# Sensor Data Transplantation for Redundant Hardware Switchover in Autonomous Vehicles

Cailani Lemieux Mack[*], Kevin Leach[†], Kevin Angstadt[*]

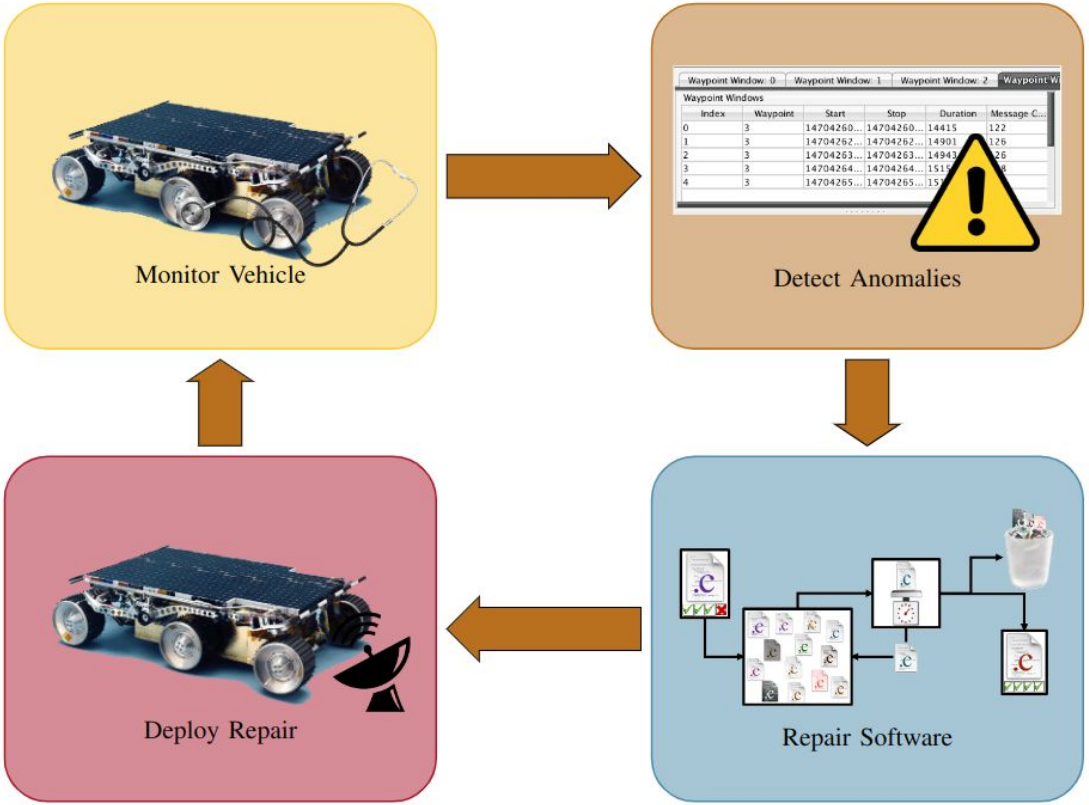
[*] St. Lawrence University [†] Vanderbilt University

# Problems with Autonomous Vehicles

- Increasingly used to complete previously difficult and dangerous tasks
- In many cases, failure could be catastrophic
- Both security vulnerabilities and software defects have been documented in control software
- An attacker could remotely compromise the control software

# Previous Work

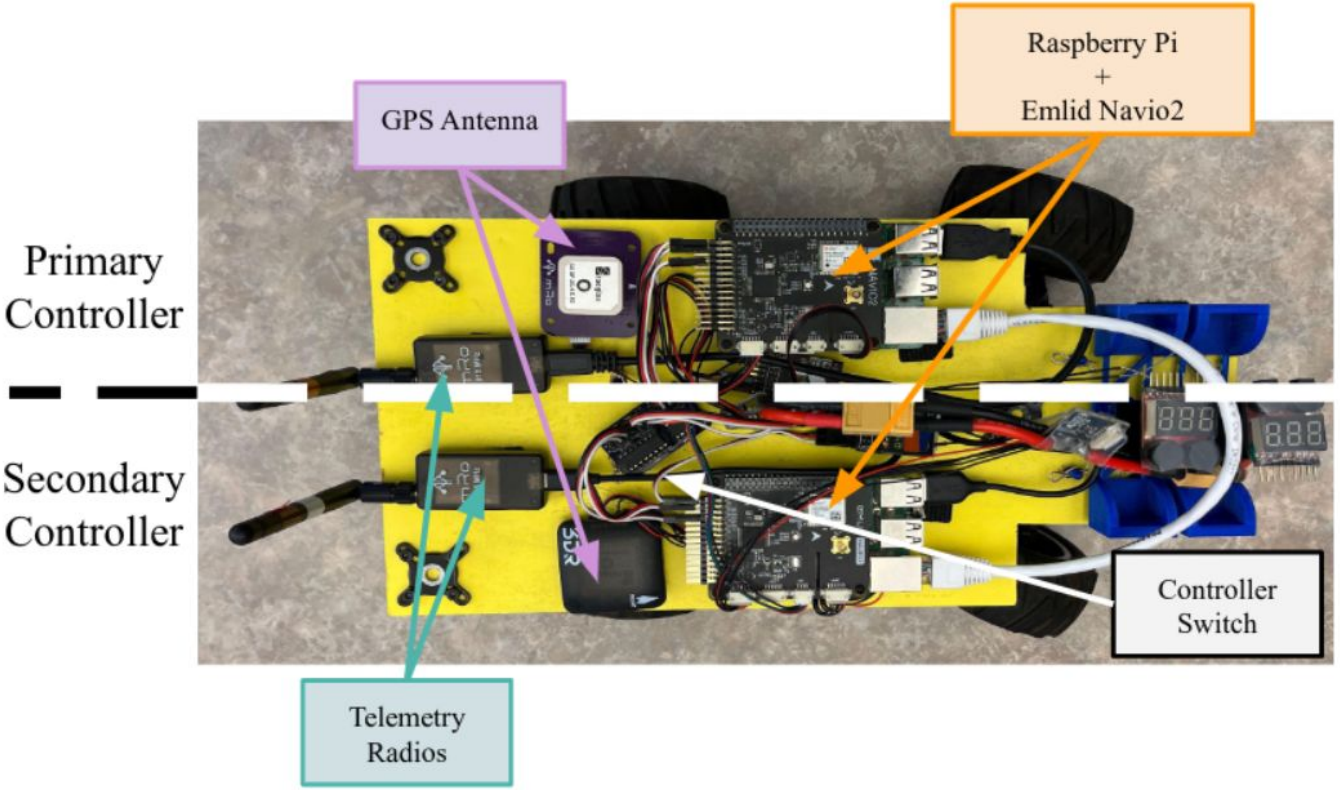
START: Software Techniques for Automated Resiliency and Trustworthiness



# Adding Redundancy

- We add a redundant flight computer to the drone that can take over control in an emergency
- Simplex architecture
  - The **primary** computer controls the motors
  - If the primary computer fails, control switches to the **secondary** computer
- Restarting the primary computer during a mission is not supported by current software
  - Requires the vehicle to be motionless for the restart
- Support does exist for larger systems
- The goal of this project was to add this support to the software

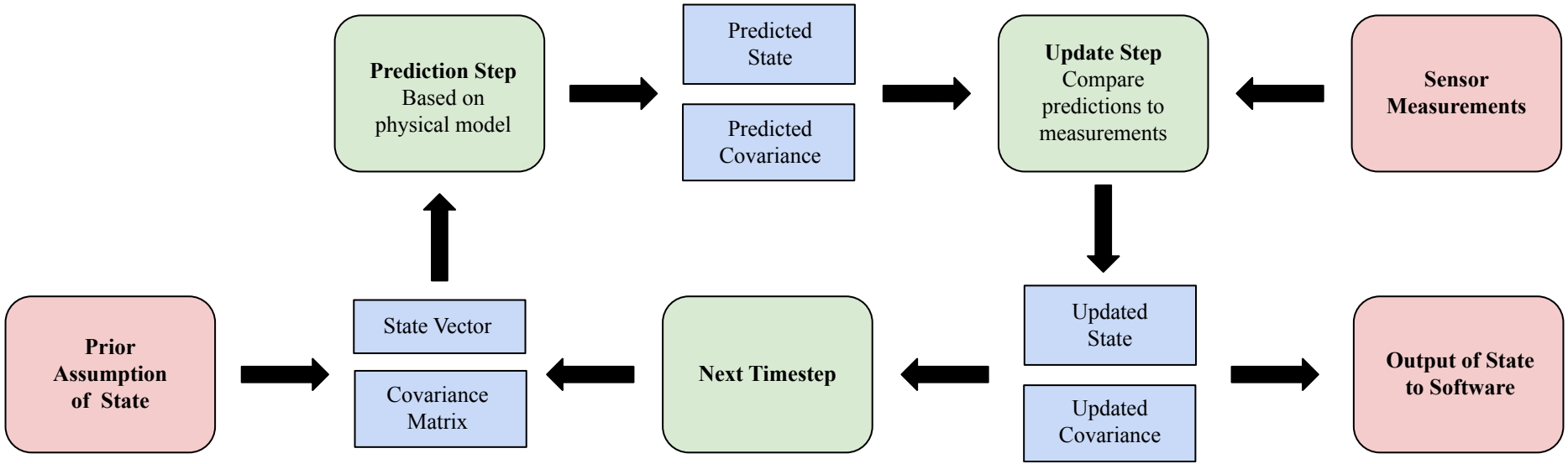
# Rover Setup


Raspberry Pi
+
Emlid Navio2
GPS Antenna
Primary
Controller
Secondary
Controller
Controller
Switch
Telemetry
Radios

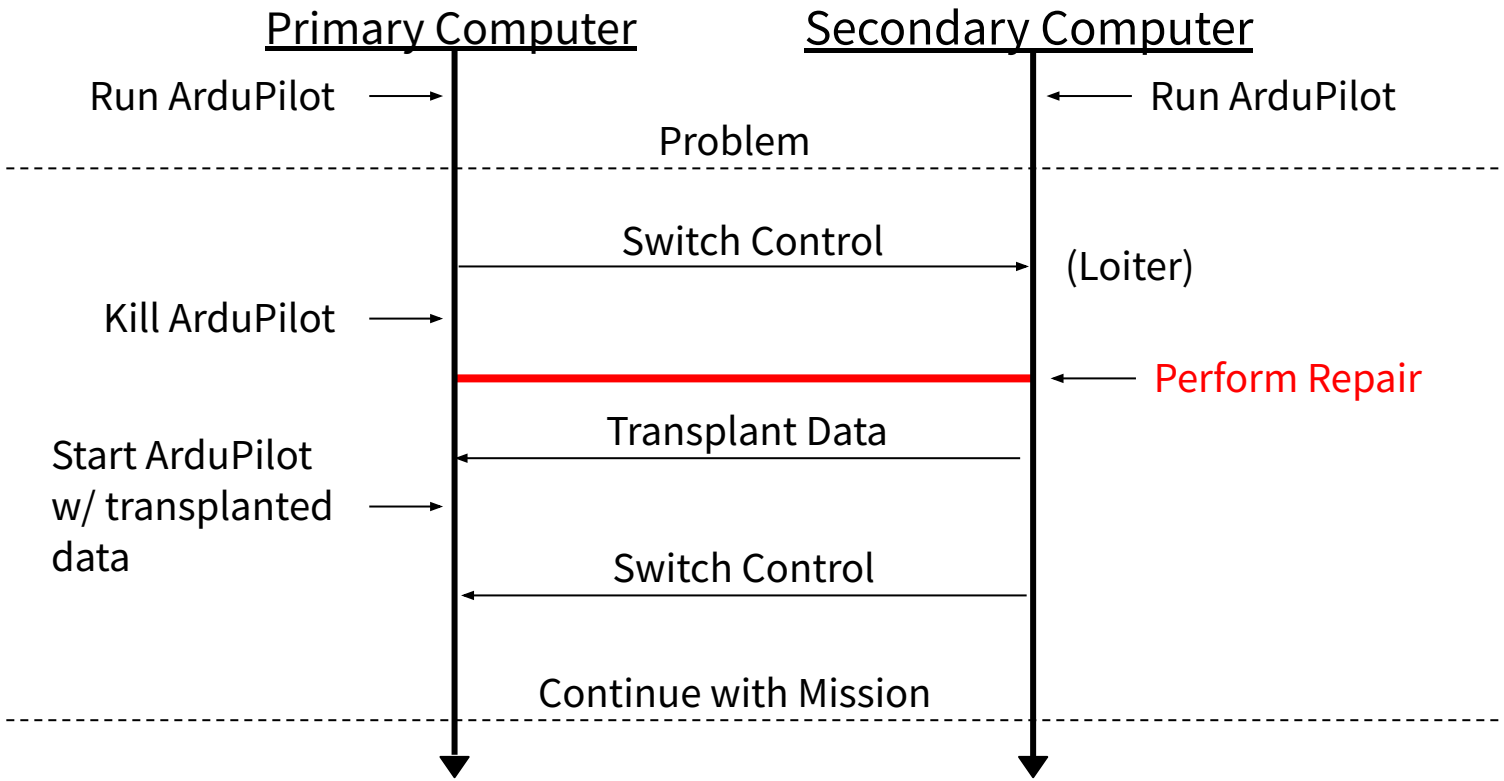
# Our Approach: Data Transplantation

- The state of the two controllers should be almost the same
- Thus we can insert the known state of the primary computer upon initialization
- Transplant data about the state of the rover from the secondary to the primary computer
- Allows us to skip much of the initialization of the primary computer

# Extended Kalman Filter



# Transplant Algorithm



# Experiments

Want to address the following research questions:

RQ 1. Is the control software able to initialize with the vehicle in motion?

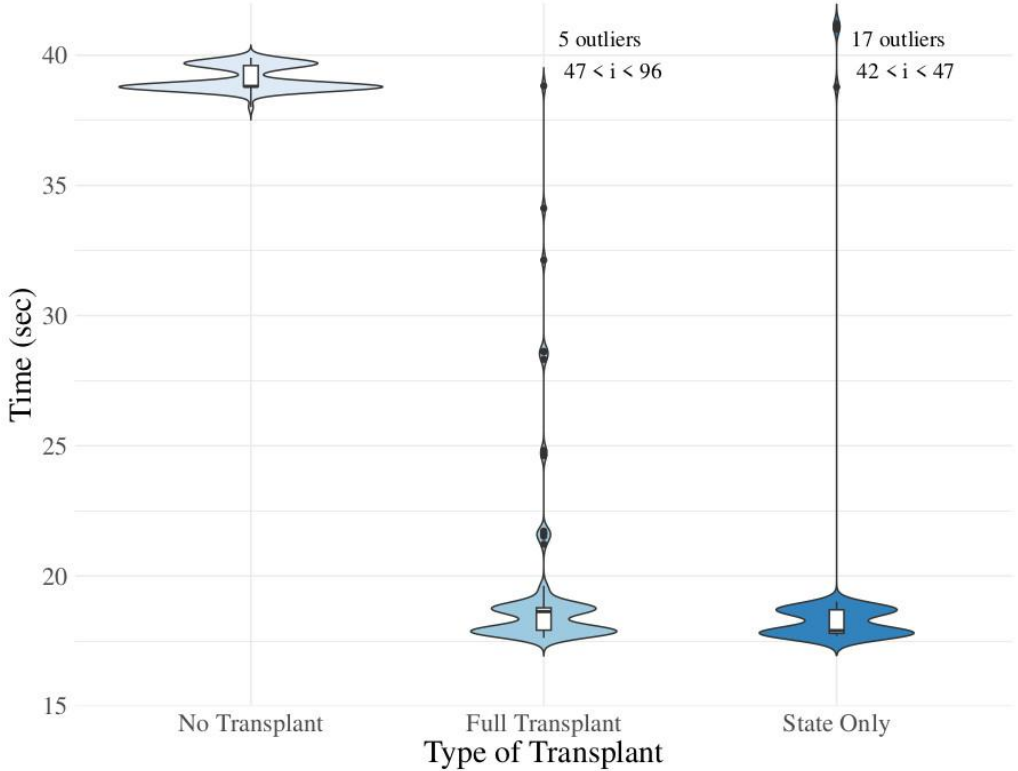
RQ 2. Does transplanting sensor data reduce the initialization time for a stationary vehicle?

RQ 3. Does transplanting a subset of EKF data improve initialization times for a moving vehicle?

RQ 4. Is the transplant algorithm robust to perceived error in the transplanted state?

RQ 5. Is the dual-controller platform able to continue operation with a control switchover?

# Stationary Tests




Averages over 100 trials:

No Transplant: 39.081 seconds

Full Transplant: 21.482 seconds

State Only: 23.376 seconds

Transplanting EKF data still allows a stationary vehicle to initialize and results in a 40-45% reduction in the initialization time of Ardupilot on a stationary vehicle.

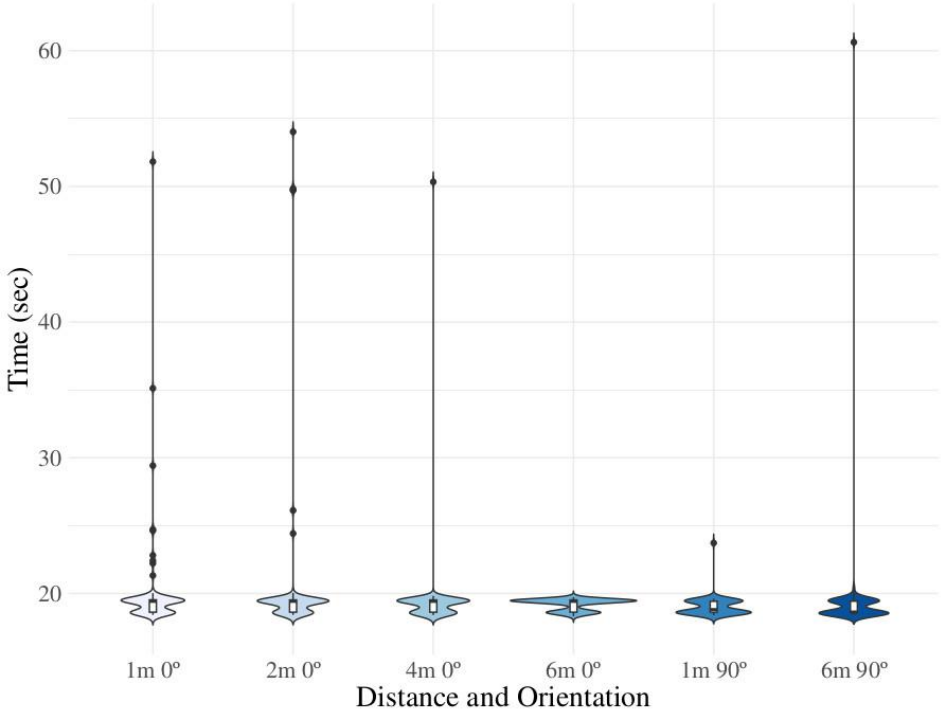
# Moving Tests

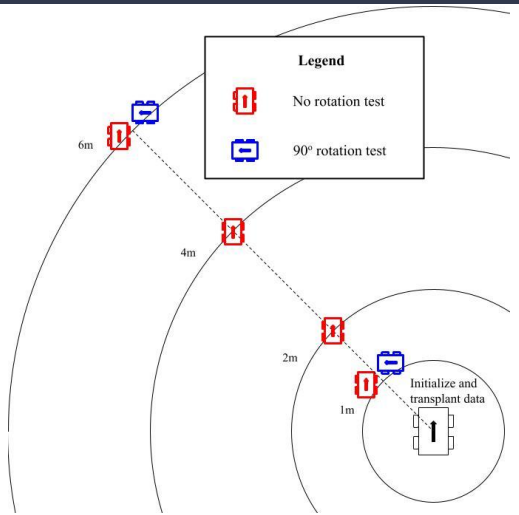
| Experiment | Avg. Init. Time (s) | Med. Init. Time (s) | Success |
|---|---|---|---|
| No Transplant† | – | – | – |
| Full Transplant | 47.196 | 29.524 | 9/10* |
| State Only | 42.899 | 24.736 | 10/10 |

† Without our modifications, Ardupilot cannot initialize in motion
* One trial failed to initialize for an unrelated hardware fault

Transplanting EKF data allows Ardupilot to initialize successfully while the vehicle is in motion. Transplanting only the state vector improves the median initialization time by approximately 16%.

# Acceptable Error Tests

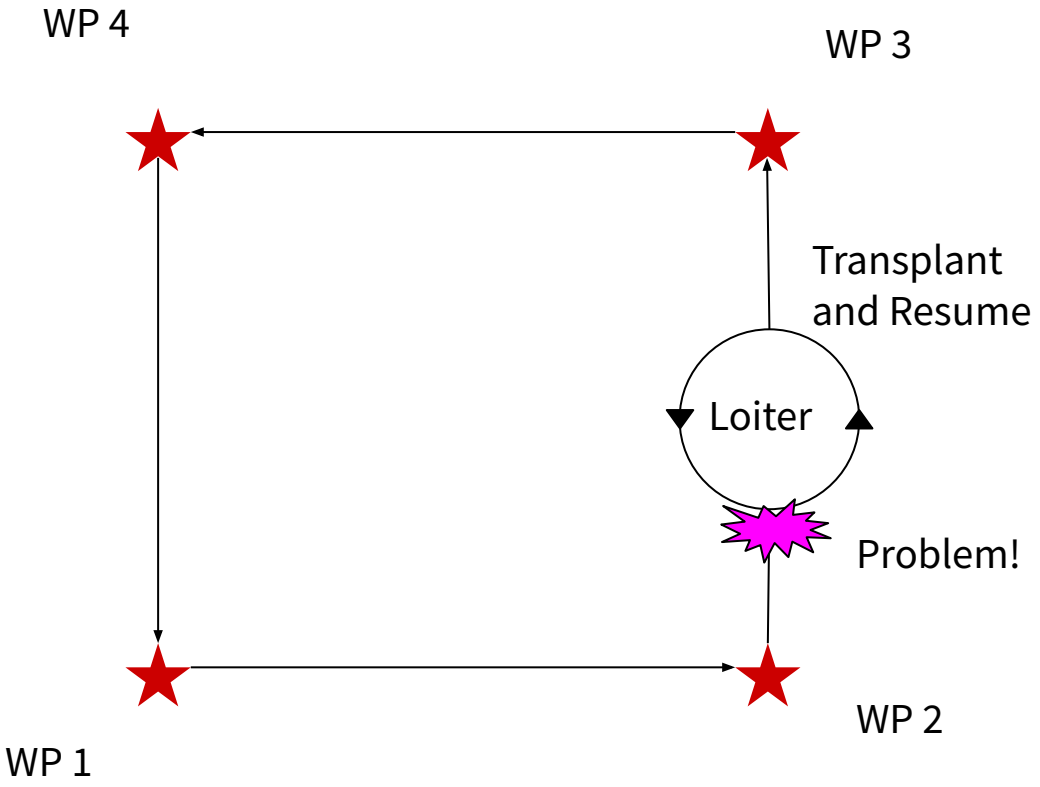



The distance travelled by the vehicle while EKF data is being transplanted has no discernible impact on Arupilot's ability to initialize. Initialization time are consistent across all measured distances and orientations

# End To End Tests

Link to video



# Your Work

The projects I found most interesting:

Human Detection Using WiFi Signals

Identifying and Locating Hidden IoT Devices in Untrusted Environments

Multi-User Augmented Reality with Collaborative Localization

# Thank You!
## Any Questions?