Technical Area 3: Control Software

John Rushby

Computer Science Laboratory
SRI International
Menlo Park, CA
Overview

- Assured sensor fusion using interval representations
- Synthetic sensors
- Controller synthesis with a safety envelope
Sensor Fusion

- Flawed sensor fusion (in the presence of faults) is a major source of accidents and incidents in commercial aircraft
  - Airbus A330 accident, Learmonth, 2008: 3 AOA sensors
  - Boeing 777 upset, Perth, 2005: 7 accelerometers

- Because of its difficulty, sometimes prefer not to use all available information
  - 737 crash, Schipol, 2009: single radar altimeter

- Rich opportunity for attackers: RQ-170 Sentinel over Iran

- So our first step is assured sensor fusion in the presence of faults and attacks
Communicating a Single Sensor Sample

- Traditional Approach: send a **single number**
  - Indicates best estimate, but not its quality

- Instead, send an **interval**
  - Nonfaulty sensor guarantees **true value is in this range**
  - **Width** of interval indicates **quality**
  - Embellishment: interval is a function of **time since sample**
  - Possibly a **use-by** time also
Fusing Multiple Point Samples

Traditional Approach (e.g., with 3 samples)

**Fusing for a single value:**
Mid-value select when 3, average when 2

**Eliminating faulty samples:**
Reject if not within 15% of the others

Problems: thumps and bad values, and worse
Experience: X29

- Three sources of air data: a nose probe and two side probes
- Selection algorithm used the data from the nose probe, provided it was within some threshold of the data from both side probes
- The threshold was large to accommodate position errors in certain flight modes
- Belated discovery: if nose probe failed to zero at low speed, it would still be within the threshold of correct readings, causing the aircraft to become unstable and “depart”
- 162 flights had been at risk
- Recent methods use more complex selection algorithms
- Take the dynamics into account
- Generally validated by Matlab simulations
Fusing Multiple Interval Samples

**Theorem:** true value must be in overlap of nonfaulty intervals

**Calculating consensus interval:** to tolerate $f$ faults in $n$, choose interval that contains all overlaps of $n - f$; i.e., from least value contained in $n - f$ intervals to largest value contained in $n - f$ (Marzullo)

An interesting small exercise in formal verification (finite sets, predicate subtypes, dependent types)

**Eliminating faulty samples:** separate problem, not needed for fusing, but any sample disjoint from the consensus interval must be faulty
True Value In Overlap Of Nonfaulty Intervals

\[ S(1) \]
\[ S(2) \]
\[ S(3) \]
\[ S(4) \]
Marzullo’s Fusion Interval

$S(1)$

$S(2)$

$S(3)$

$S(4)$
Marzullo’s Fusion Interval: Fails Lipschitz Condition

\[ S(1) \]
\[ S(3) \]
\[ S(4) \]
Schmid’s Fusion Interval

• Choose interval from \( f + 1 \)'st largest lower bound to \( f + 1 \)'st smallest upper bound

• **Optimal** among selections that satisfy Lipschitz Condition
Schmid’s Fusion Interval

$S(1)$  $S(2)$  $S(3)$  $S(4)$
Synthetic Sensors

- Once we can safely fuse sensors, we can use many of them
- Even imprecise sensors can add value
- Make use of all available information: synthesize new sensors
  
  - e.g., estimate distance from engine performance and time as well as from wheel sensors
  
  - Estimate fuel/power remaining by similar means
  
  - Radio call signs may suggest whether you are over Afghanistan or Iran
Safe Control

- We now have a lot of sensor information
- Reliably fused
- And dependable monitors for safety violations (from TA2)
- Wish to synthesize controllers to keep within safe region
- In the context of hybrid systems
Controller Synthesis With A Safety Envelope

- Synthesize a safety envelope
  - Invariants are a good start
  - Linear systems: left eigenvectors of the A matrix
  - Others: template methods using EF solving (from TA2)

- Then do certificate-based controller verification and synthesis
  - i.e., controller synthesis for a safety objective—in contrast to that for more traditional objectives (stability etc.)
  - Controller uses mode switches to keep plant within safety envelope
  - More EF solving, searching for witnesses such as invariant, Lyapunov function

- Need a DSL to specify this, including distinction between plant and controller, time-triggered interaction, etc.
  - Will extend HybridSAL (to HybridSAL-X) for this
Plan

- Develop HybridSAL-X and its toolset, including safety envelope and certificate-based controller verification and synthesis
  - Ashish Tiwari

- And methods and tools for synthetic sensors and assured fusion using intervals
  - Shankar