Sandia National Laboratories

Exceptional service in the national interest

Verification of Digital Numerics for High Consequence Systems

Samuel D. Pollard, Digital Foundations & Mathematics

proof.sandia.gov

FPTalks

Virtual

11 July 2024, 11:00

SAND2024-07142C

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

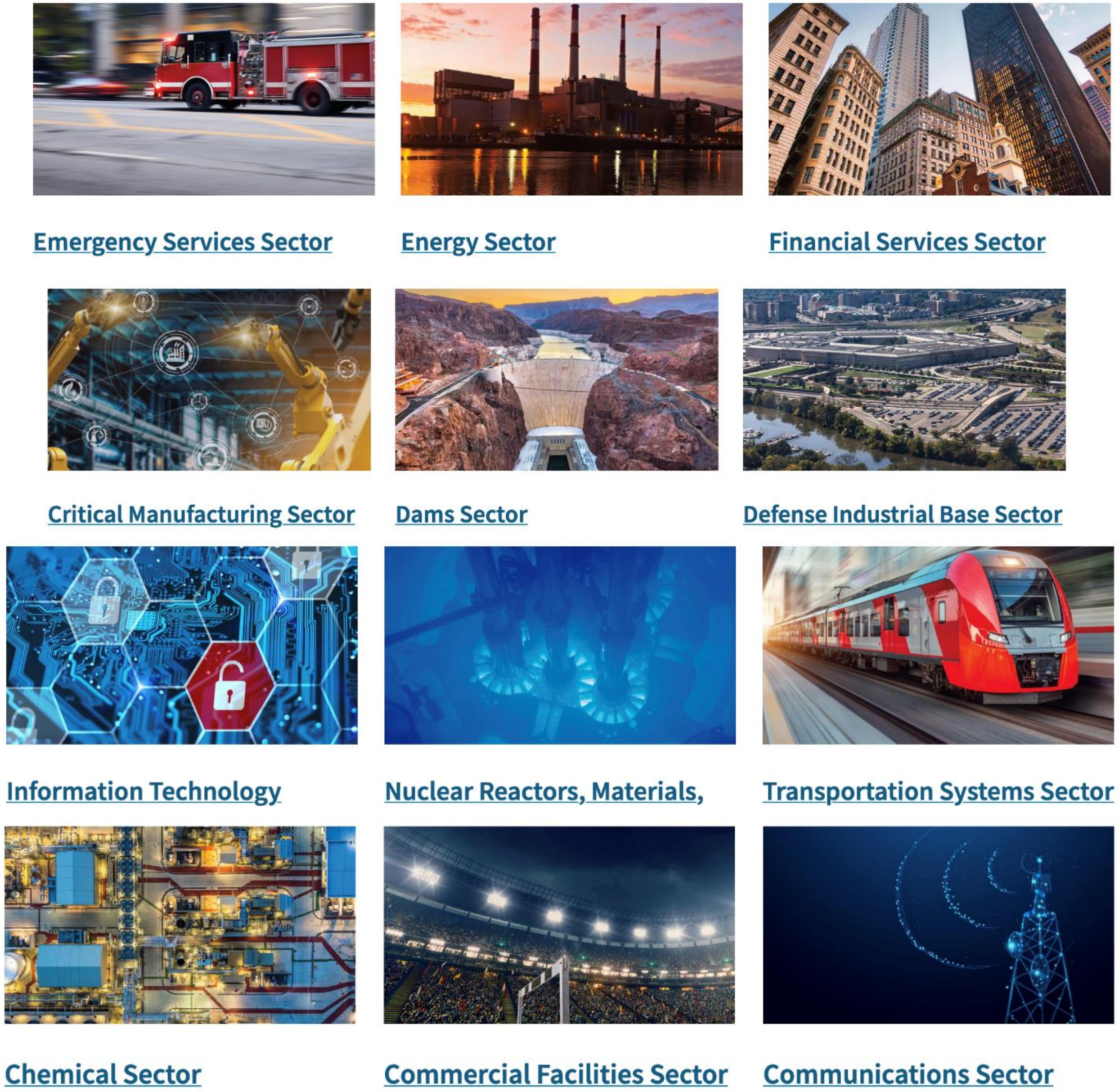
Overview

- Sandia National Labs is a US government research & development center
- Sandia develops software for high-consequence embedded control systems

Livermore, California site

Overview

- The systems are relatively simple
- The cost for error is very high
- Requirements relatively complex
- A good use case for formal methods



Emergency Services Sector

Energy Sector

Financial Services Sector

Critical Manufacturing Sector

Dams Sector

Defense Industrial Base Sector

Information Technology

Nuclear Reactors, Materials,

Transportation Systems Sector

Chemical Sector

Commercial Facilities Sector

Communications Sector

https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience/critical-infrastructure-sectors

Acknowledgments

- We gathered a small group
  - Jarom Christiansen
  - Anthony Dario
  - Ariel Kellison
  - TJ Machado

Jarom Christiansen

Anthony Dario

TJ Machado

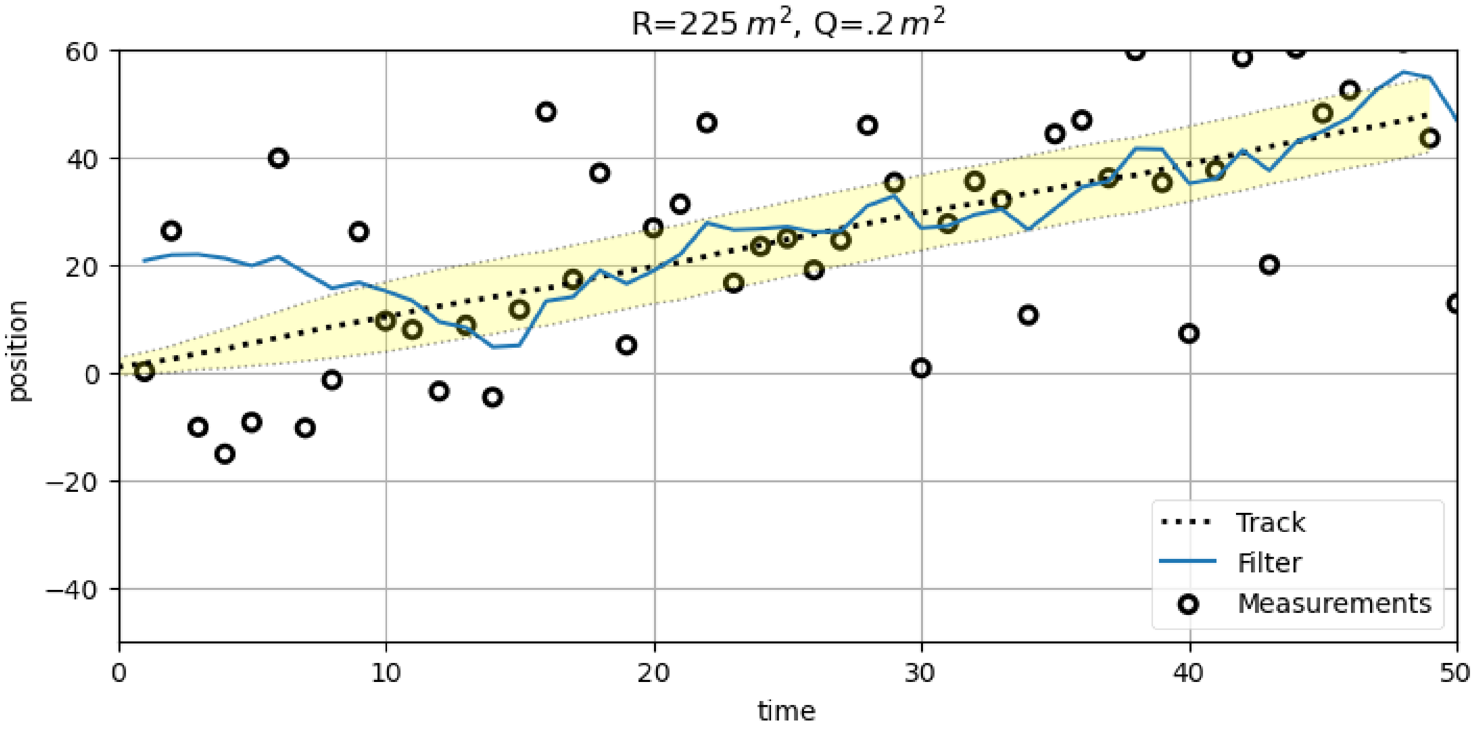
Ariel Kellison

Several Numerics projects

1. Verified Kalman Filter
2. Improving floating-point support with Frama-C
3. A secret third thing (currently under peer review!)

Verified Kalman Filter

- Verify an EKF in C
- Properties to verify
  1. memory safety
  2. numerics
  3. concurrency/ scheduling
- (1) good for Frama-C
- less so (2) and (3)

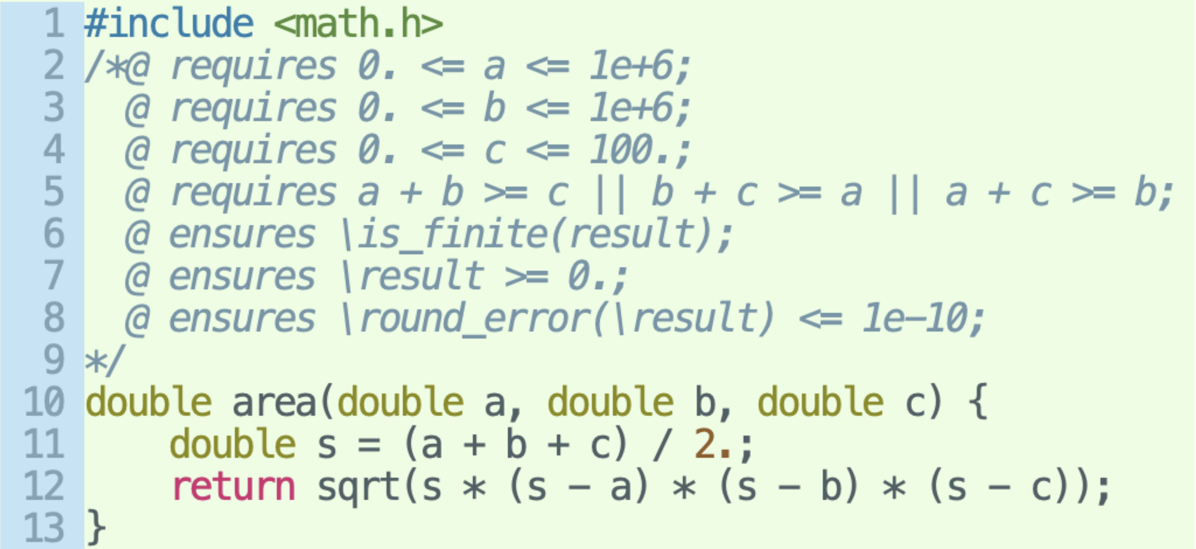


Verified Kalman Filter: Building complexity

1. Start with simple examples
   - 1D up to 3D Kalman filters
   - These may make good FPBench examples, thoughts?
   - VST proofs for some, but
   - Full Kalman filter requires LU-decomposition
2. For the real codebase, VST is not feasible
   - Build up Frama-C annotations
   - Floating-point keeps causing hang-ups
   - 2 models of numerics in Frama-C: float & real

Improving Floating-Point Support for Frama-C

- In theory
  - Frama-C in theory supports numerics via gappa
- In practice
  - most C constructs not supported
- Goal
  - Add support for FPTaylor
  - Translate code to support analysis (e.g., unroll loops)
- Challenge
  - ACSL + C + Frama-C are complex
  - Likely require modifying WP



```
#include <math.h>
/∗@ requires 0. <= a <= 1e+6;
  @ requires 0. <= b <= 1e+6;
  @ requires 0. <= c <= 100.;
  @ requires a + b >= c || b + c >= a || a + c >= b;
  @ ensures \is_finite(result);
  @ ensures \result >= 0.;
  @ ensures \round_error(\result) <= 1e−10;
∗/
double area(double a, double b, double c) {
    double s = (a + b + c) / 2.;
    return sqrt(s ∗ (s − a) ∗ (s − b) ∗ (s − c));
}
```