Dynamic Data Driven Avionics Systems: Inferring Failure Modes from Data Streams

Carlos A. Varela
Worldwide Computing Laboratory, Department of Computer Science,
Rensselaer Polytechnic Institute, Troy, NY, U.S.A.
cvarela@cs.rpi.edu
http://wcl.cs.rpi.edu/

Dynamic Data-Driven Avionics Systems (DDDAS) embody ideas from the Dynamic Data-Driven Application Systems paradigm by creating a data-driven feedback loop that analyzes spatio-temporal data streams coming from aircraft sensors and instruments, looks for errors in the data signaling potential failure modes, and corrects for erroneous data when possible.

As we saw with the 2009 crash of Air France flight 447 [Bureau d'Enquêtes et d'Analyses pour la Sécurité de l'Aviation Civile, 2012], these sensors literally protect lives, and if not detected, a data error can result in tragedy. Our research on dynamic data driven modeling and instrumentation, uses established mathematical relationships between different data streams to monitor incoming data and detect errors. In the case of flight 447, the groundspeed and wind speed data– which was available from onboard GPS monitors, and weather forecasts — could have been used to catch erroneous air speed readings coming from malfunctioning pitot tubes. If the relationship is clear, and there are many examples in our current technology where it is, our active data monitoring approach embodied in our PILOTS [1] programming language and software [Imai, Klockowski, Varela, Self-Healing Spatio-Temporal Data Streams Using Error Signatures, 2013] [Imai, Galli, Varela, Dynamic Data-Driven Avionics Systems: Inferring Failure Modes from Data Streams, 2015] is an excellent way to catch and correct errors before they become a problem, and ensure continued safe use of technology that relies on accurate data.

We exploit logical redundancy that exists between failure-independent data sensor sources in order to detect errors and to recover original data. We define an error function to capture the redundancy between input variables. The key idea is to recognize the shape of the error function on known failures by using *error signatures*, identify an erroneous variable, and finally compute a new value for the erroneous variable from redundant data if possible. An overview of the error signature-based error detection method is summarized in **Figure 1**.

We have designed and implemented a stream processing system, namely PILOTS, that provides users a high-level application programming environment for error-tolerant stream processing. **Figure 2** shows the high-level system architecture of the PILOTS runtime system. The *PILOTS application* is written in the PILOTS programming language and takes inputs from *aircraft sensors*. The *error analyzer* collects the latest ω error values from the PILOTS application and keeps detecting failures based on known *error signatures*. If it detects a recoverable error, it replaces an erroneous input with the correct one by applying a corresponding error correction equation. Finally, the PILOTS application computes output streams based on the corrected inputs produced from the error analyzer.

A PILOTS program called SpeedCheck implementing the error signatures for Air France 447 is presented in Figure 3. It has been used to successfully recover the airspeed in five seconds from pitot tube failure using data available in the plane's black box. We have also applied PILOTS to the Tuninter 1153 (TU1153) flight accident in August 2005, where the installation of an incorrect fuel sensor led to a fatal accident. The underweight condition is successfully detected with 100%

[1] ProgrammIng Language for spatiO-Temporal data Streaming applications (http://wcl.cs.rpi.edu/pilots/)

accuracy during cruise flight phases. Adding logical redundancy to avionics through a dynamic data-driven approach can significantly improve the safety of flight.

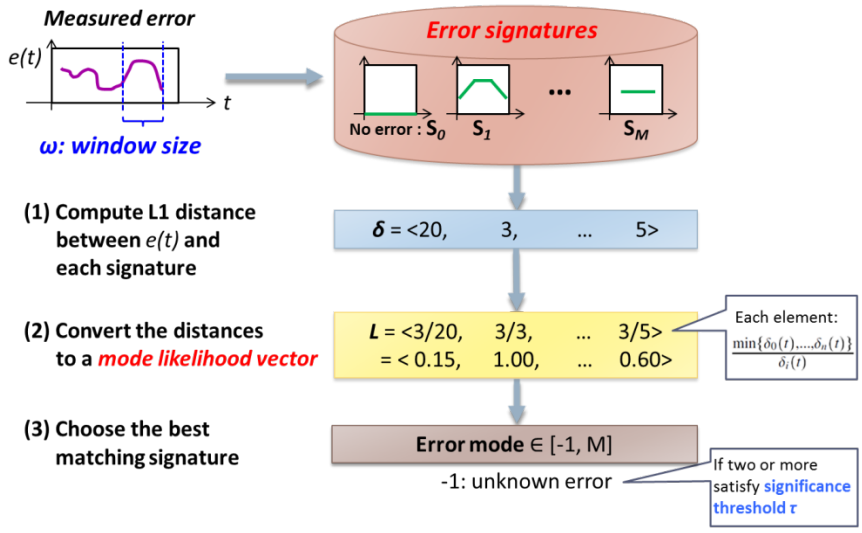
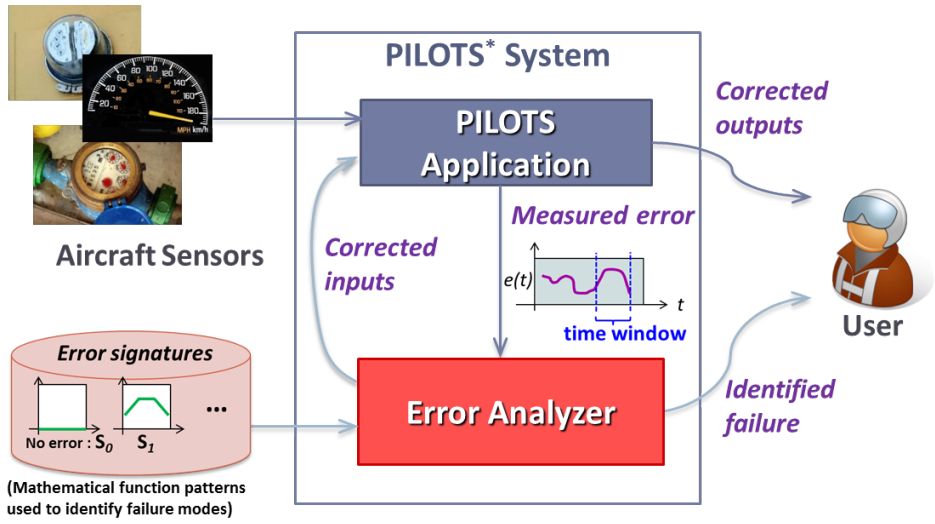


Figure 1. Error signature-based error detection method.



*: Programming Language for spatiO-Temporal data Streaming applications

Figure 2. High-level architecture of the PILOTS system.

```
program SpeedCheck;
  /* v_w: wind speed,          a_w: wind speed angle
     v_a: airspeed,            a_a: airspeed angle
     v_g: ground speed,        a_g: ground angle */
inputs
  v_w, a_w (x,y,z,t) using euclidean(x,y), closest(t), interpolate(z,2);
  v_a, a_a (x,y,t)   using euclidean(x,y), closest(t);
  v_g, a_g (x,y,t)   using euclidean(x,y), closest(t);
outputs
  crab_angle : arcsin(v_w * sin (a_w - a_a) /
               sqrt(v_a^2 + 2 * v_a * v_w * cos(a_w - a_a) + v_w ^2))
               at every 1 sec;
errors
  e: v_g - sqrt(v_a^2 + v_w^2 + 2 * v_a * v_w * cos(a_w - a_a));
signatures
  /* v_a = 470 knots */
  S0(k):e=k, -16.2 <=k, k <= 16.2        "Normal";
  S1(k):e=k,  91.8 <=k, k <= 145.8       "Pitot tube failure";
  S2(k):e=k,-178.2 <=k, k <= -145.8      "GPS failure";
  S3(k):e=k, -70.2 <=k, k <= -16.2       "Both failures";
correct
  S1:v_a = sqrt(v_g ^2 + v_w^2 + 2 * v_g * v_w * cos(a_g - a_w));
  S2:v_g = sqrt(v_a ^2 + v_w^2 + 2 * v_a * v_w * cos(a_w - a_a));
end
```

Figure 3. PILOTS program to detect and correct for speed data failures.

ACKNOWLEDGEMENTS

This research is partially supported by the DDDAS program of the Air Force Office of Scientific Research, Grant No. FA9550-15-1-0214 , an NSF EAGER-Dynamic Data award No. 1462342, and a Yamada Corporation Fellowship.

REFERENCES

Imai, S., & Varela, C. A. (2012). Programming spatio-temporal data streaming applications with high-level specifications. *Proceedings of the Third ACM SIGSPATIAL International Workshop on Querying and Mining Uncertain Spatio-Temporal Data*, (pp. 18-25). Redondo Beach, California.

Imai, S., Galli, A., & Varela, C. A. (2015). Dynamic Data-Driven Avionics Systems: Inferring Failure Modes from Data Streams. *Dynamic Data-Driven Application Systems (DDDAS 2015)*. Reykjavik, Iceland.

Imai, S., Klockowski, R., & Varela, C. A. (2013). Self-Healing Spatio-Temporal Data Streams Using Error Signatures. *2nd International Conference on Big Data Science and Engineering (BDSE 2013)*, (pp. 957-964). Sydney, Australia.