A Payload Verification and Management Framework for Small UAV-based Personal Remote Sensing Systems

Cal Coopmans, Presenter
cal.coopmans@usu.edu
Christopher M. Coffin
christopher.m.coffin@aggiemail.usu.edu
Utah State University
Center for Self-Organizing and Intelligent Systems

Brandon Stark
brandon.stark@gmail.com
U.C. Merced
MESA Lab

Co-Robotics and Tele-Presence Session
5th International Symposium on Resilient Control Systems
Salt Lake City, UT
Wednesday, August 15 2012

Outline

- Introduction
- Data Mission Assurance
- AggieAir Architecture
- PRS Payload Requirements
- ISaAC
- AggieCap Implementation
- Testing and Verification
- Example payload and results
- Conclusions and future work

Related Publications

1. Calvin Coopmans, "*Architecture, Inertial Navigation, and Payload Designs for Low-Cost Unmanned Aerial Vehicle-Based Personal Remote Sensing*," MSc. Thesis, Utah State University, Logan, UT 2010.
2. Calvin Coopmans, Brandon Stark, Christopher M Coffin, "*A Payload Verification and Management Framework for Small UAV-based Personal Remote Sensing Systems*," (future) In proc. of ISRCS 2012, Salt Lake City, UT.
3. Brandon Stark, Calvin Coopmans, YangQuan Chen, "*Concept of Operations for Personal Remote Sensing Unmanned Aerial Systems*," In proc. of ICUAS 2012, Philadelphia, PA.
4. Brandon Stark, Calvin Coopmans, YangQuan Chen, "*A Framework for Analyzing Human Factors in Unmanned Aerial Systems*," (future) In proc. of ISRCS 2012, Salt Lake City, UT.
5. Christophe Tricaud and YangQuan Chen. *Optimal Mobile Sensing and Actuation Policies in Cyber- Physical Systems*. Springer, Dec. 2011, 170p, ISBN: 978-1-4471-2261-6.

Introduction

- After years of Small UAV payload development experience, CSOIS has created a payload management system for small UASs
- We have adapted many sensors, including Visible light cameras, Near-Infrared, and yes, Thermal Infrared sensors
- Other sensors and actuators have been included for many different mission configurations

Titan airframe and Launcher

Titan Landing

Introduction

- The specific class of UAS is “Personal Remote Sensing” (PRS) [1] systems, typically 50 airborne-pounds or less
- These PRS missions vary widely
  - Aerial Imagery
  - Multi-spectral analysis
  - Airborne particle detection
  - RFID tag location detection
  - Etc.

Introduction

- Personal remote sensing must be (among other things) <u>low cost</u>
  - Commercial sensors and other elements are used whenever possible
  - We use several different kinds of Consumer Commercial Off-the-Shelf (CCOTS) hardware to accomplish our missions
  - Standard interconnect busses etc. (I2C, SPI, RS232, etc.)

Data Mission Assurance

- Like any unmanned systems, UASs are defined by their missions
- Remote sensing is the collection of data
- PRS UASs fly to collect data
- Our paradigm: "Data is the Mission"
- Therefore: Data Mission Assurance is needed for PRS

Data Mission Assurance

- What reason do we have to fly without a payload?
- Mission assurance also equates to “resilience”
- Robots are notoriously “fragile” to environmental changes
- We want to fly in the airspace with other well-engineered systems
- DMA is necessary to provide entities such as the FAA with good reasons to trust us
  - **Airworthiness**

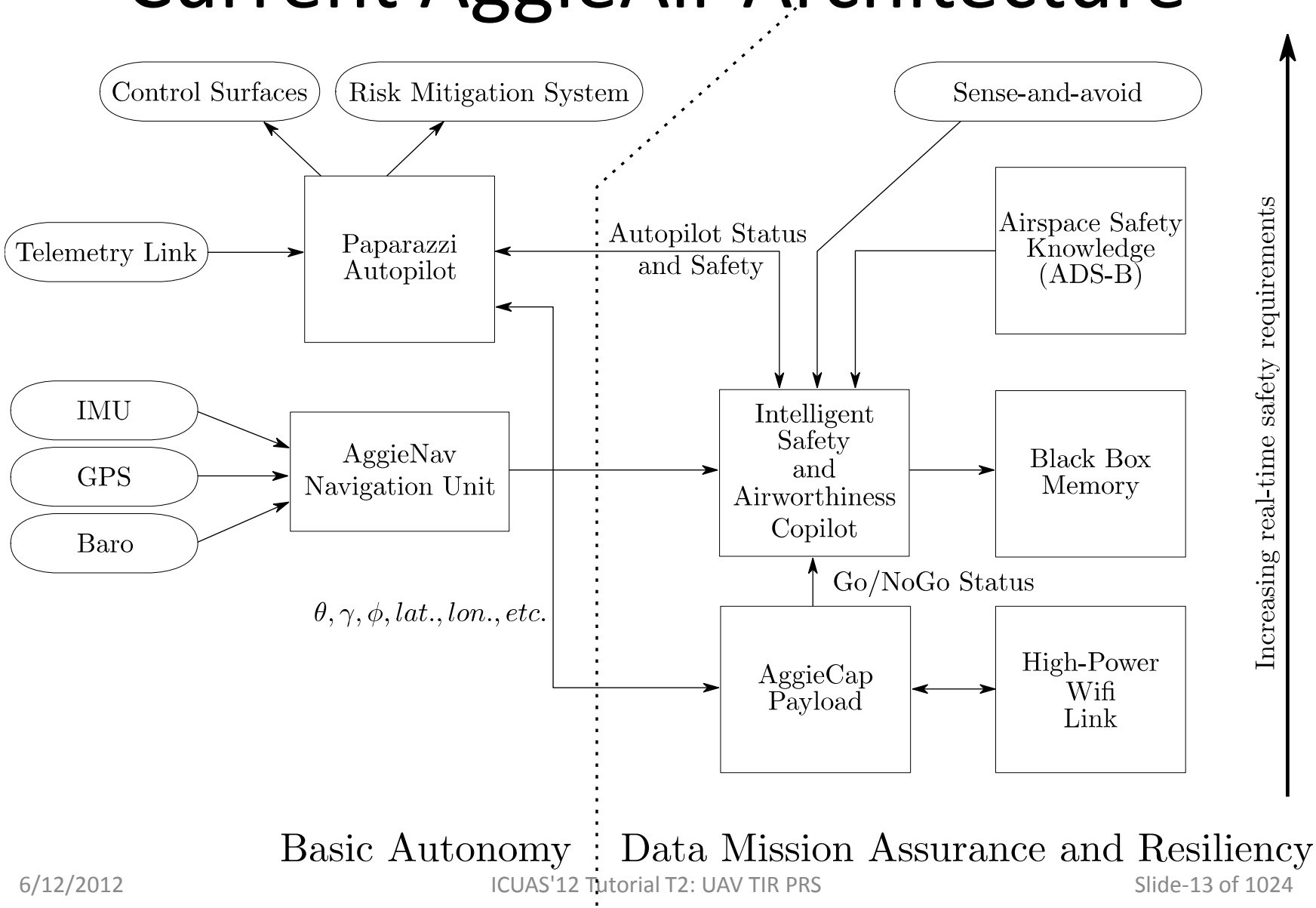
Data Mission Assurance

- How do we assure success without traditional methods (hot-spare, radiation-hardening, TMR, etc.)?
  - Well, we can not (to that same degree)
- What can we do?
  - Systems engineering (choosing best components for the job)
  - Software engineering
  - Adding intelligence to key elements of the system
  - Testing
  - Testing!
  - More testing! (less testing in the field)
- Inexpensive aircraft: bring a full backup system

AggieAir Architecture

- AggieAir philosophy:
  - Start with a well tested, flying airframe
  - Use this platform for data collection (add payload)
  - The payload should not (reasonably) be able to interfere with the normal flight performance of the aircraft
  - The payload should be adaptable and change with various flight parameters
  - The payload should be able to communicate with the operators about important information

Current AggieAir Architecture

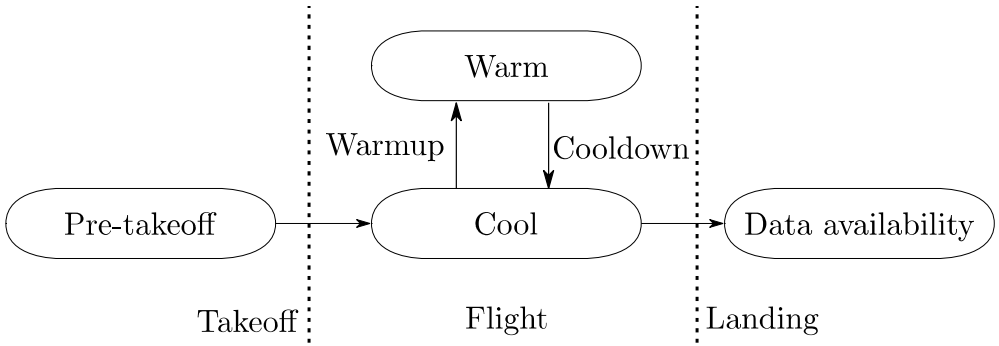


AggieAir Architecture

- AggieAir is designed for scalability
  - "If you don't need it, we don't fly it"
  - We think most people will need it all, however
- Failures roll "downhill"
  - An errant payload will not crash the aircraft
  - Reasonable payload requests can be attended to be the autopilot
- AggieAir works
  - We started with a robust UAS package (Paparazzi) and built AggieAir
  - We have tested things modularly, allowing bugs to be worked out before complexity becomes prohibitive

AggieAir: PRS Payload Requirements

- Payloads are tested and verified in a standardized way before integration with the airframe for resilience reasons
- The payload should collect data when needed for minimal power and complexity
- Payloads have a state machine for missions:



AggieAir: Go/NoGo Status

- Before flight, payloads go through internal Go/NoGo (GnG) checks to assure that faulty sensors will keep flights grounded until operators can fix errors
- GnG signals can be routed to useful places such as ground station operators, warning lights, klaxons, etc.
- GnG signals help increase DMA

ISaAC: The Intelligent Safety and Airworthiness Copilot

- A PRS UAS is able to fly with simply a autopilot and payload such as AggieCap
- However, to provide safe and reliable data
  - Over various external and internal conditions
  - Over many different kinds of missions
- It is possible to integrate knowledge about the surrounding airspace, weather, and operator conditions to increase safety and airworthiness

AggieAir: ISaAC

- ISaAC's main function is to record ('black box' functionality):
  - Global pose (roll, pitch, yaw, X, Y, Z) information
  - Payload status
  - RC safety radio link data
  - Autopilot status
  - GnG Messages
- ISaAC also is connected to the ADS-B network, allowing real-time knowledge of the position and status of other aircraft in the nearby airspace

AggieAir: ISaAC

- ISaAC also facilitates communication between payloads and the autopilot
- This allows the UAV ground station to view payload status and send commands to payloads
- ISaAC will also allow data-driven missions by processing payload commands and updating the autopilot, changing the mission to collect more interesting data
- ISaAC increases DMA

AggieCap Modular Payload Software Architecture

- Since PRS UASs can have many different configurations for payloads, modularity allows for quick reconfiguration, even in the field
- To enable this functionality, modules are tested and verified independently, providing DMA as part of the development process
- Tested modules can be reused and modified for speedy prototyping

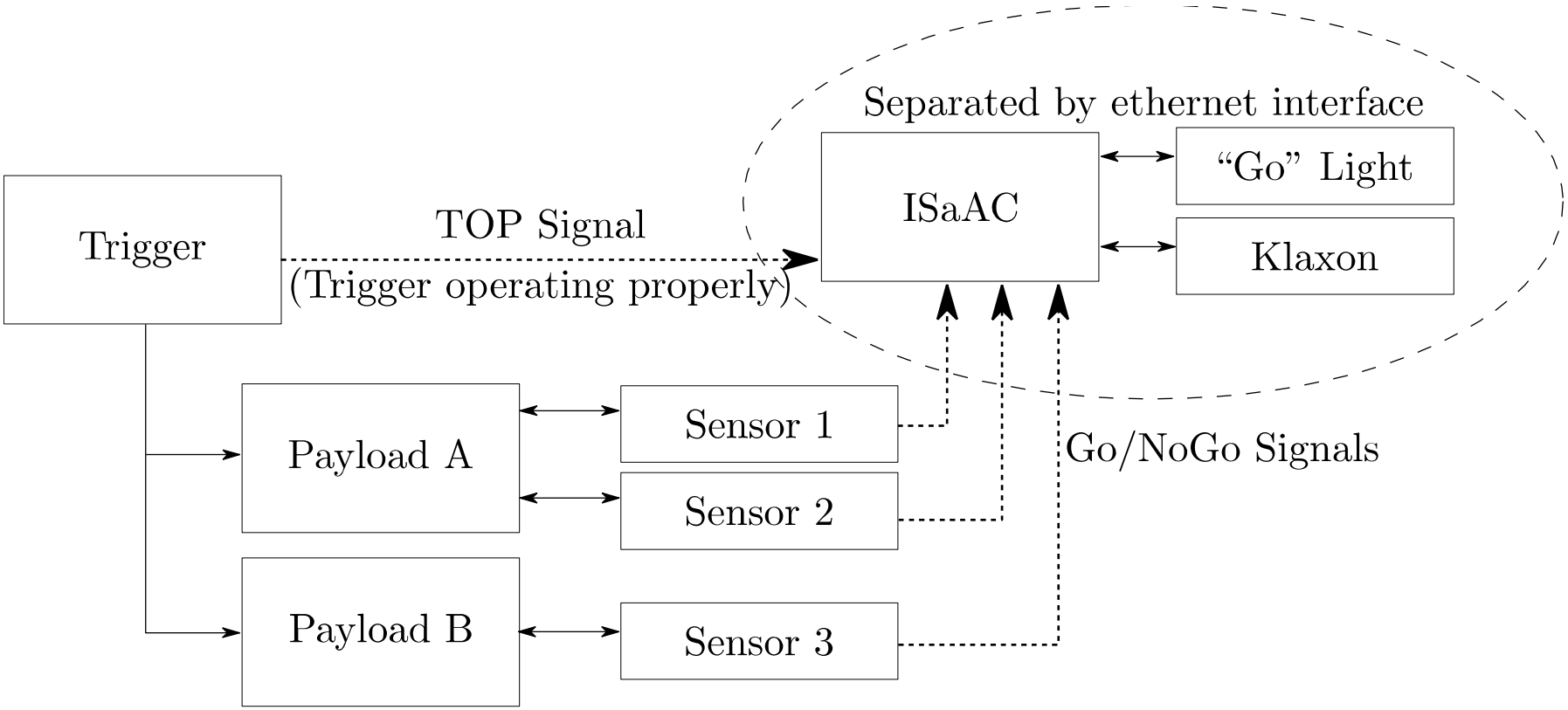
AggieCap Modular Payload Software Architecture

- Abstraction of the sensor and actuator architectures is the key to achieving modularity
- Careful software engineering gives readily available modules, code reuse, and standardized testing before actual missions are flown
- Small PRS UASs make use of CCOTS hardware, so fragility is assumed. Therefore intelligent error logging is important for DMA. Modularity allows this to happen in a standard way

AggieCap Design

- AggieCap is composed of three main parts:
  - Sensors and Actuators
  - Trigger
  - ISaAC interface
- Sensors take data and make sensor status logs
- Actuators are abstractions of any physical output
- Sensors and actuators are grouped together into Payload objects, which are then inherited by the Trigger object
- AggieCap is based on Unix fundamentals, allowing various parts to be on local or separate computing hardware for maximum flexibility and modularity

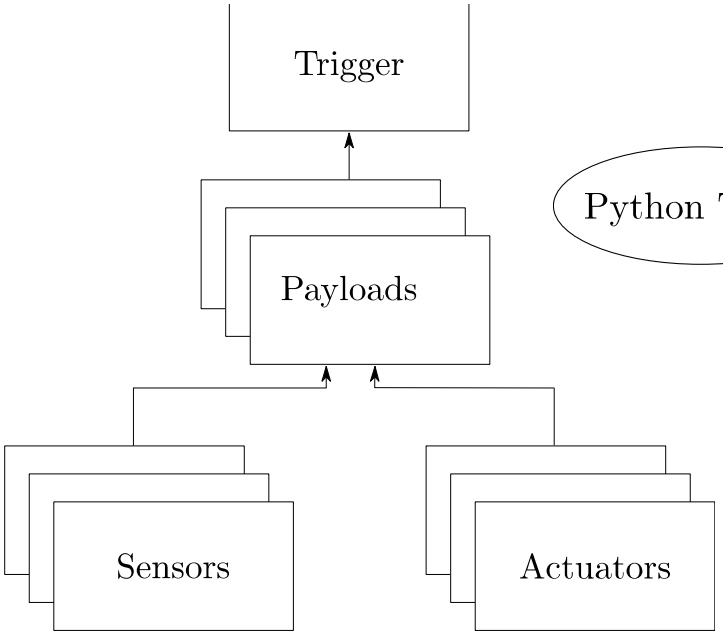
AggieCap: GnG Block Diagram



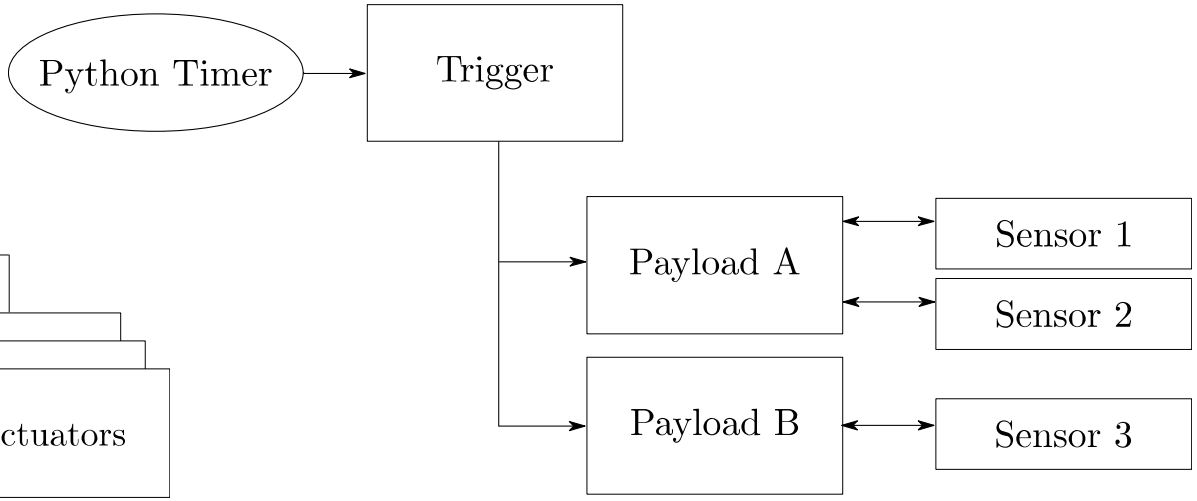
AggieCap Design: Trigger

- The Trigger object is responsible for timing and synchronization of data captures from the various Sensors
  - Ex: Two cameras, each triggered a 4-second intervals for 2-second effective ground coverage
- The Trigger also receives the pose data from the navigation system, so it can be associated with each Sensor capture event
- The Trigger also updates the ISaACwith a regular heatbeat message (TOP) to show AggieCap is functional

AggieCap Design: Trigger



Inheritance



Timing

AggieCap Implementation

- AggieCap is implemented in the Python programming language
  - Maintainability
  - Human readability
  - Platform independence
  - Ease of prototyping
  - Ability to link in external binary object libraries
- Python is a high-level language, so it is easy to read and interpret.
  - Much higher chance of ‘coding what you mean’

AggieCap Implementation

- Python lacks pointers; this lowers the risk of memory access violations
- Python is interpreted, so most problems with platform-independence are avoided
  - Linking and loading of binary drivers are allowed if these exist for a target architecture
- Python is automatic garbage collection
  - It's more difficult to leak memory
- Yes, Python *is* slow, but nearly 100% of AggieCap's processing is IO-bound, making code speed irrelevant

Standardized Testing and Verification for DMA

- Achieving resilience is a challenging task, especially in Systems like AggieAir where cost is a driving factor in system architecture
- Sensors are prone to faults
  - CCOTS properties (cheap hardware)
  - “Unofficial” Open-Source interface libraries such as gPhoto and the Cannon Hardware Development Kit (CHDK)

Standardized Testing and Verification for DMA

- AggieCap is part of AggieAir's standardized testing framework
- Payloads undergo payload-in-the-loop (PiL) testing
  - Payload hardware is built as a separate unit
  - A full mission simulation is injected just a it would be during a flight
  - Payload operation is observed in real-time
    - Hardware and software
- Airframes and payloads can be developed in parallel, speeding overall development

Standardized Testing and Verification for DMA

- PiL testing allows for a scientific environment in which to observe new hardware and diagnose many problems
  - Fragility boundaries such as captures-per-second can be explored
- With the GnG, interactions between various sensors can be studied and robustness verified

Standardized Testing and Verification for DMA

- AggieAir payload testing rubric:
  - Physical: power usage, electro-magnetic interference, wiring etc.
  - Environmental: temperature, humidity via test chambers, vibration
  - In-flight faults: USB disconnection, storage space full, low battery, etc.
- Given logs, proper fault tree-based mission responses can be tested.

AggieCap Sensors and Actuators

Nikon D90	DSLR camera (visible and NIR)	gPhoto
Canon T1i	DSLR camera (visible and NIR)	gPhoto
Canon s95	Small still camera (visible and NIR)	CHDK
Canon is110/115	Small still camera (visible and NIR)	gPhoto
Sensoray 2255	Video framegrabber	USB
FLIR Photon 320	Thermal IR camera	Sensoray
ICI7640	Digital thermal IR camera	USB
Lotek Nanotag Radio	Small animal radiotag locator	CSOIS internal
System	Computer hardware health	Linux System
Pololu Micro Maestro	6-channel servo controller and I/O	USB
Microswitch	Limit switches for motion control	Pololu Maestro
Standard servo motor	General servo actuator	Pololu Maestro
Firgelli linear actuator	Miniature actuator w/ position feedback	Pololu Maestro
Overo System LED	System I/O	Linux Kernel
Overo audio CODEC	Audible GnG Warning	Linux Kernel

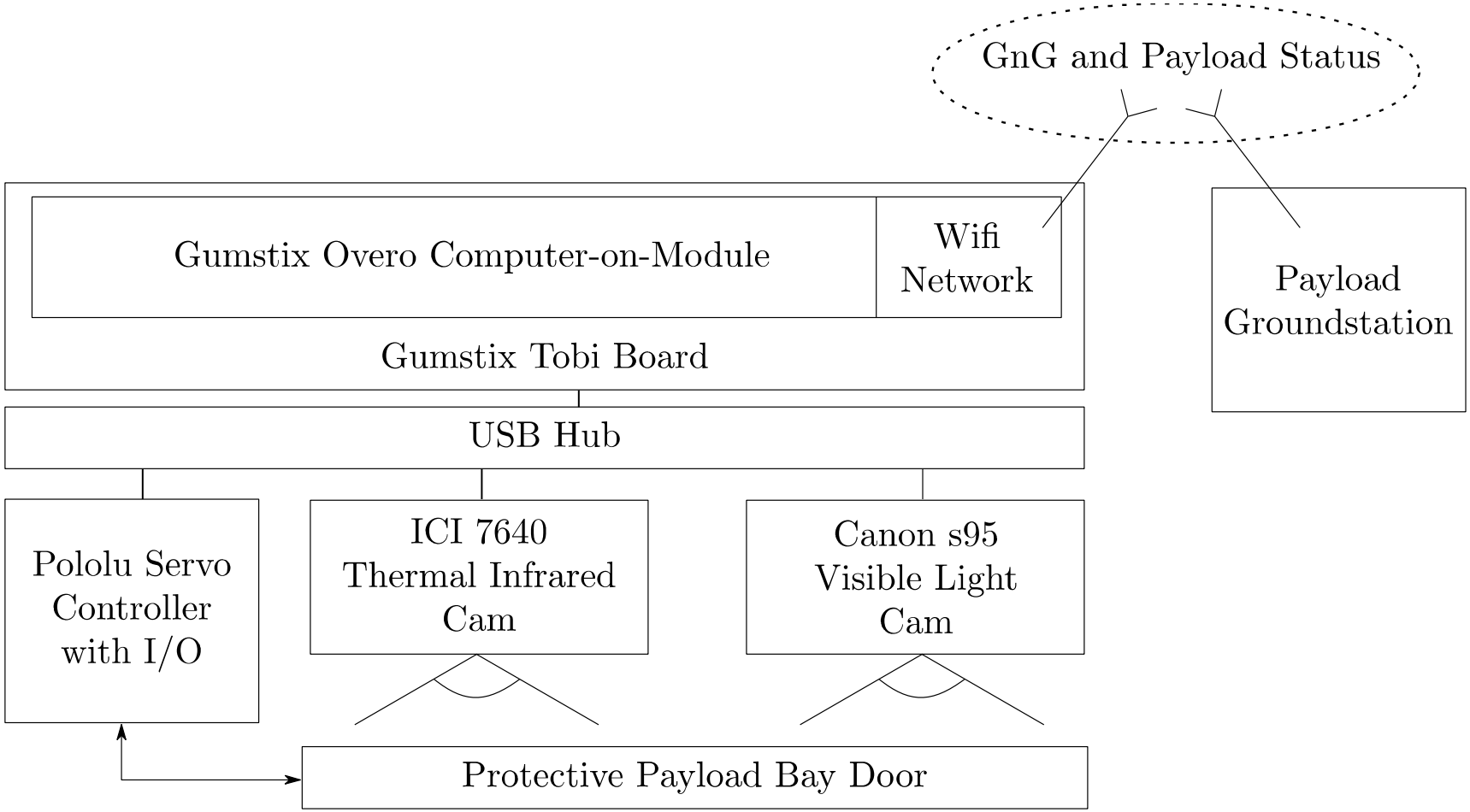
AggieCap Thermal Infrared Payload

- CSOIS has used AggieAir and AggieCap to create a TIR multispectral payload
  - Infrared Cameras, Inc. 7640 TIR camera (USB)
  - Cannon s95 CCOTS point-and-shoot camera
- We used the CSOIS fixed-wing payload box
- We created an actuated door system so we can land safely on the payload
- Camera trigger cadence: 4 seconds

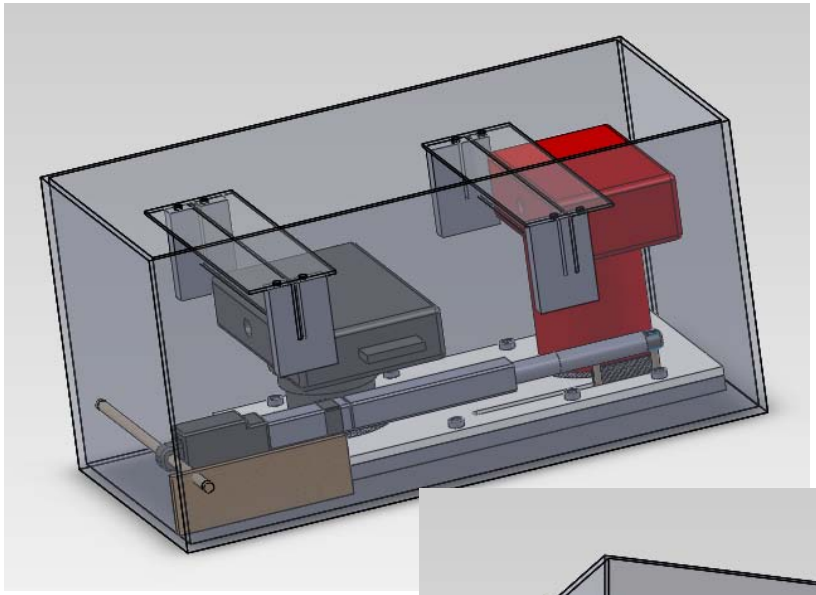
AggieAir Payload Mechanical Design

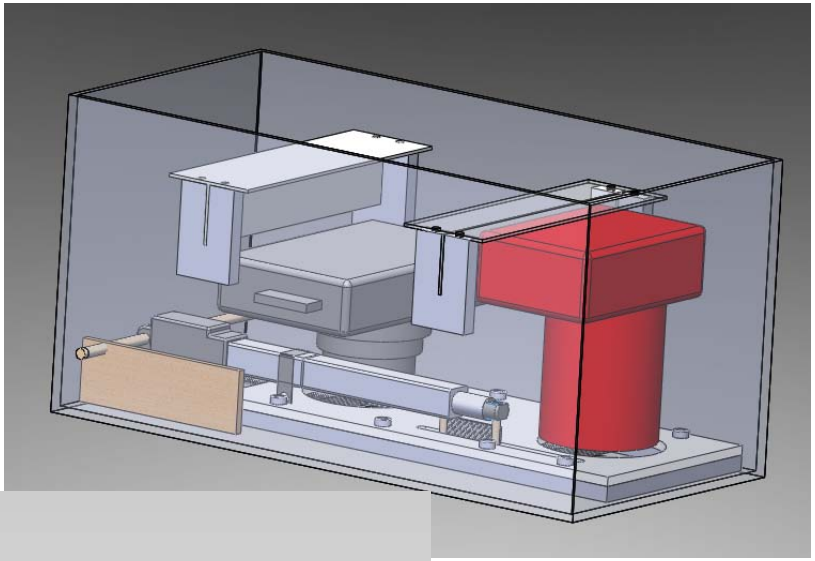
- Payload boxes are interchangeable in the field and come with ground support equipment (GSE) and ground control stations (GCS)
- AggieAir payloads are integrated into a standardized box
- 5” x 5” x 10” (1.3Kg)
- Made of carbon fiber/Kevlar weave to withstand the rigors of AggieAir belly landings, usually on rough ground

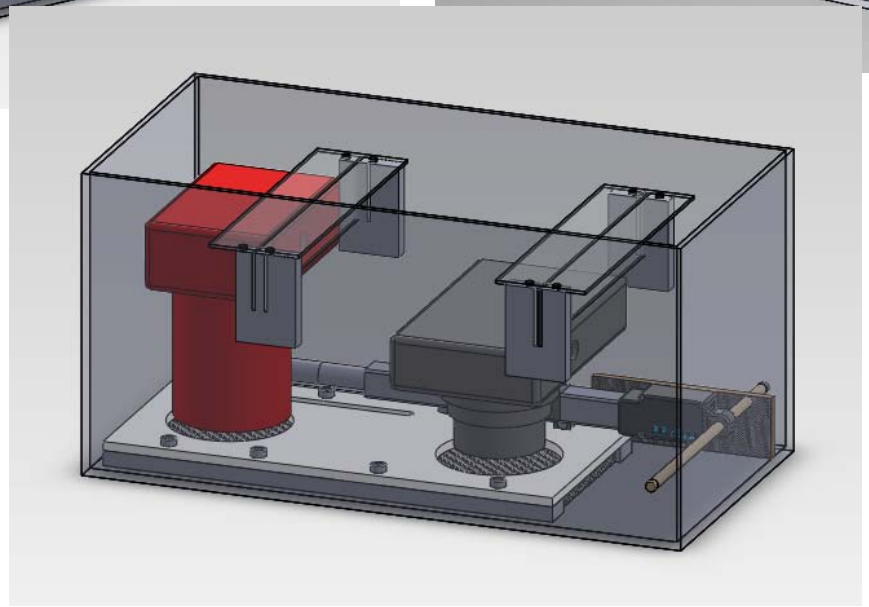
AggieCap Thermal Infrared Payload



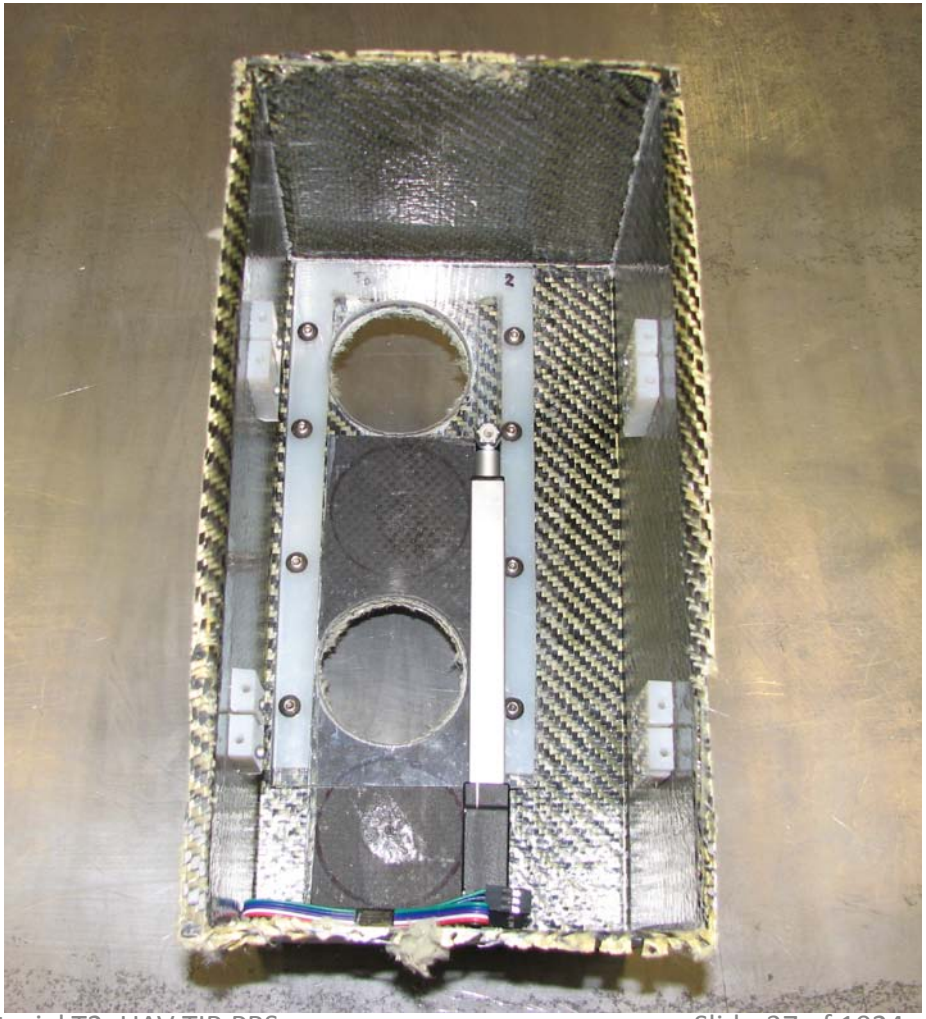
AggieCap Thermal Infrared Payload







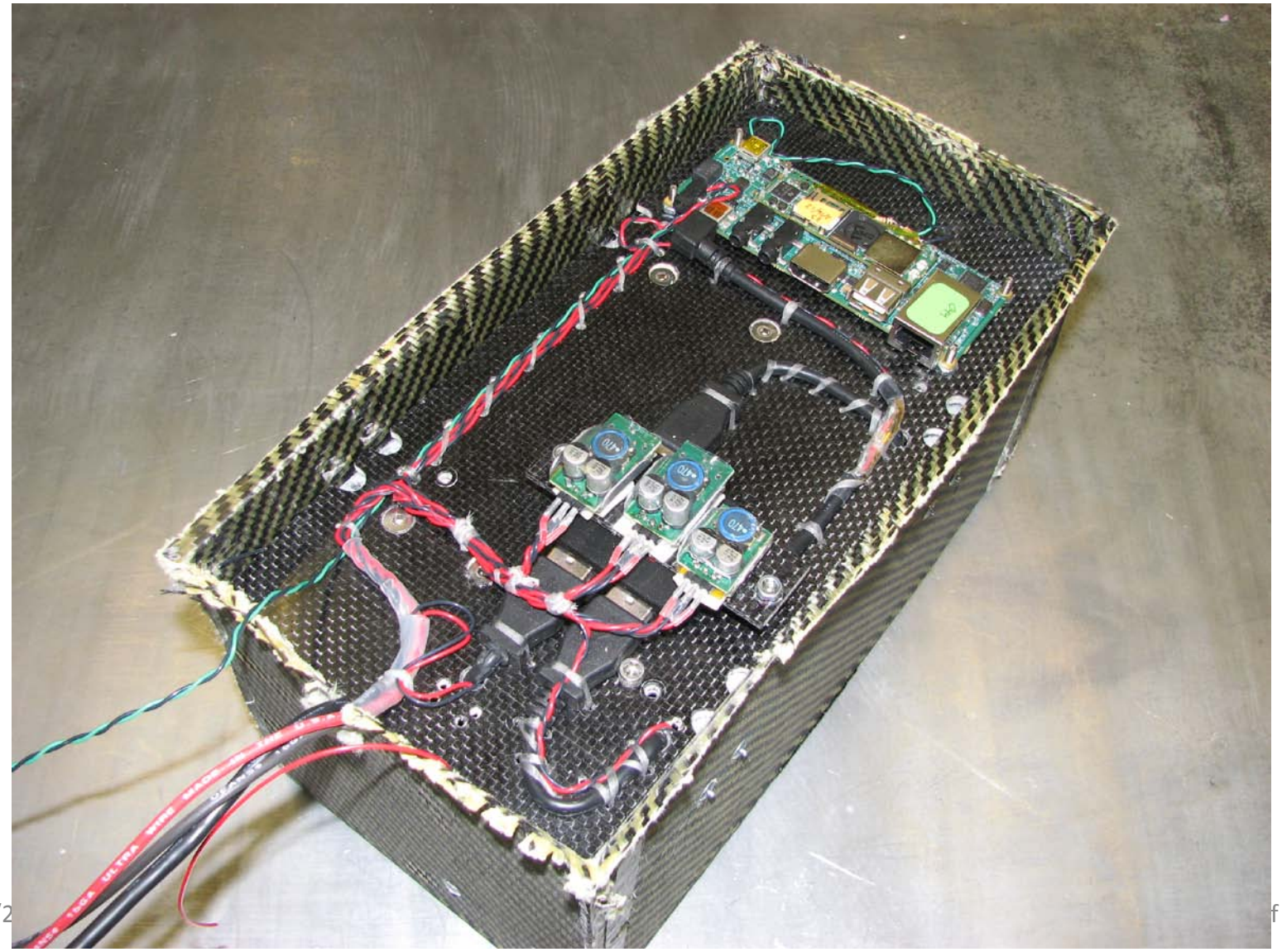
AggieCap Thermal Infrared Payload



AggieCap Thermal Infrared Payload

AggieCap Thermal Infrared Payload

AggieCap Thermal Infrared Payload



AggieCap Thermal Infrared Test Flight

- After simulation of the payload in the laboratory, we tested the payload in the air
- We flew for 37 minutes, and collected 365 total images
- These images were stitched to form a composite thermal map of the area
- This same system was then flown on an actual field site, more data from Mr. Austin Jensen…

AggieCap Thermal Infrared Payload

- Stitched farm TIR data:

Conclusions and Future Work

- Presented have been motivations, requirements and CSOIS' designs for PRS UAS payload designs
- I have detailed our system design decisions for both hardware and software
- Our multispectral payload with TIR and visible light was also shown, along with data collected while testing
- Future work will include more spectrums
  - HD-NIR, HD-TIR, SWIR, hyperspectral sensors

Acknowledgements

- This project is supported in part **by Utah Water Research Laboratory MLF 2006-2012** and by **NASA UAS2NAS program (Grant Number NNX11AO77A)**. Part of the work was also supported by **NSF REU Site program (Award # 0851709)**.
- CSOIS would like to thank all OSAM UAV team members in CSOIS and the Paparazzi UAV community for providing the wonderful paparazzi project to work on, as well as Gumstix, Linux, and many other wonderful open-source projects
- Special thanks to Infrared Cameras, Inc. for working with CSOIS to make their cameras compatible with AggieAir

Thank you!

- EOP.
- Any questions?