\ \vdots \\ H_{m,i} \end{bmatrix} \Delta \hat{x}_i$$

- Fault source is isolated as the smallest error in hypothesized residuals

Example Diagnostic Algorithm

Results for 210 operating points including engine degradation and sensor noise

True\\Predicted	Nf	Nc	P2	Ps30	P50	Pamb	T2	T24	T48	Wf	VBV	VSV	Fan	LPC	HPC	HPT	LPT	None	Ambiguous
Nf	**0.82**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	*0.04*	0.00	0.00	*0.14*	0.00
Nc	0.00	**0.57**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	*0.00*	0.00	0.00	*0.00*	*0.08*	*0.04*	*0.30*	0.00
P2	0.00	*0.07*	**0.84**	0.00	0.00	0.00	0.00	0.00	0.00	*0.01*	0.00	0.00	*0.04*	*0.00*	0.00	0.00	0.00	*0.03*	0.00
Ps30	0.00	0.00	0.00	**0.97**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	*0.01*	0.00	0.00	*0.02*	0.00
P50	0.00	0.00	0.00	0.00	**0.96**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	*0.04*	0.00
Pamb	0.00	0.00	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
T2	0.00	0.00	0.00	0.00	0.00	0.00	**0.96**	0.00	*0.01*	0.00	0.00	0.00	0.00	0.00	*0.00*	0.00	0.00	*0.02*	0.00
T24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.89**	0.00	0.00	*0.04*	0.00	0.00	*0.00*	0.00	0.00	0.00	*0.07*	0.00
T48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.72**	0.00	0.00	0.00	0.00	0.00	*0.00*	0.00	0.00	*0.27*	0.00
Wf	0.00	0.00	0.00	*0.04*	0.00	0.00	0.00	0.00	*0.00*	**0.68**	0.00	0.00	*0.00*	0.00	0.00	0.00	0.00	*0.28*	0.00
VBV	0.00	*0.01*	*0.00*	0.00	0.00	0.00	0.00	*0.01*	*0.00*	0.00	**0.45**	*0.00*	0.00	*0.48*	*0.00*	0.00	0.00	*0.03*	0.00
VSV	0.00	*0.76*	*0.06*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.02**	*0.00*	*0.06*	0.00	0.00	0.00	*0.10*	0.00
Fan	*0.28*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.60**	0.00	0.00	*0.00*	0.00	*0.11*	0.00
LPC	0.00	*0.09*	*0.03*	0.00	*0.04*	0.00	0.00	0.00	*0.00*	0.00	0.00	0.00	0.00	**0.38**	*0.02*	0.00	0.00	*0.43*	0.00
HPC	0.00	*0.01*	*0.00*	0.00	0.00	0.00	0.00	0.00	*0.13*	0.00	0.00	0.00	0.00	0.00	**0.84**	0.00	0.00	*0.01*	0.00
HPT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	*0.00*	**1.00**	0.00	0.00	0.00
LPT	0.00	*0.22*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	*0.04*	0.00	*0.01*	0.00	0.00	**0.68**	*0.04*	0.00
None	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00

Conclusion

Transient Test Case Generator

- Produces realistic engine data, including sensor noise and engine health effects
- Incorporates sensor, actuator, and turbomachinery component faults
- Includes diagnostic algorithm evaluation tools
  - Algorithm training
  - Quantifiable detection and isolation metrics
  - Benchmarking
- Future public release planned

Conclusion

Piecewise Linear Detection & Isolation

- Included as a sample solution with the Transient Test Case Generator
- Fault detection based on threshhold violations
- Fault isolation of underdetermined problem using sensed-vs.-model residuals
- Reasonably successful based on initial testing
- Requires an engine health estimation scheme