0x104 – What is Computer Safety?

ECED4406 – Computer Security

Dr. Colin O'Flynn

Dalhousie University.

Let’s Look at Failures

Failure #1 - Therac-25

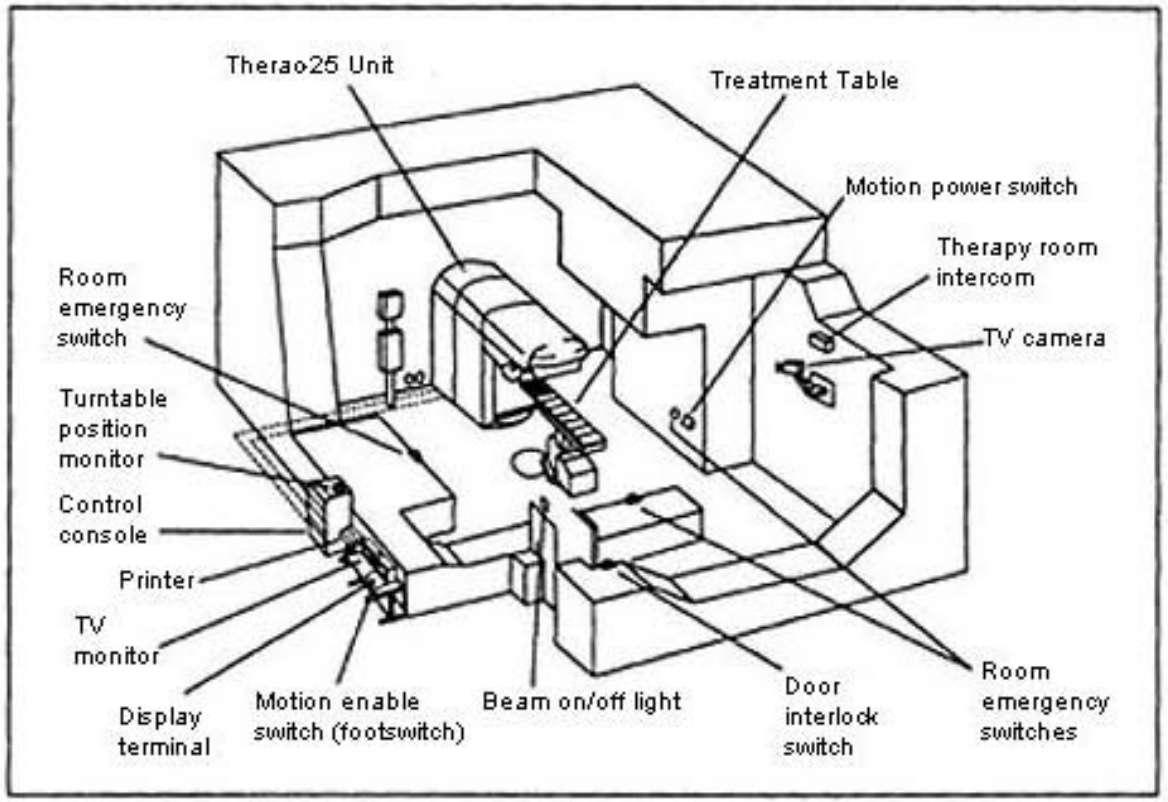


Figure 1. Typical Therac-25 facility

https://hackaday.com/2015/10/26/killed-by-a-machine-the-therac-25/
https://en.wikipedia.org/wiki/Therac-25
https://www.computer.org/csdl/magazine/co/2017/11/mco2017110008/13rRUxAStVR

Therac-25 Quick History

- New “Revolutionary” machine could deliver radiation in two modes:
  - Electrons (low-energy, good for attacking shallow tissue such as skin cancer).
  - X-Ray (high-energy, attacking deeper tissue such as lung cancer).

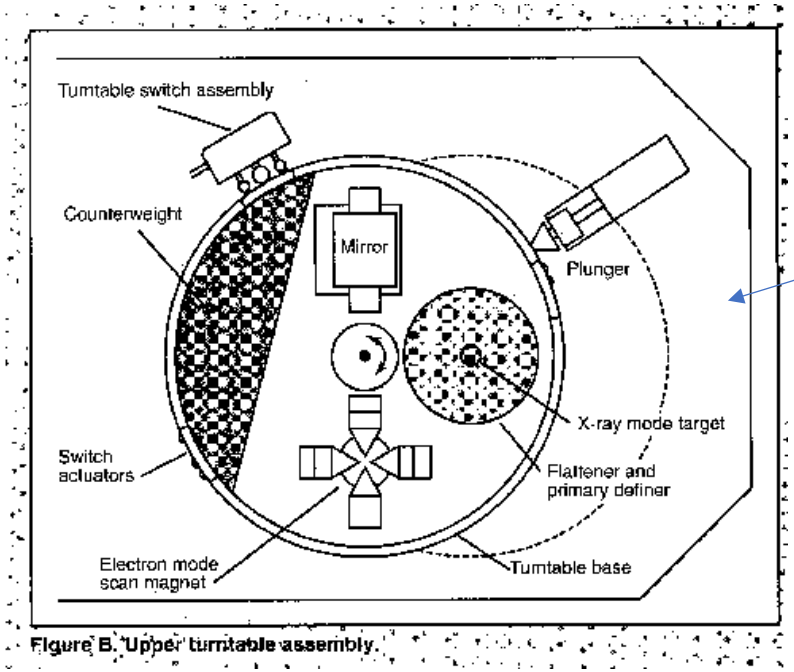
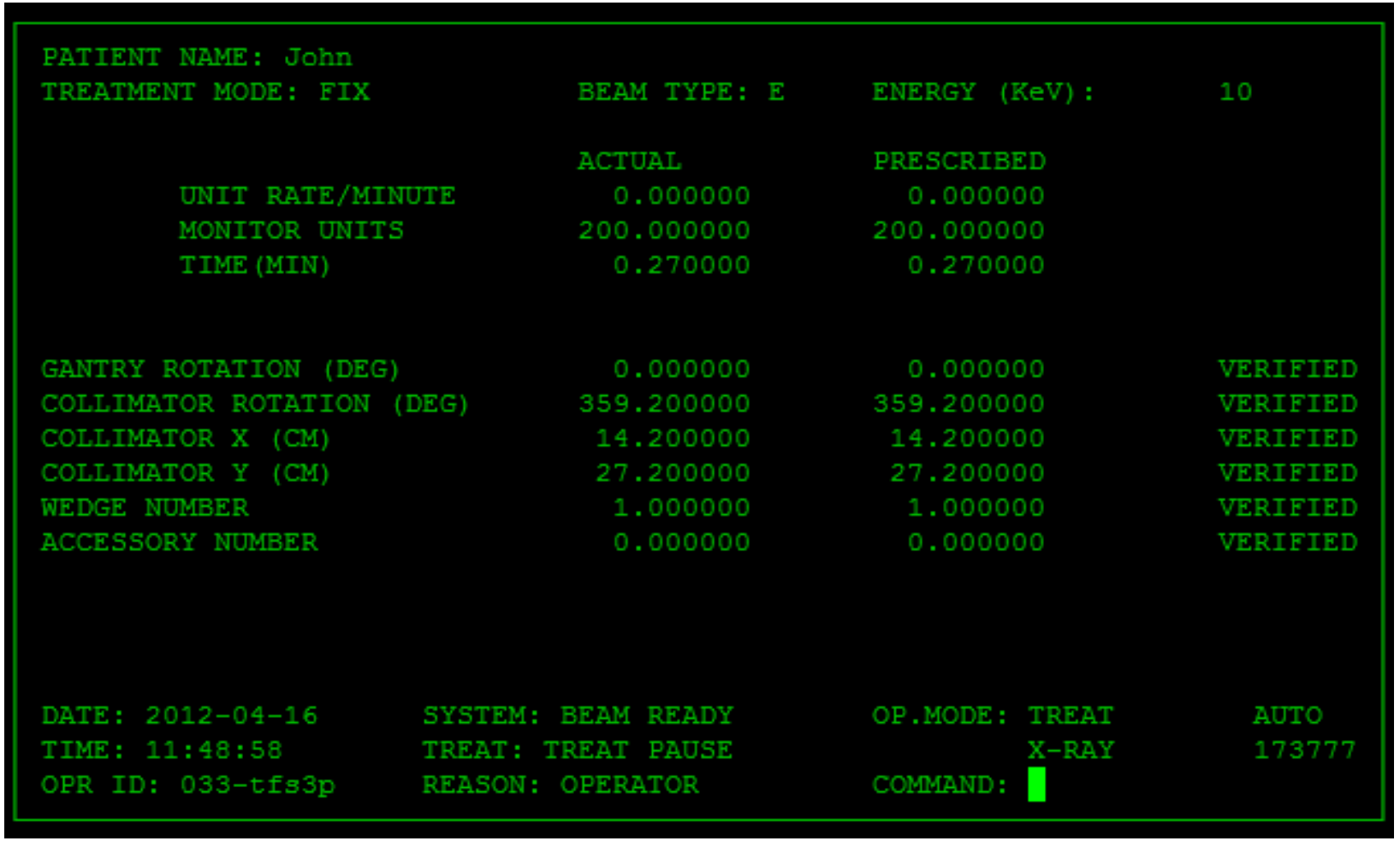


Figure B. Upper turntable assembly.

Turn-table rotates to insert into beam path.

Therac-25 User Interface



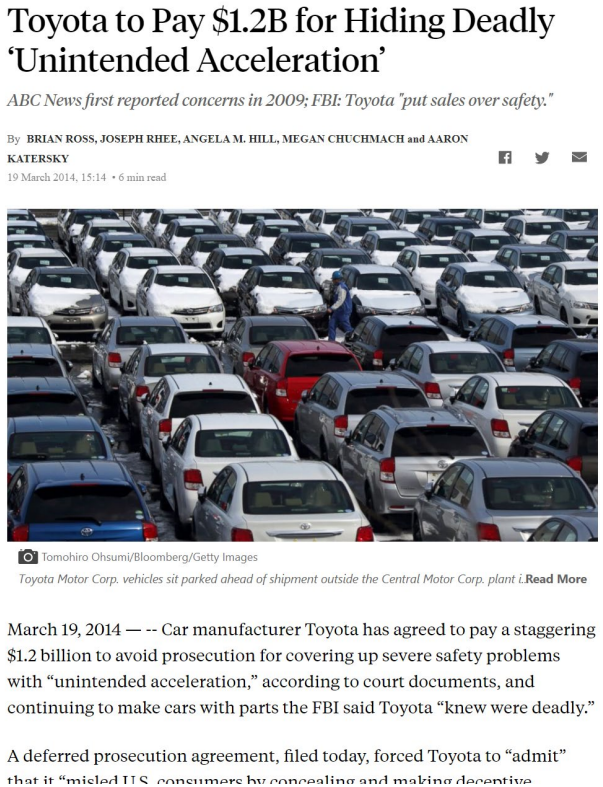
Therac-25 User Cost

- Six documented accidents where incorrect beam or energy delivered.
- **In three cases patients <u>died</u> as a result of radiation.**

Examples of Code Issues

- Hardware safety switches from previous versions were now done purely in software.
- Errors gave simple code # errors (‘MALFUNCTION 53’) without explaining what that meant. Operators got accustomed to just continuing because they came up frequently.
- Even critical errors could be ‘skipped’ by this process.

Failure #2 – Toyota Unintended Acceleration



Toyota to Pay $1.2B for Hiding Deadly 'Unintended Acceleration'

ABC News first reported concerns in 2009; FBI: Toyota "put sales over safety."

By BRIAN ROSS, JOSEPH RHEE, ANGELA M. HILL, MEGAN CHUCHMACH and AARON KATERSKY

19 March 2014, 15:14 • 6 min read

Tomohiro Ohsumi/Bloomberg/Getty Images

*Toyota Motor Corp. vehicles sit parked ahead of shipment outside the Central Motor Corp. plant i.*Read More

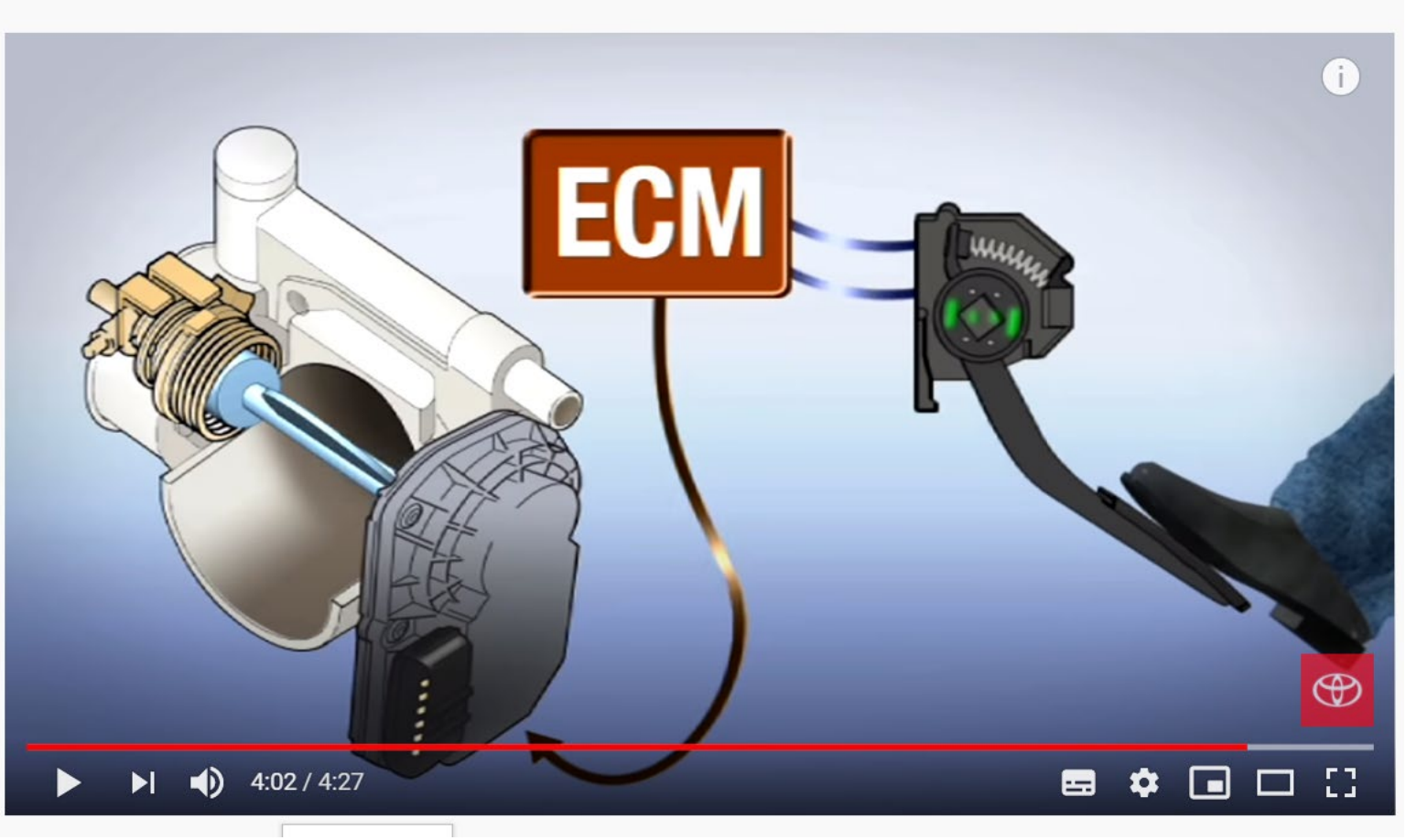
March 19, 2014 — -- Car manufacturer Toyota has agreed to pay a staggering $1.2 billion to avoid prosecution for covering up severe safety problems with "unintended acceleration," according to court documents, and continuing to make cars with parts the FBI said Toyota "knew were deadly."

A deferred prosecution agreement, filed today, forced Toyota to "admit" that it "misled U.S. consumers by concealing and making deceptive

- https://abcnews.go.com/Blotter/toyota-pay-12b-hiding-deadly-unintended-acceleration/story?id=22972214

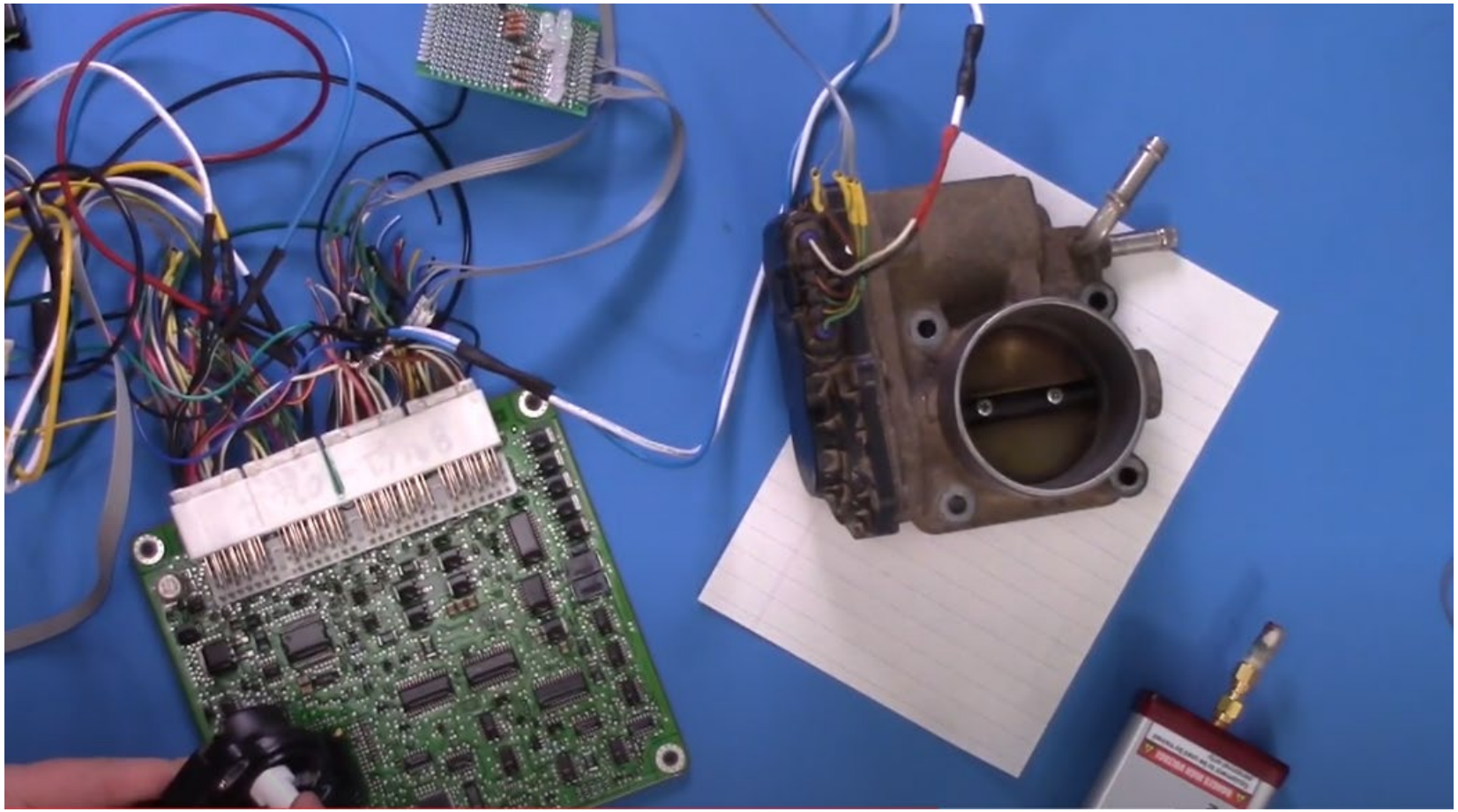
Small note – there is (for unknown reasons) many popular incorrect claims about this case I won't go into. Many are a result of people being unclear about how automotive systems work or performing tests incorrectly. Vehicles rely on vacuum system to provide braking power assist – on "true" full throttle brake assist is lost. Many cars back off from full throttle during braking to keep your power brakes working, but the vehicles in question do not do this. If you have used a car which looses power brakes it is a very hard pedal (you can turn a car off to experience this).

How does a Throttle Position Work



https://www.youtube.com/watch?v=Fn_cDN56Oy8S

How does it NOT work?

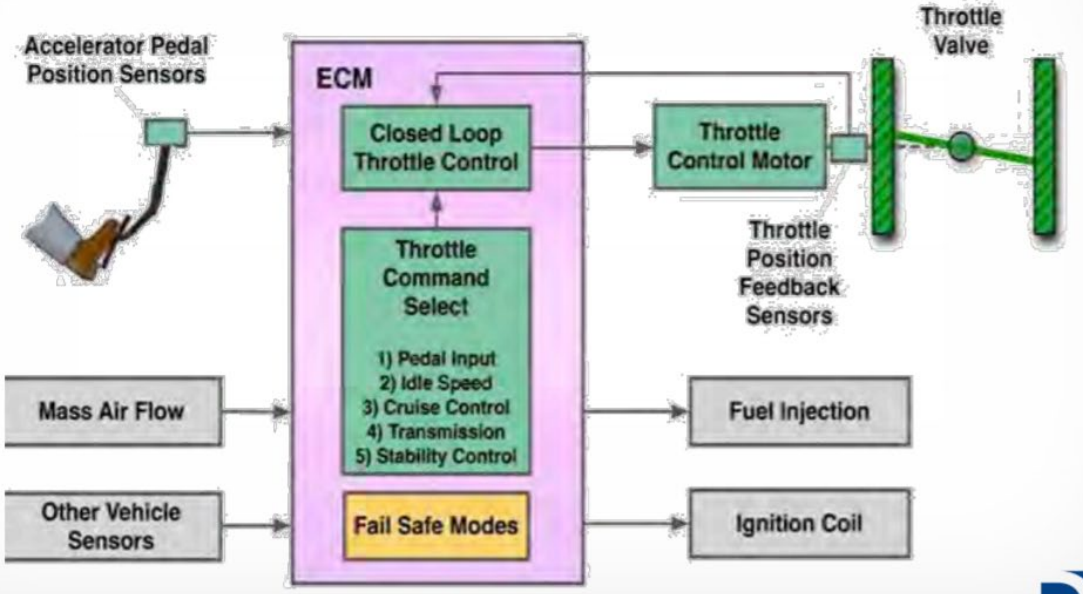


https://www.youtube.com/watch?v=26yulQc-7XM

How could this happen?

"Toyota ETCS-i is an example of a safety-critical hard real-time system.

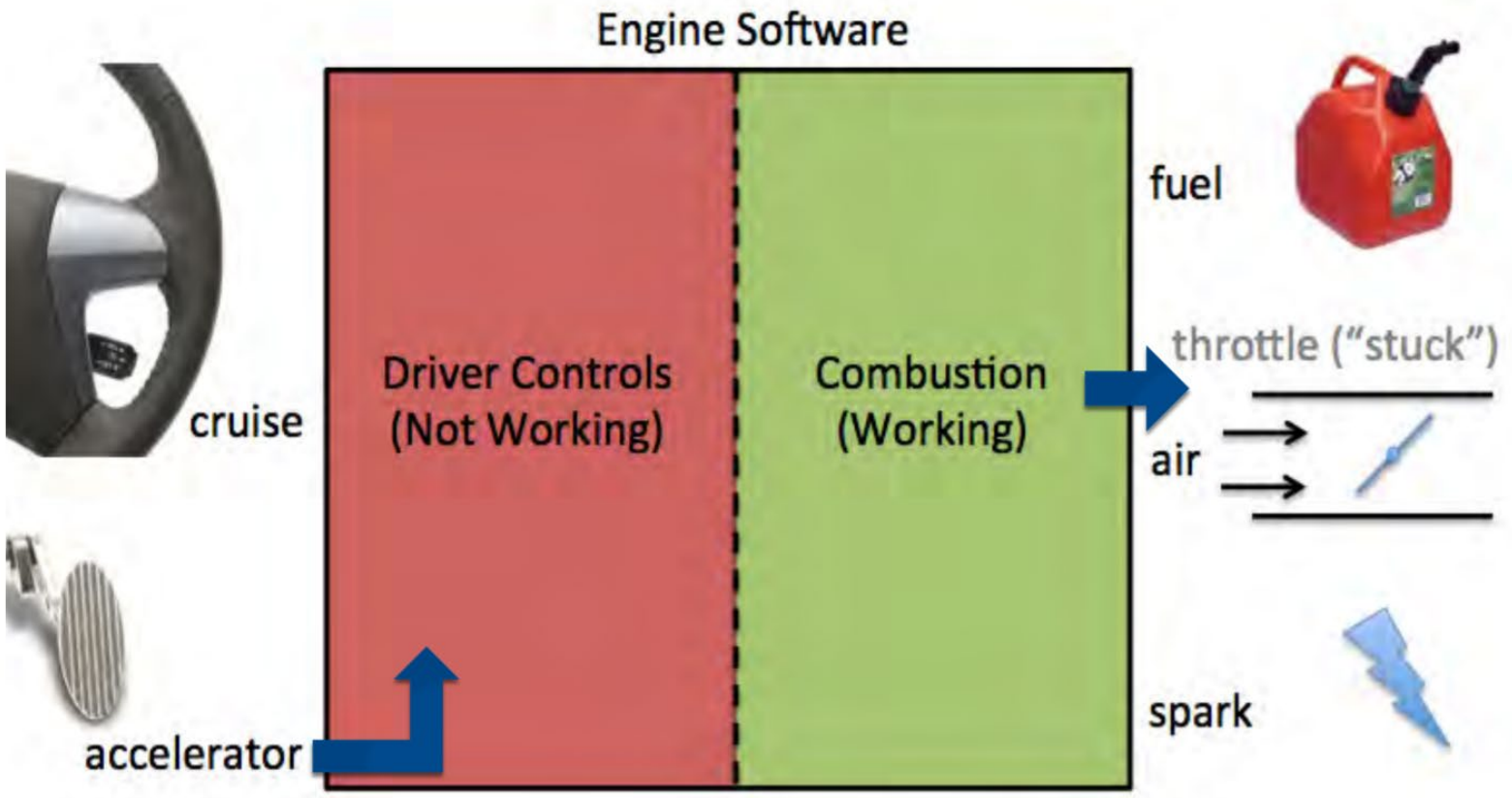
- *NASA, Appendix A, p. 118*



NASA, p. 13

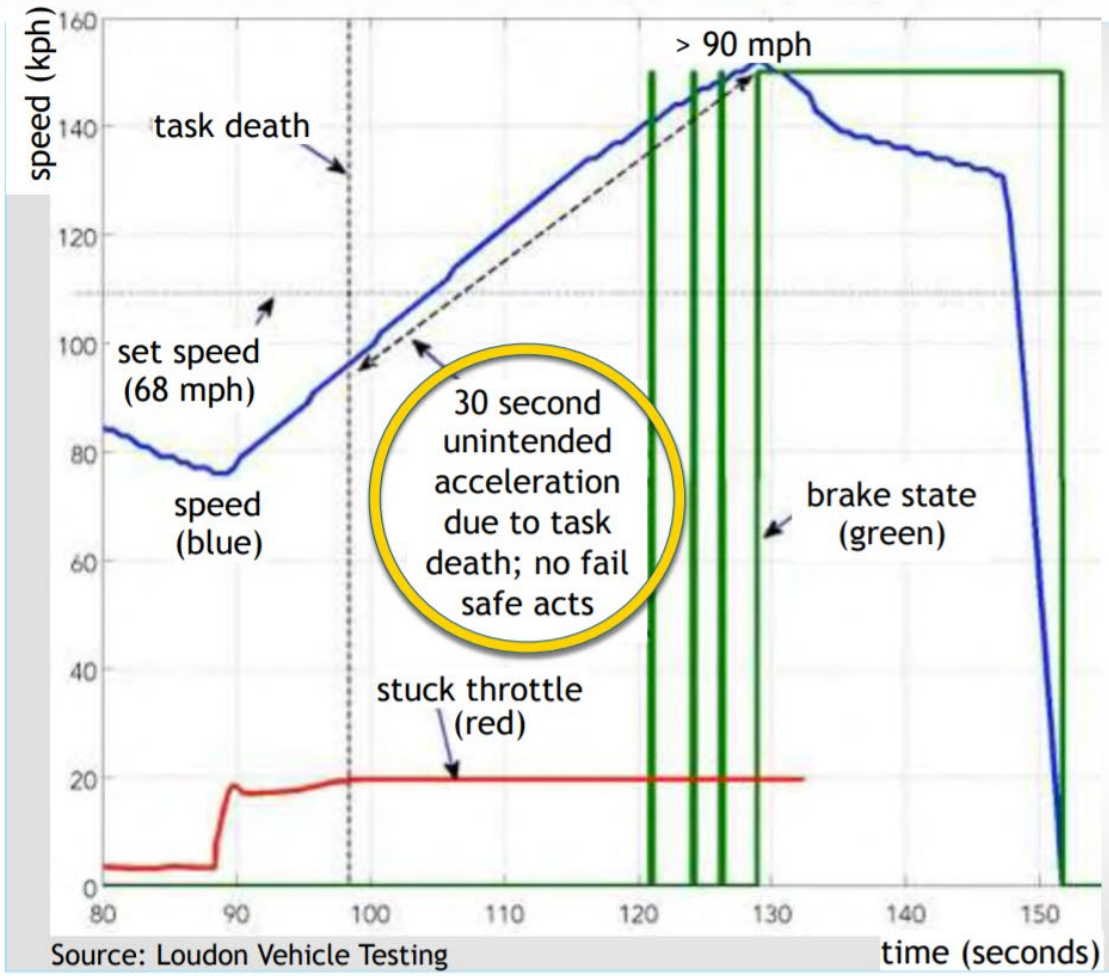
More Details: https://www.safetyresearch.net/Library/BarrSlides_FINAL_SCRUBBED.pdf

Example of Two Tasks – One Dies?



Testing…

EXAMPLE OF UNINTENDED ACCELERATION



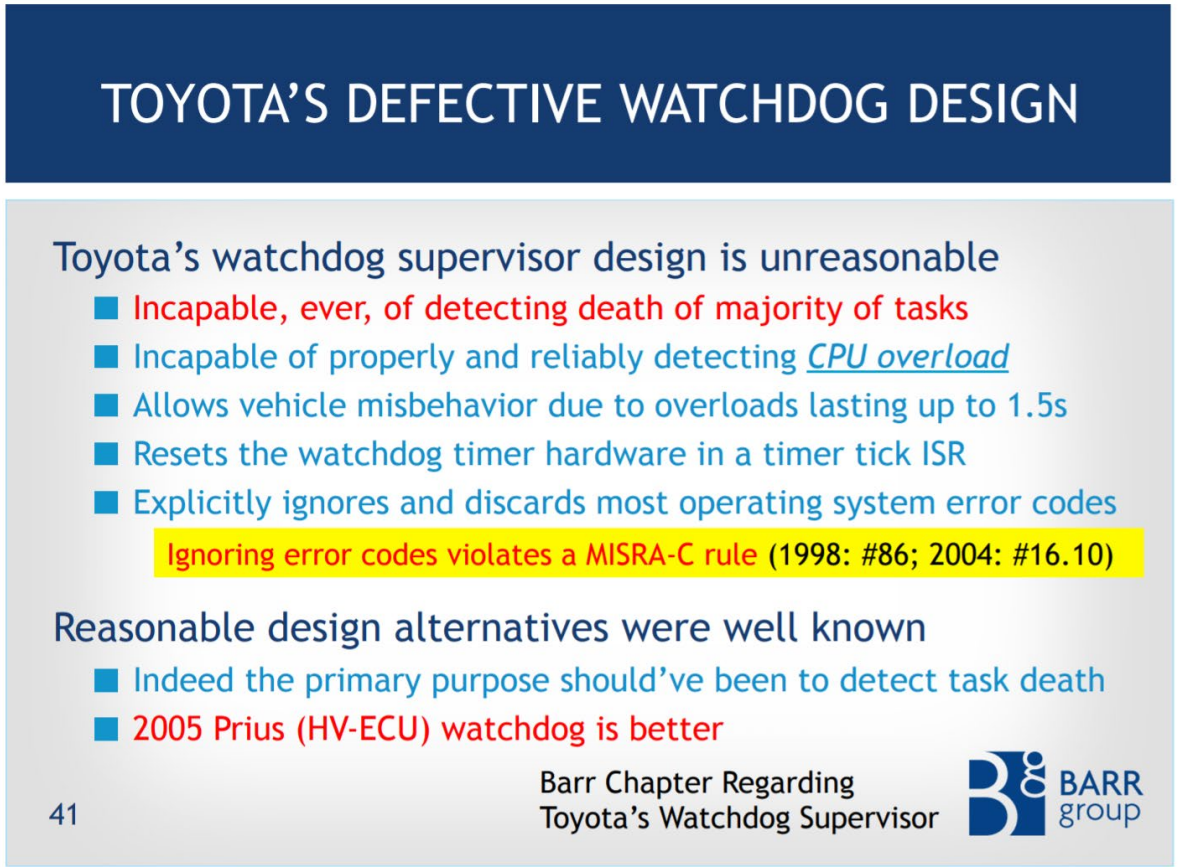
Source: Loudon Vehicle Testing

- Representative of task death in real-world
- Dead task also monitors accelerator pedal, so loss of throttle control
  - ✓ Confirmed in tests
- When this task's death begins with brake press (any amount), driver must fully remove foot from brake to end UA
  - ✓ Confirmed in tests

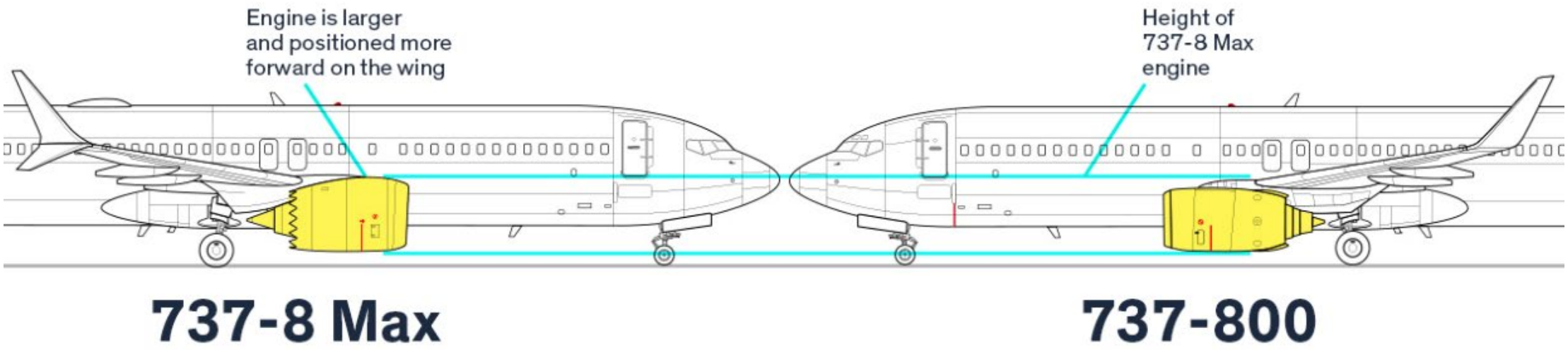
BARR group

Huge number of failures…

- See Barr's presentation (linked earlier) for list of… many many things like:
- Code which has complex connections and cannot be tested.
- Fail-safes that don't fail.
- Unsafe memory usage.

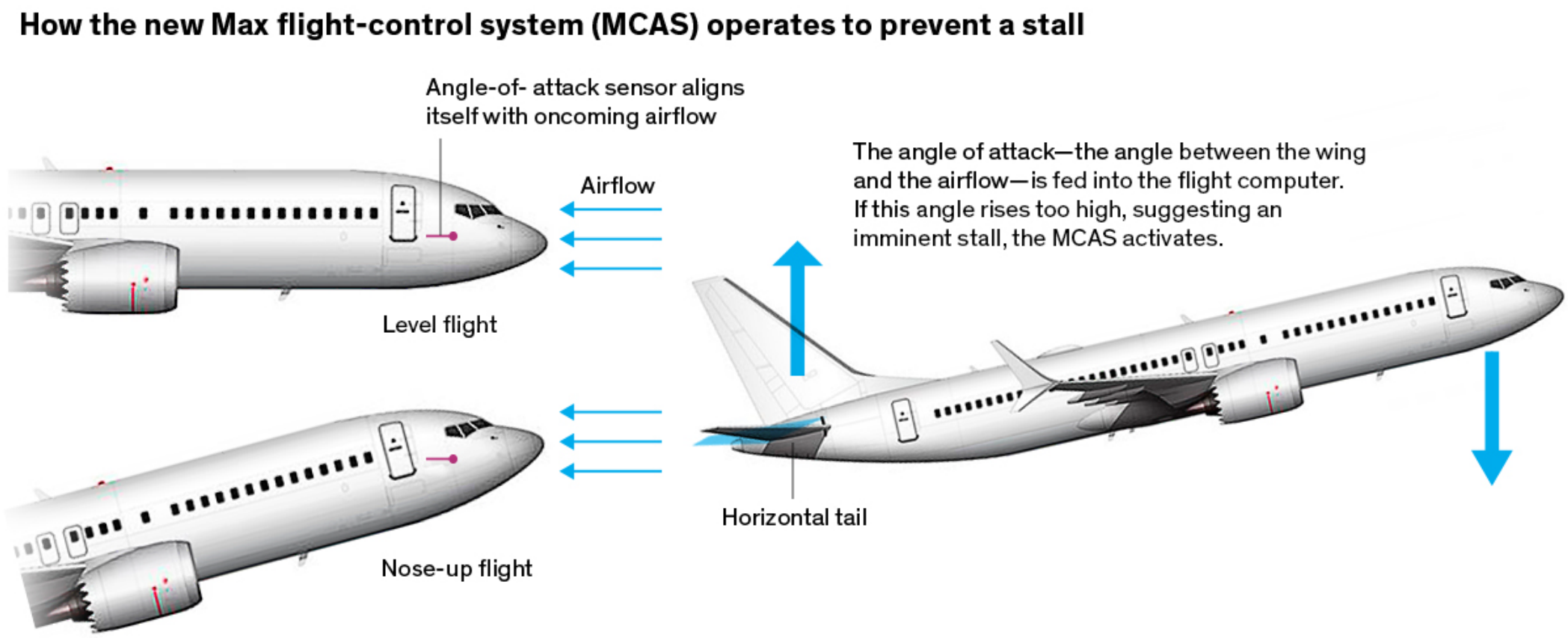


Failure #3 – 737 Max 8

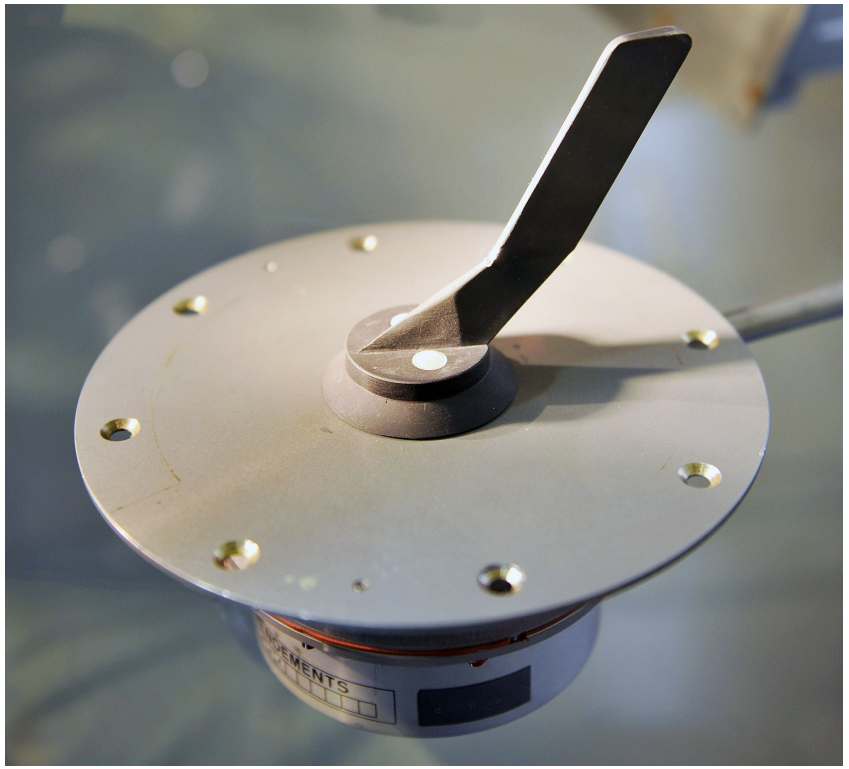


https://spectrum.ieee.org/aerospace/aviation/how-the-boeing-737-max-disaster-looks-to-a-software-developer

Fixing the “Engines Now Unstable” Problem… in software!



Angle of Attack Sensors…



x1 used by MCAS??

- Aircraft actually have multiple – but possibly they are not used in a redundant manner it appears.

https://www.seattletimes.com/business/boeing-aerospace/a-lack-of-redundancies-on-737-max-system-has-baffled-even-those-who-worked-on-the-jet/

https://en.wikipedia.org/wiki/Maneuvering_Characteristics_Augmentation_System

More Failures you can Research:

- Ariane 5 rocket launch (https://www.bugsnag.com/blog/bug-day-ariane-5-disaster).
  - $370m cost (rocket go boom) - https://www.youtube.com/watch?v=N6PWATvLQCY&t=1m10s .
  - Cause – incorrect conversion of types (64 bit converted to 16-bit).
- Mars Climate Orbiter (https://en.wikipedia.org/wiki/Mars_Climate_Orbiter)
  - $330m cost (orbiter lost).
  - Cause – incorrect internal unit conversion between systems.

Duty as an Engineer

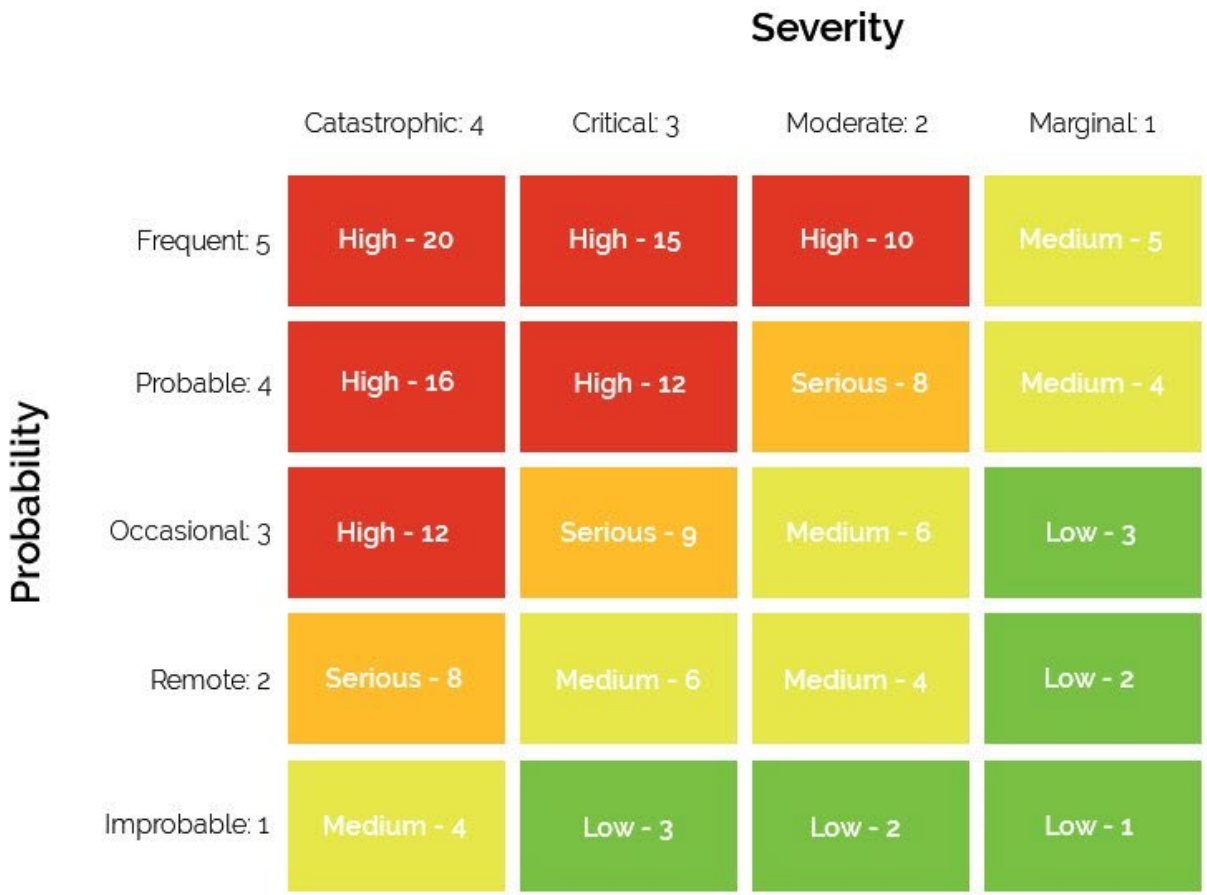
- Safety of public is <u>always the most important thing</u> as an engineer.
- Previous examples all caused damage to public, all with various root causes:
  - Engineers who were not familiar with best practices.
  - Management pressure.
  - Engineers may be “blind” to how final system is used, keep head down to avoid trouble.

Designing Safe Computer Systems

- This course isn't about computer safety – at least another course in itself!

- But we can fairly easily apply the basics of computer safety here.

Generic Risk Matrix Thingy



IEC 61508 – Functional Safety

Table 1: Defining categories of likelihood of occurrence

Category	Definition	Range (failures per year)
Frequent	Many times in system lifetime	$> 10^{-3}$
Probable	Several times in system lifetime	10^{-3} to 10^{-4}
Occasional	Once in system lifetime	10^{-4} to 10^{-5}
Remote	Unlikely in system lifetime	10^{-5} to 10^{-6}
Improbable	Very unlikely to occur	10^{-6} to 10^{-7}
Incredible	Cannot believe that it could occur	$< 10^{-7}$

Table 2: Defining consequence categories

Category	Definition
Catastrophic	Multiple loss of life
Critical	Loss of a single life
Marginal	Major injuries to one or more persons
Negligible	Minor injuries at worst

IEC 61508 – Functional Safety

Table 4: A risk class matrix

	CONSEQUENCE			
LIKELIHOOD	**Catastrophic**	**Critical**	**Marginal**	**Negligible**
Frequent	I	I	I	II
Probable	I	I	II	III
Occasional	I	II	III	III
Remote	II	III	III	IV
Improbable	III	III	IV	IV
Incredible	IV	IV	IV	IV

Where:

- Class I: Unacceptable in any circumstance;
- Class II: Undesirable: tolerable only if risk reduction is impracticable or if the costs are grossly disproportionate to the improvement gained;
- Class III: Tolerable if the cost of risk reduction would exceed the improvement;
- Class IV: Acceptable as it stands, though it may need to be monitored.

Automotive & Other Standards

- Many industries add their own standards, such as automotive which has ISO 26262.
- MISRA project has rules known as “MISRA C” which specify certain things you can do to improve your C code.

NOTE: Colin called this ‘MIRSA C’ incorrectly in early revision of this.

Example of MISRA C Code Rule

Rule 59 (required): The statement forming the body of an "if", "else if", "else", "while", "do ... while", or "for" statement shall always be enclosed in braces

Basically, this says that from now you must clean up your act, you can't write sloppy things like the `else` clause in following example.

```
if (x == 0)
{
y = 10;
z = 0;
}
else
y = 20;
```

The idea of this rule is to avoid a classical mistake. In the example below the line `z = 1;` was added. It looks as though it's part of the `else` clause but it's not! In fact, it is placed after the `if` statement altogether, which means that the assignment will always take place.

If the original `else` clause would have contained the braces from the beginning this problem would never have occurred.

```
if (x == 0)
{
y = 10;
z = 0;
}
else
y = 20;
z = 1;
```

Problems with These “Standards”

- How frequent are the errors in real life?
  - Therac-25 designers assumed it was user error.
  - Considerable $$$ spent on PR campaigns to assure everyone that Toyota unintended acceleration was a myth & due to users pressing wrong pedal (whether true or not – code in ECUs was horribly bad and should not have been allowed).
  - Errors that happen 1/100000 may happen a lot if you have 100000 units in the field…
- What about hardware problems?

Aggressive Testing & Fuzzing

- Code testing critical to truly catch errors.
  - But how good are your test? Do you just test correct & “expected incorrect” inputs?

- We can use code fuzzing to help catch more errors.
  - Code fuzzing sends in incorrect data.
  - More on that in the next topic…