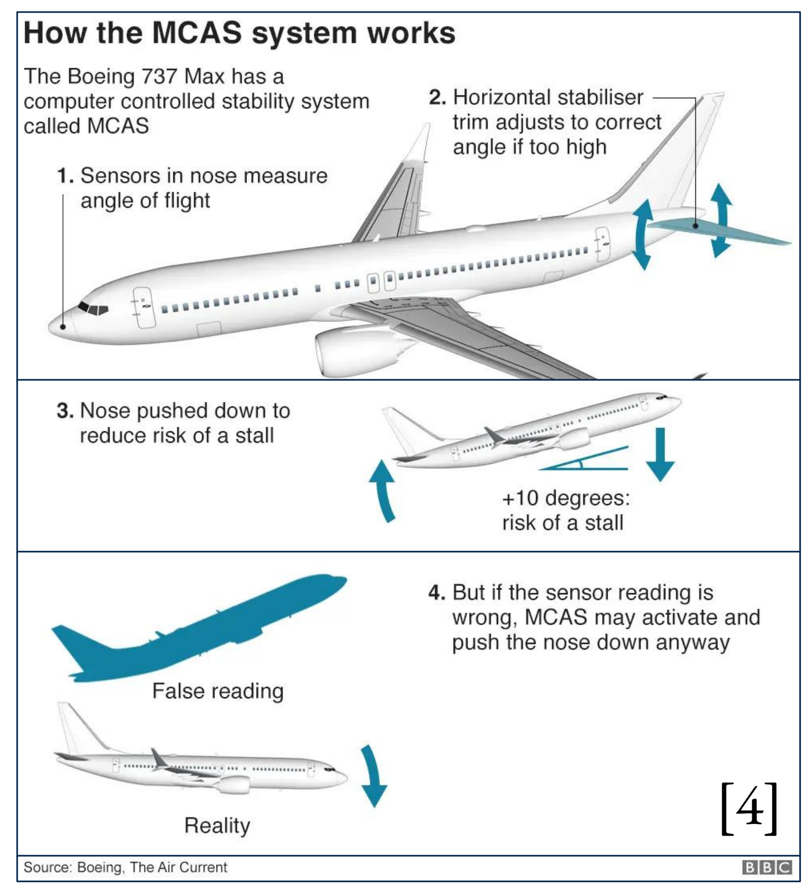
Analysis and Prevention of MCAS-Induced Crashes

Noah T. Curran, Thomas W. Kennings, Kang G. Shin
University of Michigan–Ann Arbor, Department of Computer Science and Engineering

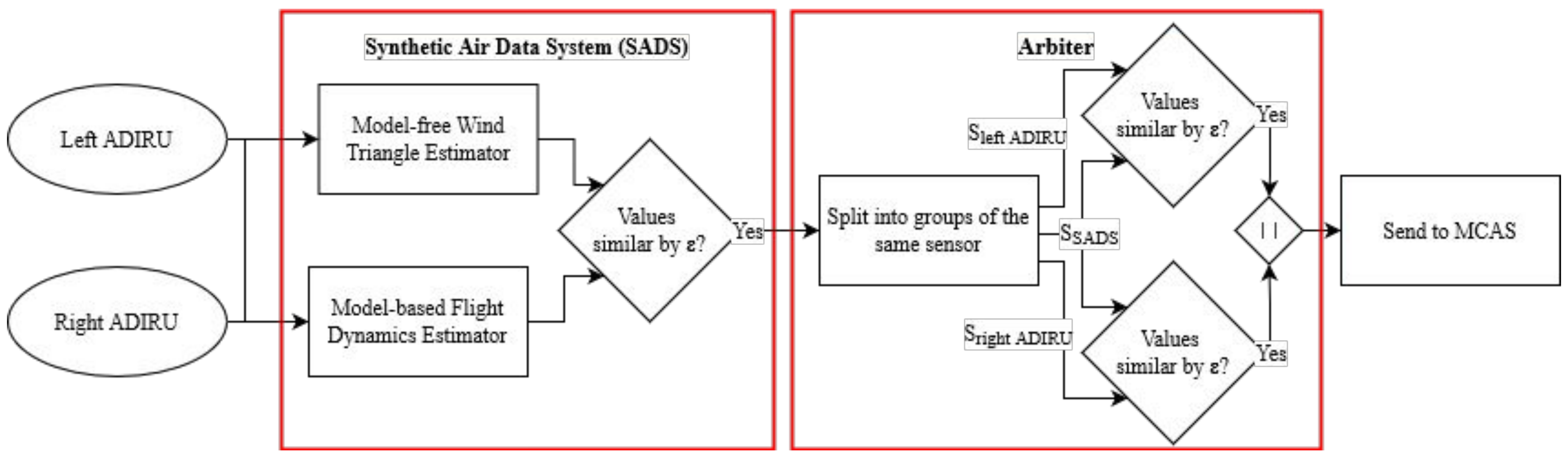
Background — 1

- Maneuvering Characteristics Augmentation System (MCAS) [1]
  - An autonomous flight pitch stabilizing controller used in Boeing 737-MAX
  - Introduced due to concerns related to pilots ability to safely fly the 737-MAX
- Software defect lead to crash and grounding of two 737-MAX aircraft [2]
- FAA grounded 737-MAX and required updates that resolve three issues [3]:
  1. All available AoA sensors must exceed 17° and cannot disagree >5.5°
  2. MCAS can activate only once in order to stop runaway stabilizer
  3. Pilots should be trained on how to manually disengage MCAS
- The update reverses the purpose of MCAS → prioritizes pilot control



[4]

Semi-Autonomous MCAS (SA-MCAS) — 2



Internal Consistency Check

- SADS model uses sensors collected by Air Data Inertial Reference Unit (ADIRU)
  - Includes AoA, noted with α
- Model-free: $\alpha = \tan^{-1}(u/v)$
- Model-based: $\alpha = f(C_l, M, h)$
- Check whether the estimated α is similar and pass onward to next stage if they are

External Consistency Check

- Compare α from ADIRU with α from SADS
- If the α from *either* the left or right ADIRU is similar to the SADS estimate, then pass the value to MCAS

Experimental Methods & Results — 3

- Perform experiments with Simulink that has:
  - Custom rule-based pilot controller
  - Sensor failure injection module
  - JSBSim flight dynamics engine [5]
  - MCAS control logic module
- Experiments include 9 sensor failure modes and 2 pilot failure modes
  - For each failure mode, there is a single parameter that is not constant
  - Goal is to find the first value for the parameter for which a crash occurs
- Compare the original MCAS ($MCAS_{old}$), the MCAS with the FAA's requirements ($MCAS_{new}$), and SA-MCAS

Stress Test \ MCAS	$MCAS_{old}$	$MCAS_{new}$	SA-MCAS
Sudden Val	17°	No failure	No failure
Sudden Duration	140.5450s	No failure	No failure
Sudden Recovery	2.7991s	No failure	No failure
Delta Val	13.8750°	No failure	No failure
Delta Duration	140.5450s	No failure	No failure
Delta Recovery	2.7991s	No failure	No failure
Gradual Linear	1.5000	No failure	No failure
Gradual Log	222.5000	No failure	No failure
Gradual Quadratic	1.4999	No failure	No failure
Stall Pitch	51.5497°	46.2531°	51.5497°
Stall Recovery	5.6333s	3.9084s	5.6333s

SA-MCAS is capable of securing the best of both $MCAS_{old}$ and $MCAS_{new}$, preventing crashes during sensor failures while also maintaining performance during dangerous pilot control.

Discussion, Future Work, & Conclusion — 4

- Discussion:
  - Passenger trust of autonomous control still needs to be regained
  - Limited ability to prevent dangerous pilot control due to limitations inherent to MCAS's control authority
- Future Work:
  - What do we do when neither the pilot nor the autonomous control is safe?
  - Currently we do equal to the better of autonomous/manual control, but can we do better?
- Takeaways:
  - Semi-autonomous systems should not default control to the manual operator or the autonomous controller
  - A good design philosophy should provide dynamic control authority depending on which option is safer
  - SA-MCAS demonstrates this point, showing that it can safely control the 737-MAX when either the pilot or MCAS is unsafe

References:
[1] D. Gates and M. Baker, "The Inside Story of MCAS: How Boeing's 737 MAX System Gained Power and Lost Safeguards," The Seattle Times, 2019.
[2] Federal Aviation Administration. (2019) Emergency Order of Prohibition. [Online]. Available: https://web.archive.org/web/20230417205843/https://www.faa.gov/news/updates/media/Emergency Order.pdf
[3] Boeing. (2019) 737 MAX Software Update. [Online]. Available: https://www.boeing.com/commercial/737max/737-max-software-updates.page
[4] BBC. (2019) Boeing 737 MAX Lion Air Crash Caused by Series of Failures. [Online]. Available: https://www.bbc.com/news/business-50177788
[5] J. Berndt, "JSBSim: An Open Source Flight Dynamics Model in C++," in AIAA Modeling and Simulation Technologies Conference and Exhibit, 2004.