A Simplex Architecture for Intelligent and Safe Unmanned Aerial Vehicles

Prasanth Vivekanandan[+], Gonzalo Garcia[*], Heechul Yun[+], Shawn Keshmiri[*]

Electrical Engineering and Computer Science [+]

Aerospace Engineering [*]

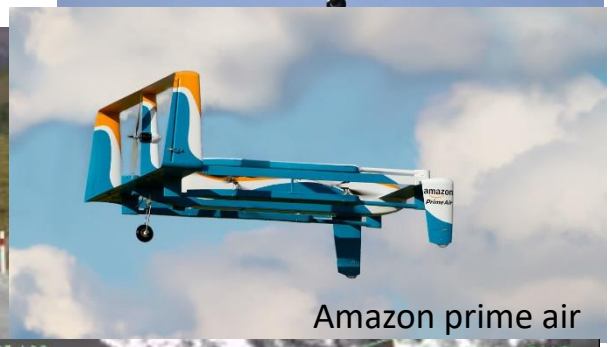
University of Kansas

KU THE UNIVERSITY OF KANSAS

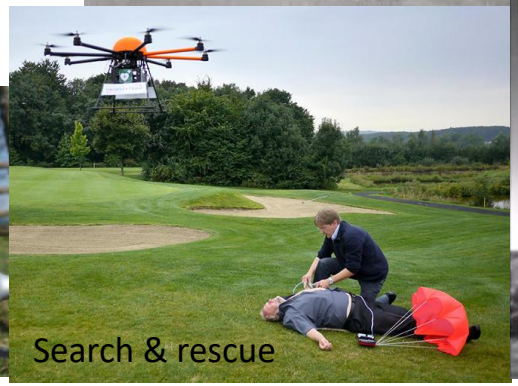
Intelligent UAVs

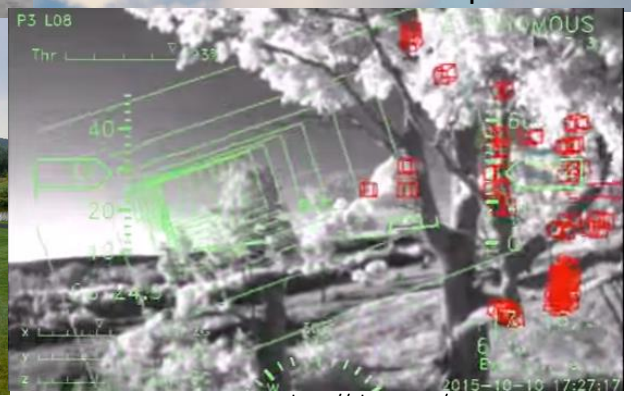
- Many applications
  - Commercial, military, police,…
  - $10B in 3 years*


Follow me


Amazon prime air

surveillance


Search & rescue

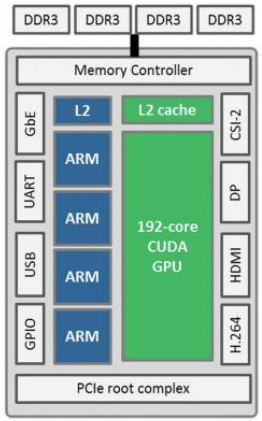


http://abarry.org/

(*) http://gizmodo.com/some-good-things-drones-can-actually-do-1475717696

Intelligent UAVs

- Powerful computer hardware
  - Multicore SoC, GPU
    - High performance, Low cost, size, weight, and power



- Powerful software framework
  - Linux, middleware, libraries
    - Productivity, ease of development
  - Like a PC

NVIDIA CUDA

ROS

Open Source Robotics Foundation

Safety Challenges

http://www.nytimes.com/2015/01/28/us/white-house-drone.html

http://petapixel.com/2015/12/23/crashing-camera-drone-narrowly-misses-top-skiier/

Domestic drone accidents

http://rochester.nydatabases.com/map/domestic-drone-accidents

The use of unmanned aerial vehicles (also known as UAVs or drones) has been growing dramatically, but the practice has not been without risks. Several dozen drone crashes have been reported within the United States. The database below contains information on all of the known incidents.

DATE	OPERATOR	UAV TYPE	DESCRIPTION
6/7/2016	US Air Force	MQ-9 Reaper	An Air Force Reaper drone crashed at Nevada Test and Training Range northwest of Las Vegas, Nevada. The aircraft was being controlled from Creech AFB. (Location is approximate.)
5/5/2016	Private citizen	Unknown small UAV	A small UAV slammed into and broke a dining room window in a home in the Capitol Hill section of Seattle, Washington.
4/13/2016	Private citizen	Unknown small UAV	A man fell into the Black River in Port Huron, Michigan after the small UAV he was flying crashed into a riverside light pole. He had to be rescued by firefighters.

UAVs are safety critical systems

KU THE UNIVERSITY OF KANSAS

Sources of Failures

- Sensors

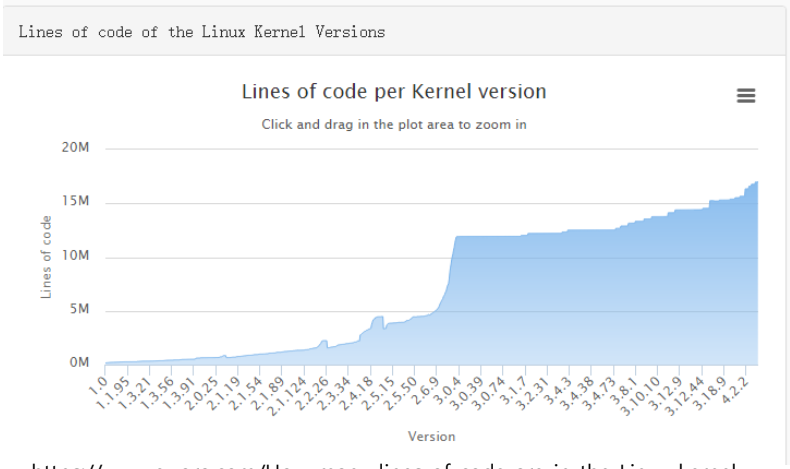
- Airframe

- Actuators

- **Onboard computing platform**
  - **Software**
  - **Hardware**

Safety Challenges: Software

- Increasing complexity
  - E.g., Linux: > 15M SLOC
- Concurrency
  - Multithreading is hard
    - Race condition. Order violation
- Timing unpredictability
  - Shared resource contention affects timing
    - >21X slowdown on a cache partitioned multicore (*)



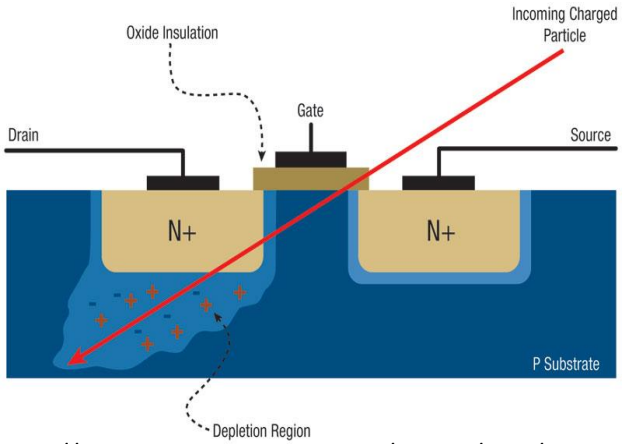
https://www.quora.com/How-many-lines-of-code-are-in-the-Linux-kernel

➔ Software bugs are hard to weed out

(*) Prathap Kumar Valsan, Heechul Yun, Farzad Farshchi. Taming Non-blocking Caches to Improve Isolation in Multicore Real-Time Systems. *IEEE Intl. Conference on Real-Time and Embedded Technology and Applications Symposium (RTAS)*, IEEE, 2016.. ***Best Paper Award***

Safety Challenges: Hardware

- Hardware bugs
  - Pentium floating point bug (FDIV bug)
  - Intel CPU bugs in 2015: http://danluu.com/cpu-bugs/
    - "Certain Combinations of AVX Instructions May Cause Unpredictable System Behavior"
    - "Processor May Experience a Spurious LLC-Related Machine Check During Periods of High Activity"
    - …
- Transient hardware faults (soft errors)
  - Single event upset (SEU) in SRAM, logic
    - Due to alpha particle, cosmic radiation
  - Manifested as software failures
    - Crashes, wrong output: silent data corruption
  - Bigger problem in advanced CPU
    - Increased density, freq → higher soft error



http://www.cotsjournalonline.com/articles/view/102279

Safety Challenges: Hardware

- SRAM SER vs. technology scaling
  - Per-bit SER decreases
  - Per-chip SER increases (due to higher density)

Design rule	SER (A.U)		MCUratio	MCU maximum size	Maximum bit multiplicity
nm	per device	per Mbit	%	bit	bit
130	1	1	5.8	182	10
90	1.9	0.94	13.5	2790	15
65	3.1	0.77	18.2	110860	19
45	4.3	0.53	26.4	118665	42
32	5.8	0.36	37	1933244	53
22	6.7	0.21	42.6	1075296	174

Ibe et al., "Scaling Effects on Neutron-Induced Soft Error in SRAMs Down to 22nm Process" (Hitachi)

➔ Complex hardware is buggy and less reliable

How to Improve Safety of a System?

- Correct by design
  - Formal method based software development
    - Difficult for a complex system
  - Radiation hardened processors
    - Expensive and low performance

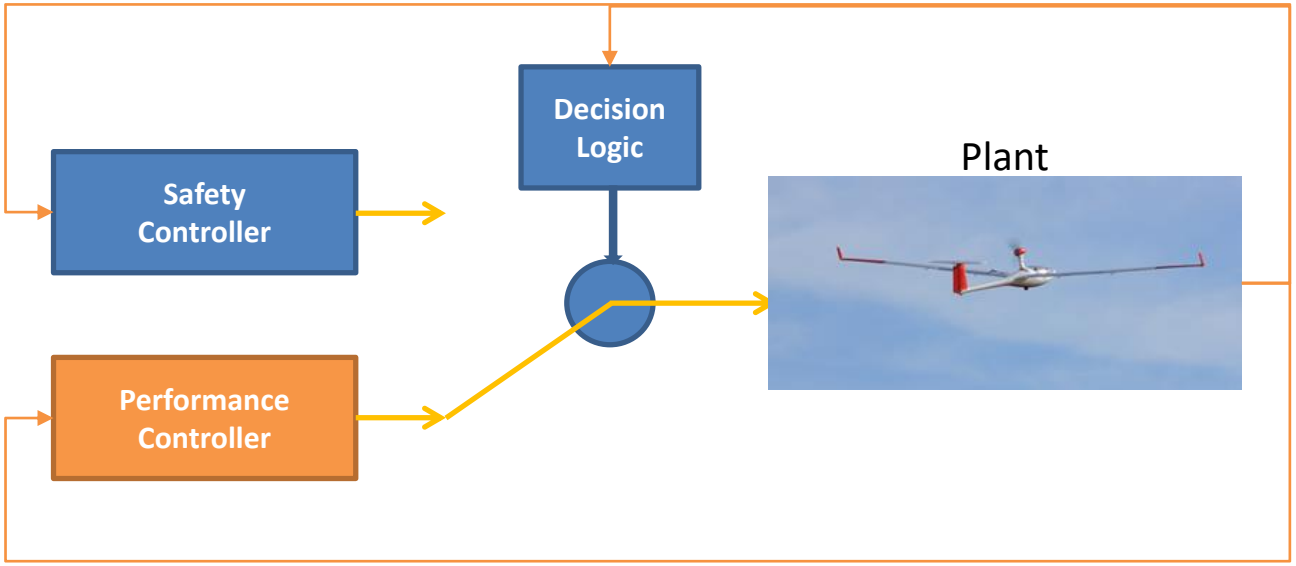
- **Deal with failures**
  - **Run-time monitoring and redundancy**

Outline

- Motivation
- **UAV Simplex Architecture**
- Prototype and Case Study

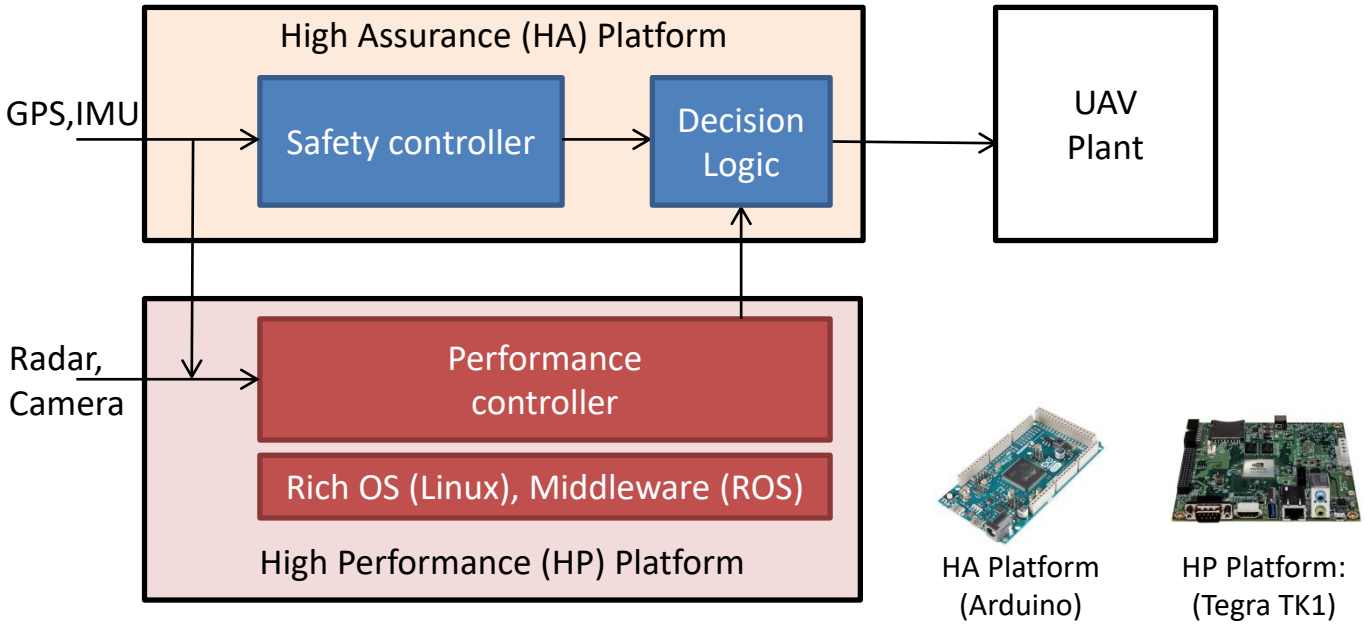
Simplex Architecture (*)

- Protect an untrusted complex controller with a trusted backup controller
- General architectural principal



(*) L. Sha, Using Simplicity to Control Complexity, IEEE Software, 2001

UAV Simplex Architecture



- Idea: use two hardware/software platforms with distinct performance and reliability characteristics to realize Simplex

Two Platforms

Platform	High-Assurance (HA)	High-Performance (HP)
Hardware	SEU resistant	SEU susceptible
Software	Verifiable	Unverifiable

- High Assurance (HA) Platform
  - Simple hardware and software for **verification** and **reliability**
  - Hardware: low frequency and density to reduce SEUs
  - Software: certifiable, simple, low SLOC

- High Performance (HP) Platform
  - Complex hardware and software for **performance**
  - Hardware: performance oriented multicore, multi-gigahz, gpu
  - Software: productivity oriented software framework, millions SLOC

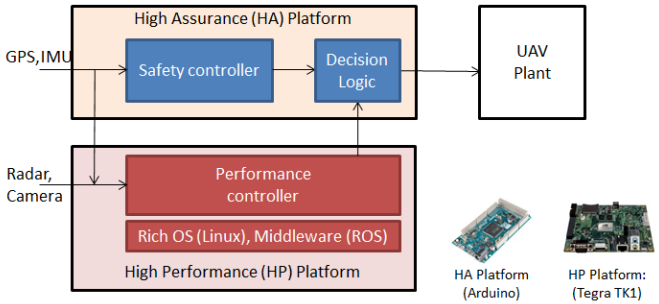
KU THE UNIVERSITY OF KANSAS

Outline

- Motivation
- UAV Simplex Architecture
- **Prototypes and Case Study**

Prototype Avionics

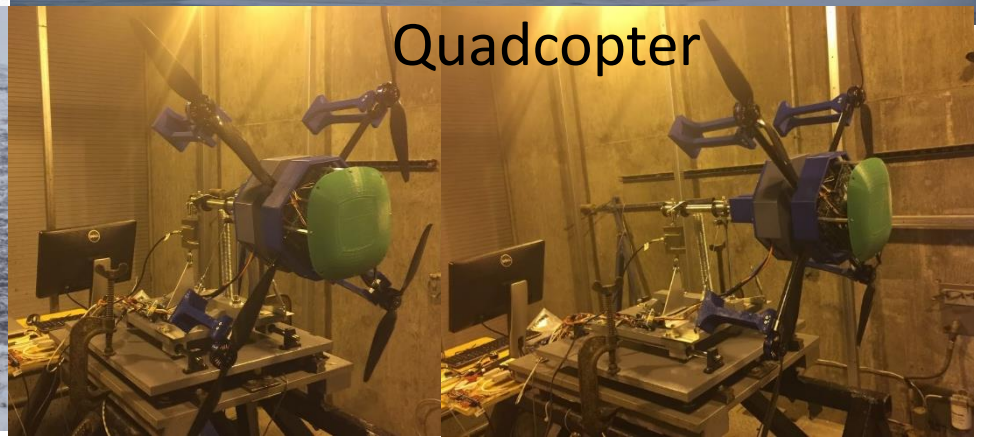
- AFS: our custom built avionics
  - Arduino based custom DAQ
    - Basic sensors: IMU, GPS
  - Nvidia Tegra TK1
    - 4 x ARM cores + 192 GPU cores
    - Advanced sensors: camera, radar



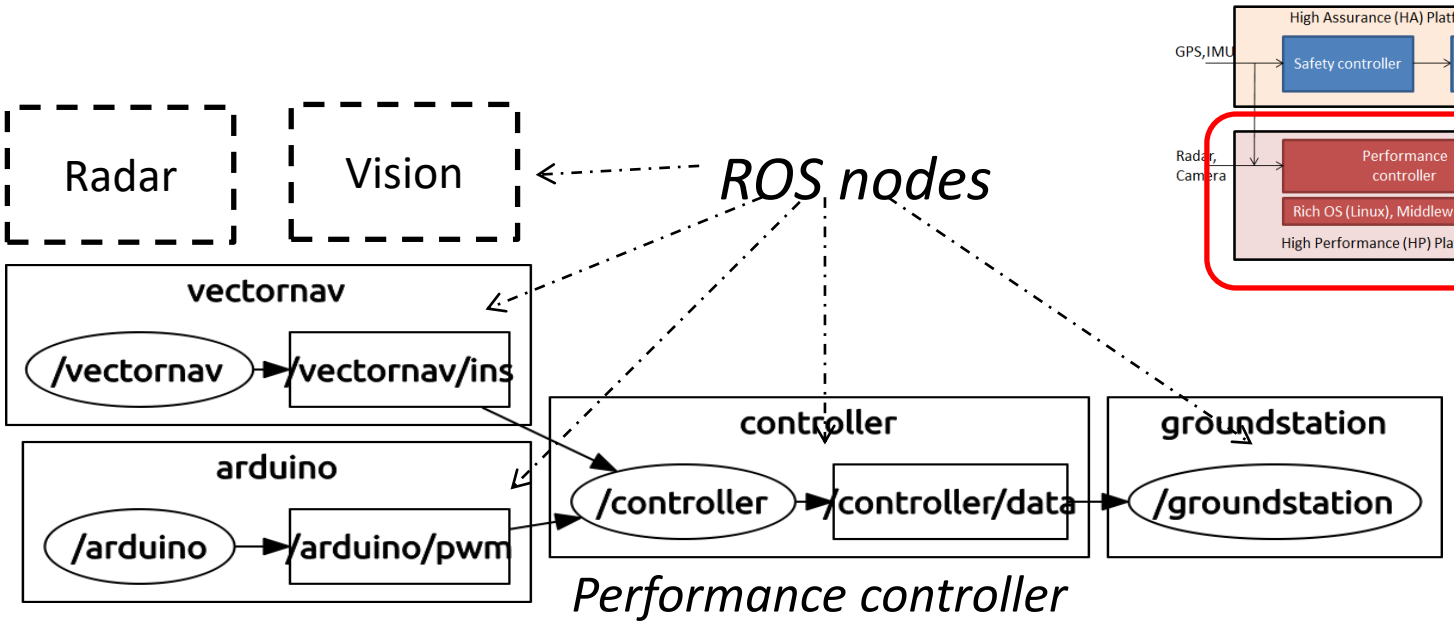
- UAVs with the AFS
  - Applied to four UAVs in Dr. Keshmiri's lab in KU Aerospace Engineering
  - Fixed wing (DG 808, G1XD, G1XB) and a Quadcopter

KU THE UNIVERSITY OF KANSAS

UAVs with AFS

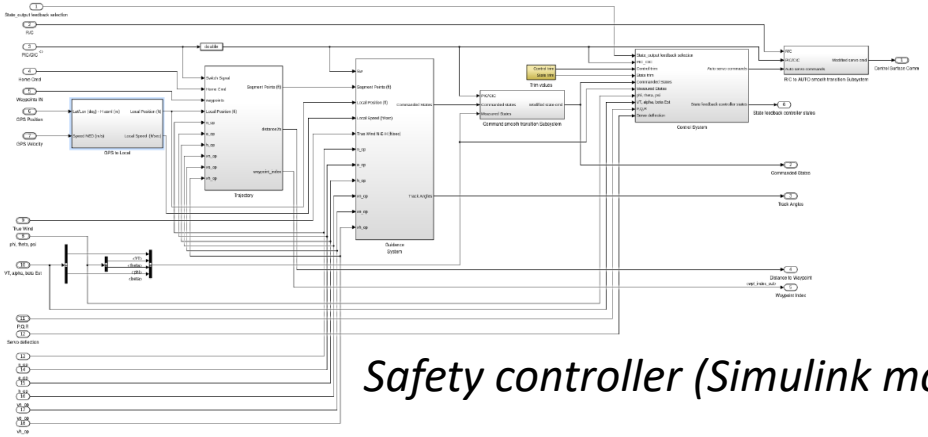


Performance Controller

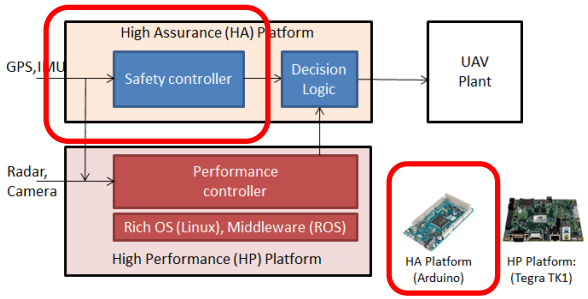


- Hardware
  - Nvidia Tegra TK1, 4 x ARM Cortex-A15 @ 2.3GHz, 192 core GPU
  - 28nm process, > a billion transistors → complex, high potential SEUs
- Performance controller
  - Intelligent adaptive non-linear control using advanced sensor packages (goal)
  - Use Linux (Ubuntu), Robot Operating System (ROS) → difficult to verify

Safety Controller



Safety controller (Simulink model)

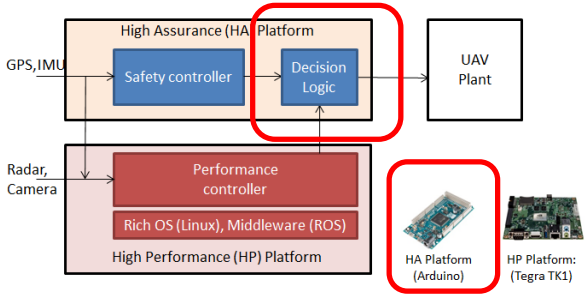


- Hardware
  - Arduino Due, a single ARM Cortex-M3 @ 80MHz
  - Low density, low operating freq. → less susceptible for SEUs
- Safety controller
  - Matlab Simulink coder + Arduino sketch, no OS →small and easier to verify

Decision Logic

Observed behaviors	Faults in the HP platform
No output	OS/controller crash
Delayed output	Deadline miss
Unsafe output	Bugs, SEUs, bad controller design, etc.

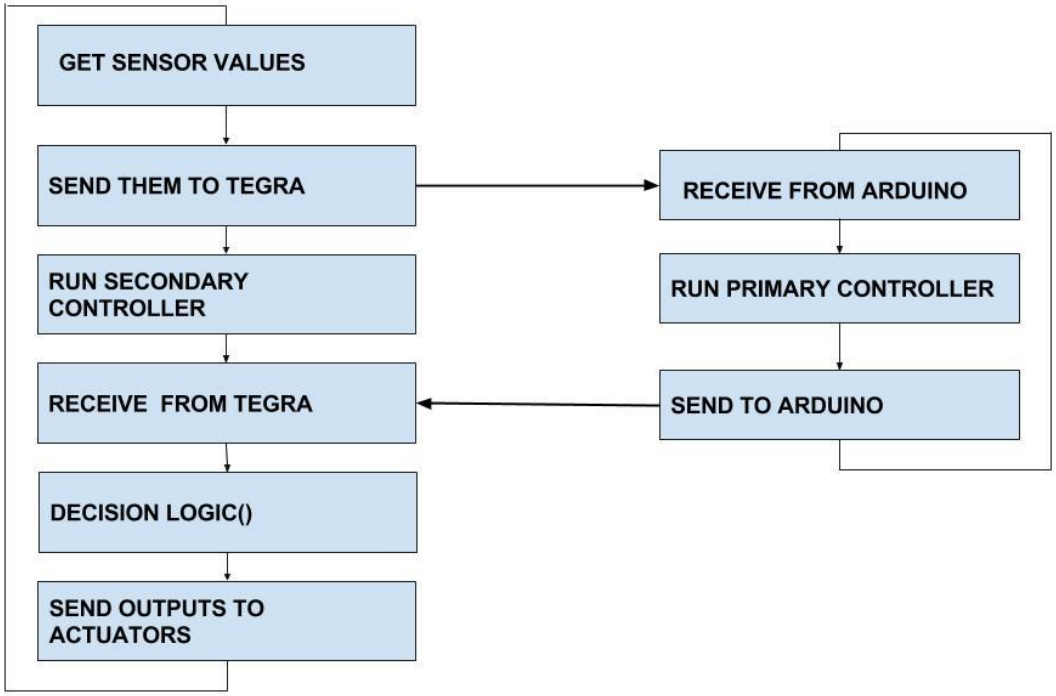
Detectable faults



- Fault models
  - HA (safety controller, decision logic) is trusted
  - HP is not trusted
- Decision logic
  - Detect crash, connect failure, timing violation, invalid outputs (e.g., NaN)
  - Recovery: reboot the HP platform
  - *Limitation*: Currently don't know "unsafe" states

KU THE UNIVERSITY OF KANSAS

Execution Flow



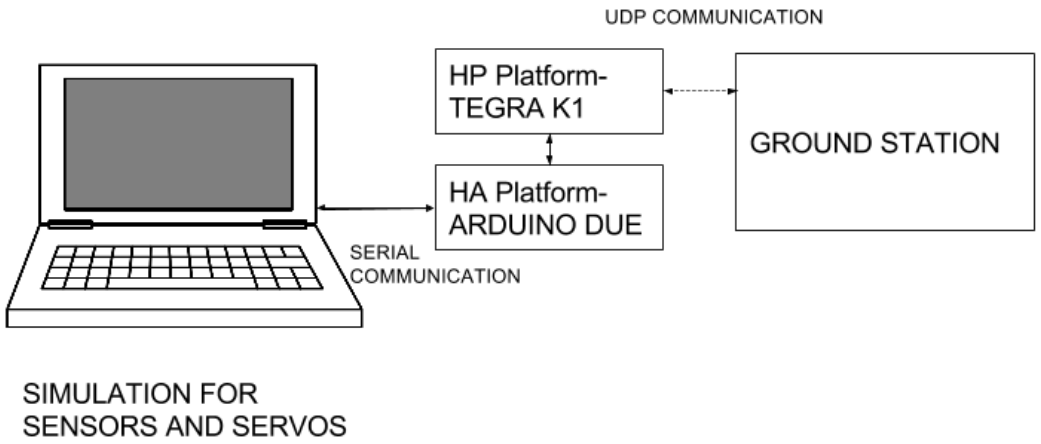
HA platform
(Arduino)

HP platform
(Tegra TK1)

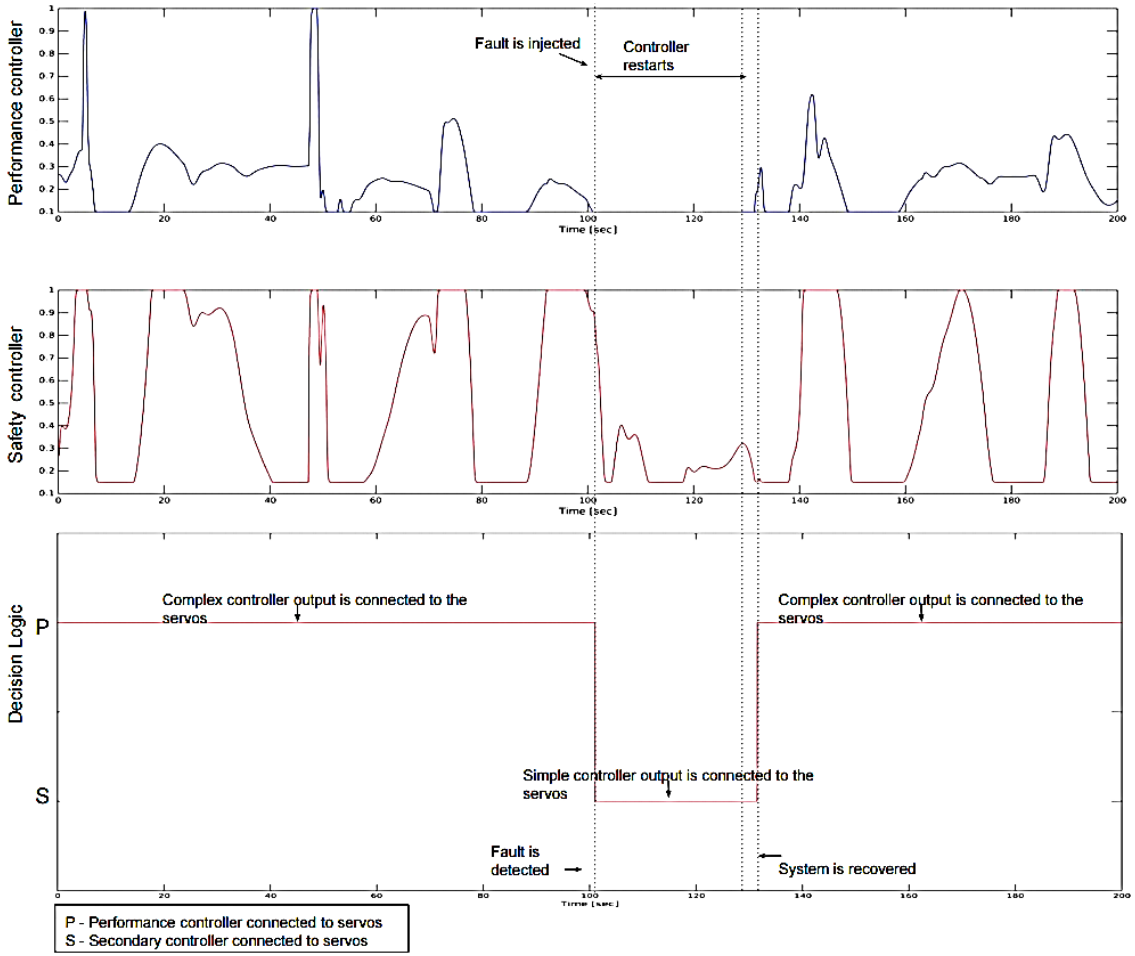
Case Study: Fault (Crash) Injection

- Experiment
  - Kill the performance controller in the middle flight

- Hardware-in-the-loop (HIL) setup



Case Study: Fault (Crash) Injection



Case Study: Fault (Crash) Injection

- Monitored from the ground station software

Conclusion and Future Work

- Safety challenges of intelligent UAVs
  - Software: increasing complexity, concurrency and timing non-determinism
  - Hardware: increasing reliability issues. E.g., transient hardware faults (SEUs)

- UAV Simplex architecture
  - Two platform based realization of Simplex
    - High assurance (HA) platform: simple, verifiable
    - High performance (HP) platform: performant, unverifiable

Conclusion and Future Work

- Prototype development and case study
  - Nvidia Tegra TK1 + Arduino based
  - Can survive from performance controller crash
- Ongoing and Future work
  - Radar and vision based sense & avoid
  - Define and detect *unsafe state* (not just crash)
  - Detect and recover intrusion (security)
  - Handling of sensor faults

KU THE UNIVERSITY OF KANSAS

Thank You

Disclaimer:
This work is supported by the National Aeronautics and Space Administration's (NASA's) Leading Edge Aeronautics Research for NASA (LEARN) fund under grant number NNX15AN94A and Paul G. Allen Family Foundation (PGAFF) grant number KUAE#40956.

More details can be found in the following publication.

Prasanth Vivekanandan, Gonzalo Garcia, Heechul Yun, Shawn Keshmiri. "A Simplex Architecture for Intelligent and Safe Unmanned Aerial Vehicles." *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA),* IEEE, 2016

KU THE UNIVERSITY OF KANSAS