CETI | SHAMU

Cetacean **E**cholocation **T**ranslation **I**nitiative

Search and **H**elp **A**quatic **M**ammals **U**AS

Test Readiness Review

Team

Ian Barrett
Grant Dunbar
George Duong
Jesse Holton
Sam Kelly
Lauren McIntire
Benjamin Mellinkoff
Justin Norman
Severyn Polakiewicz
Michael Shannon
Brandon Sundahl

Customers

Jean Koster
James Nestor
David Gruber

Advisor

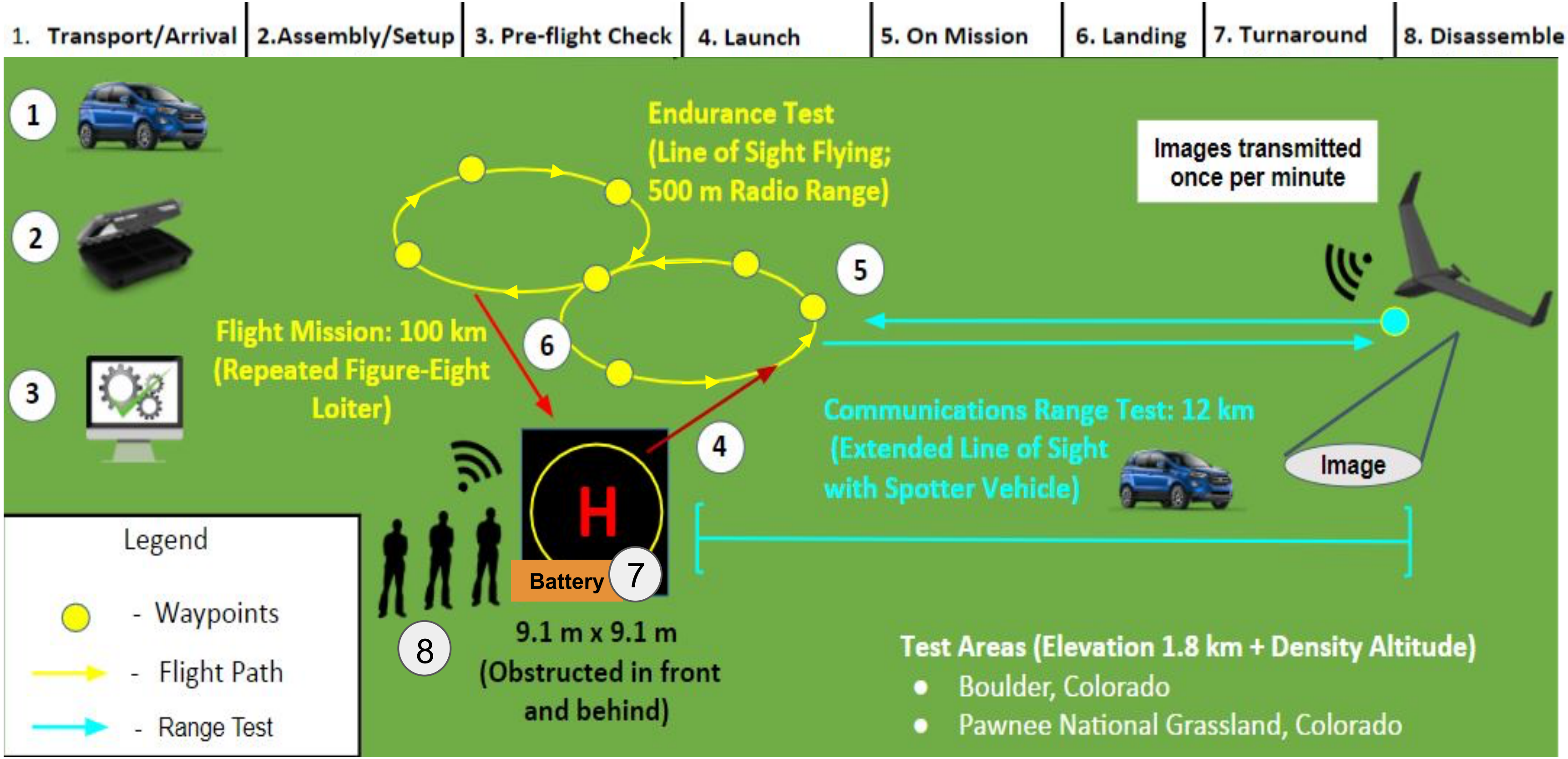
Donna Gerren

Project Description

Search and **H**elp **A**quatic **M**ammals **UAS**

will design an **unmanned aerial system** to carry a **future** instrument payload capable of **locating sperm whales in the ocean.** The future unmanned aerial vehicle will be **launched and recovered from a research vessel's helipad.**

SHAMU Test CONOPS



Levels of Success

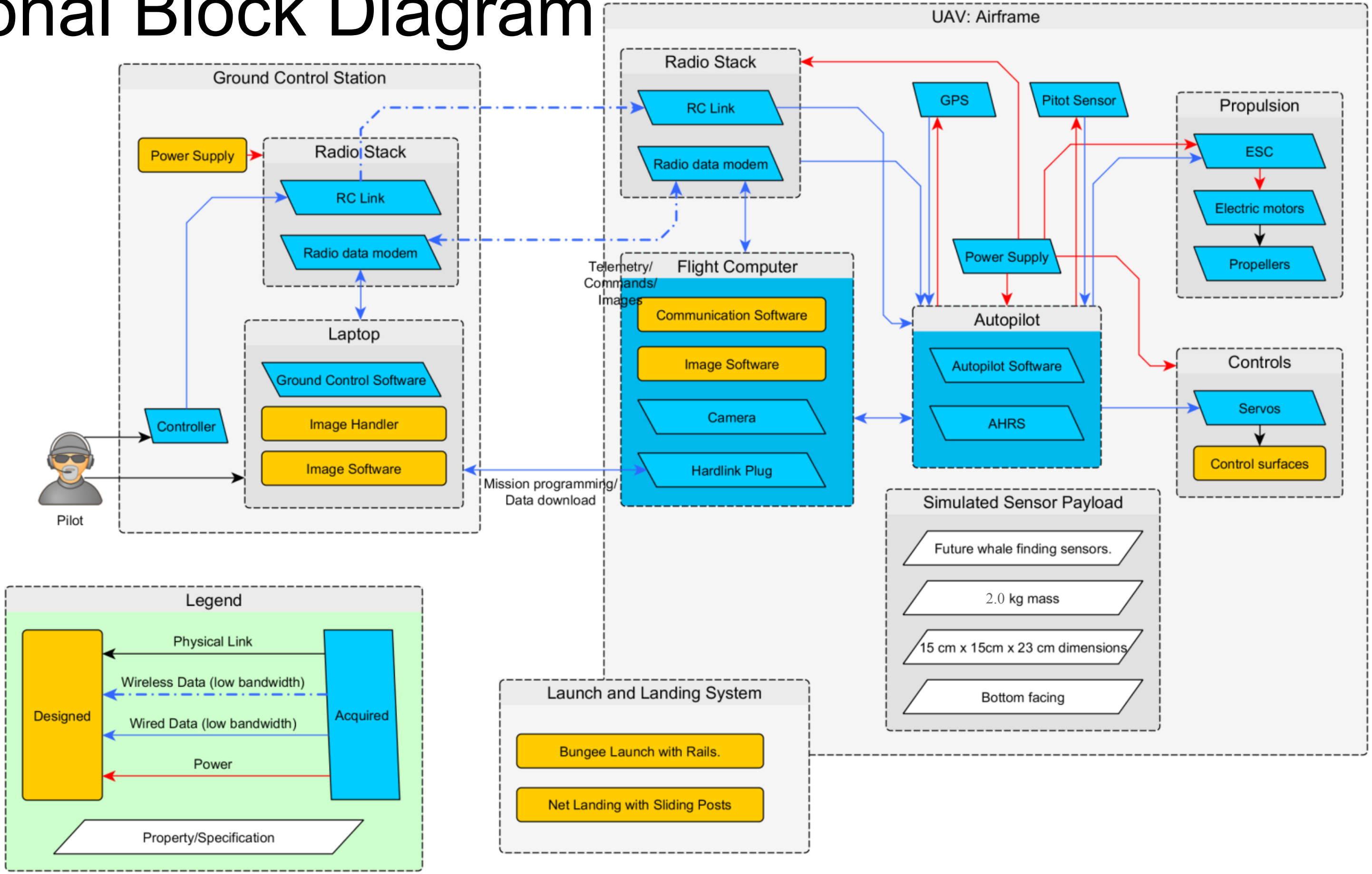
1. The **aircraft and associated systems pass ground tests**: Aircraft has **2 kg instrument payload** with **15 x 15 x 23 cm volume**; wing loading test of **5g**; aircraft mass below **22.7 kg**. Power source endures **1 hour** simulated flight mission. Locally **downlink telemetry**; **full manual control** over control surface servos.

2. The **aircraft is airworthy and proven to fly**: piloted **takeoff** and **landing**, **5 minutes** on mission, uplink **waypoints**, **telemetry** displayed to pilot.

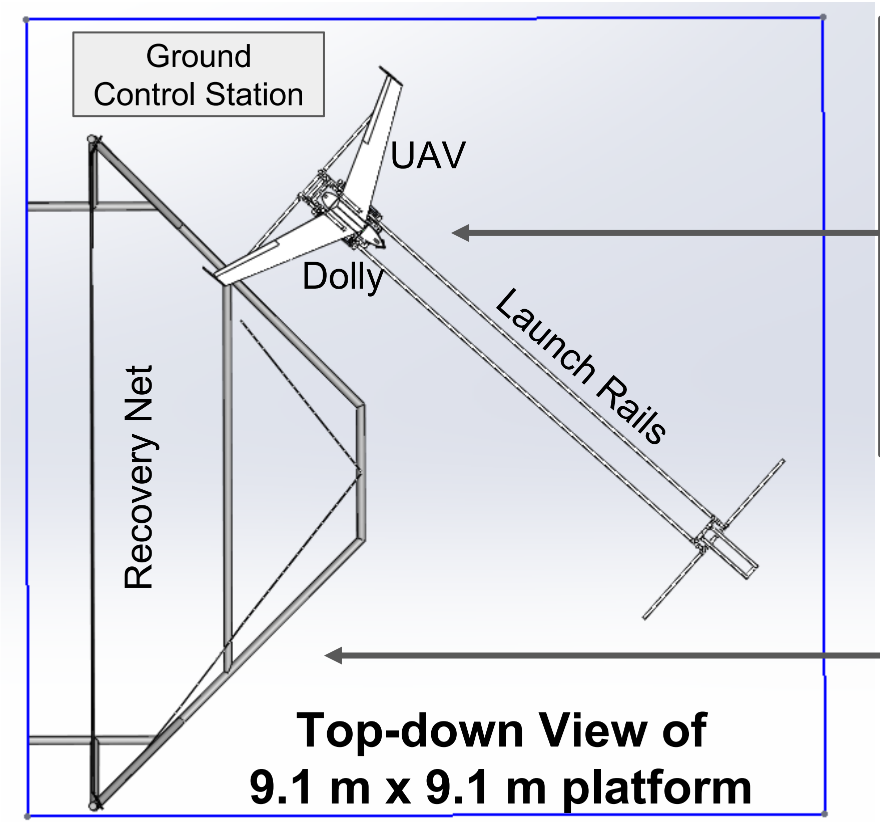
3. The **aircraft has improved flight performance**: **30 minutes** on mission, **full autonomy** at cruise, **500m** radio range, **images** are **saved onboard once per minute**.

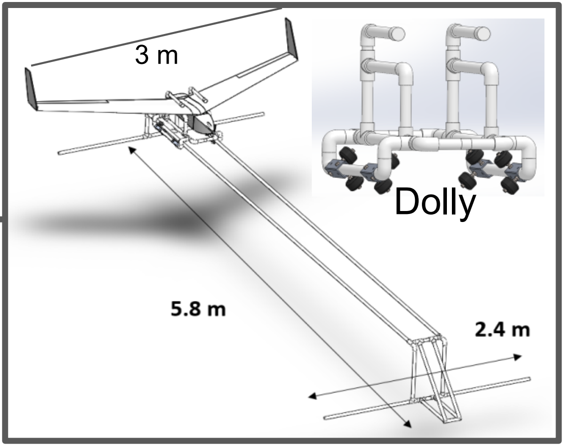
4. The **UAS meets all mission objectives**: **1.4 hours** on mission, **20 m/s cruise speed, 12km radio range, images transmitted once per minute.**

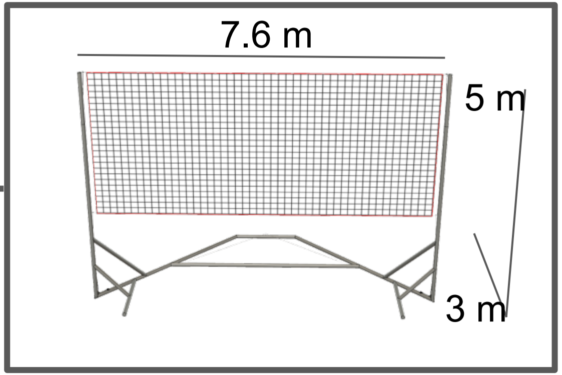
Functional Block Diagram



Review of Baseline Design







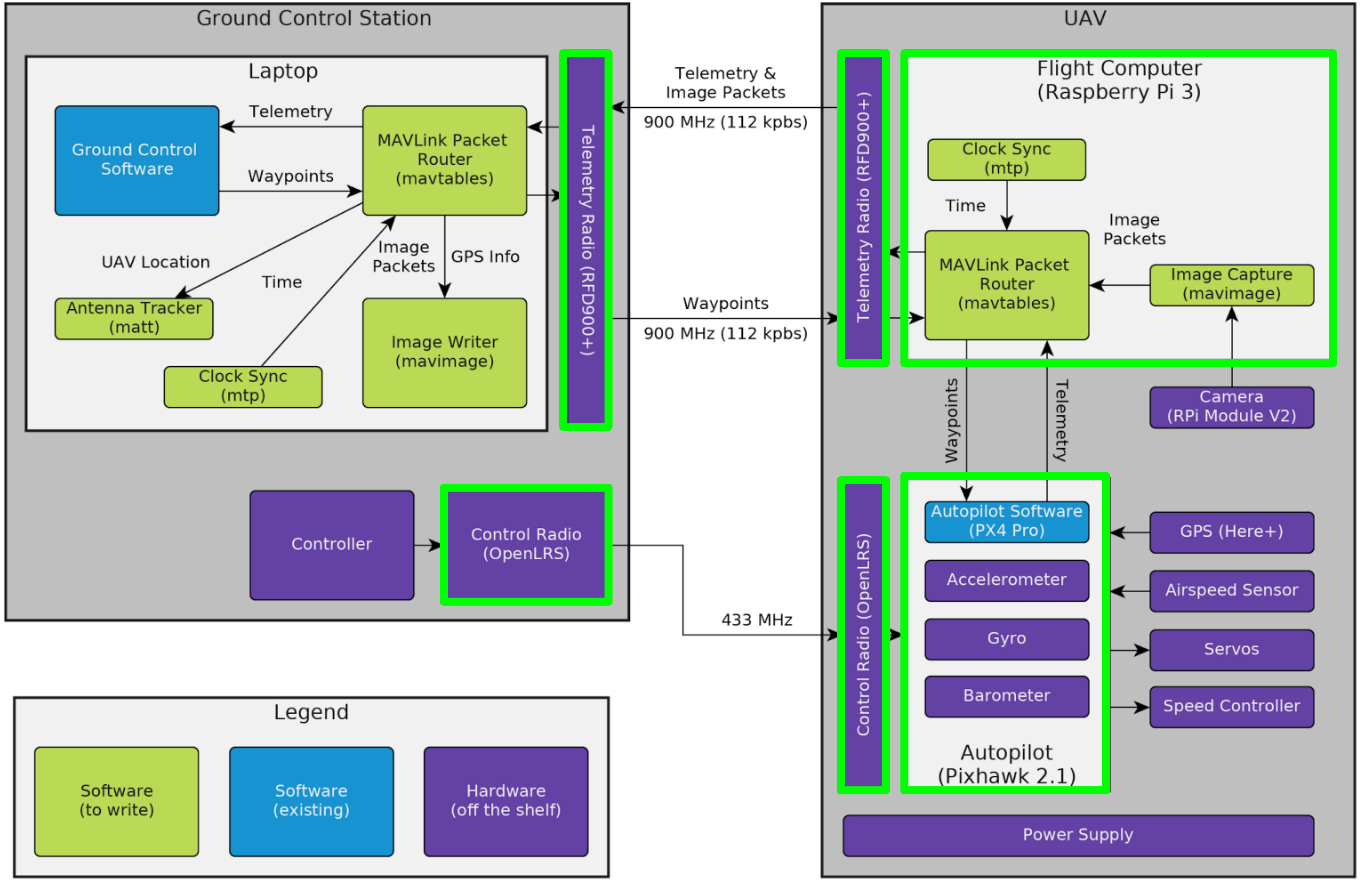
UAV on Launch Rails

- Dolley rides on rails
- UAV accelerated via dolley and bungees
- UAV ejected by sudden stop of dolley via restraining rope

Recovery Net

- Net extends
- Lines, pulleys, and bungees enable net extension
- Sailing cleat prevents rebound

Review of Navigation Hardware/Software Design



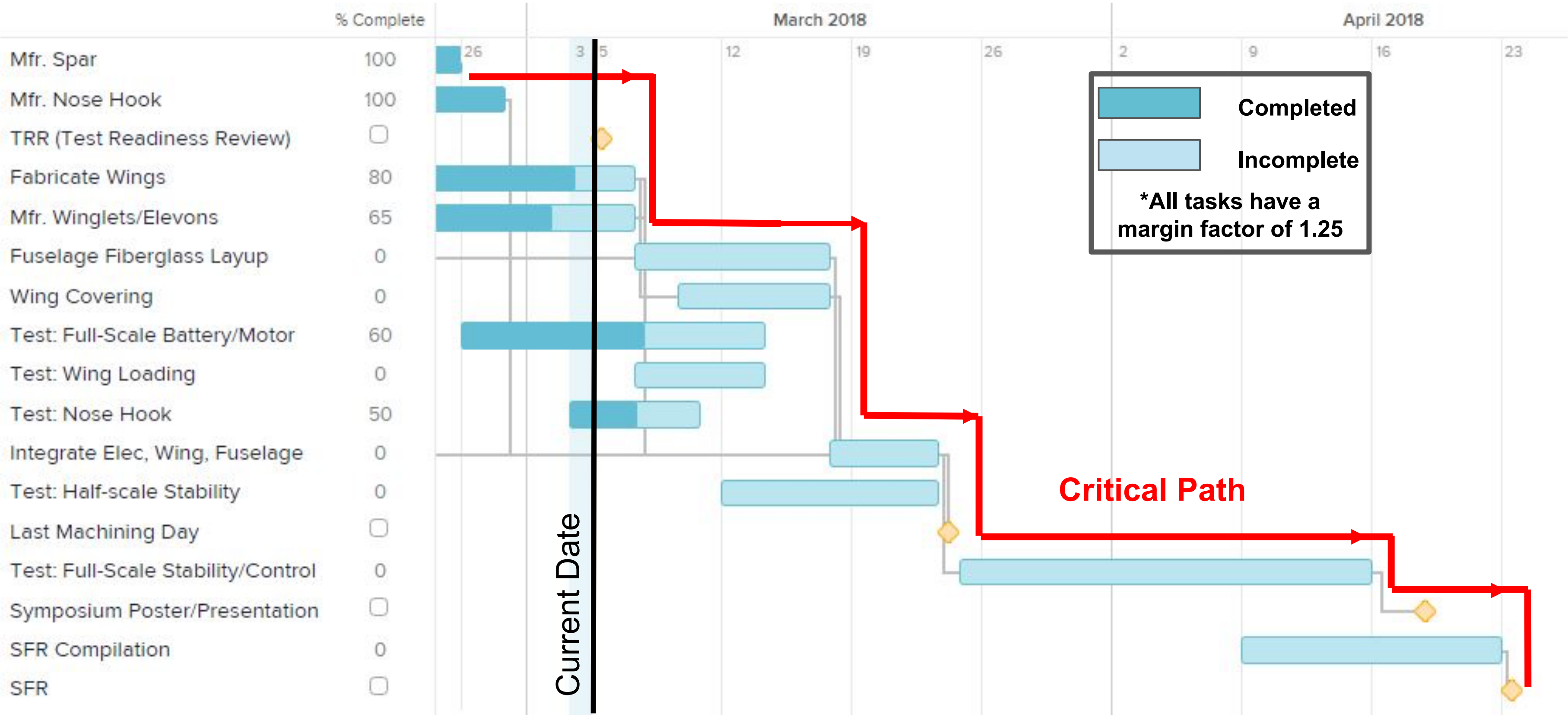
Critical Project Elements

CPE	Requirement Considerations
Aerial Vehicle	• **Stability and control** • **Future sensor payload** • **Tradeoff** between **maximizing lift-to-drag** ratio and **structural/manufacturing complexity**
Takeoff and Recovery	• Accelerate/decelerate aircraft under **maximum structural load** • Capability to transport and setup on **9.1m x 9.1m helipad**

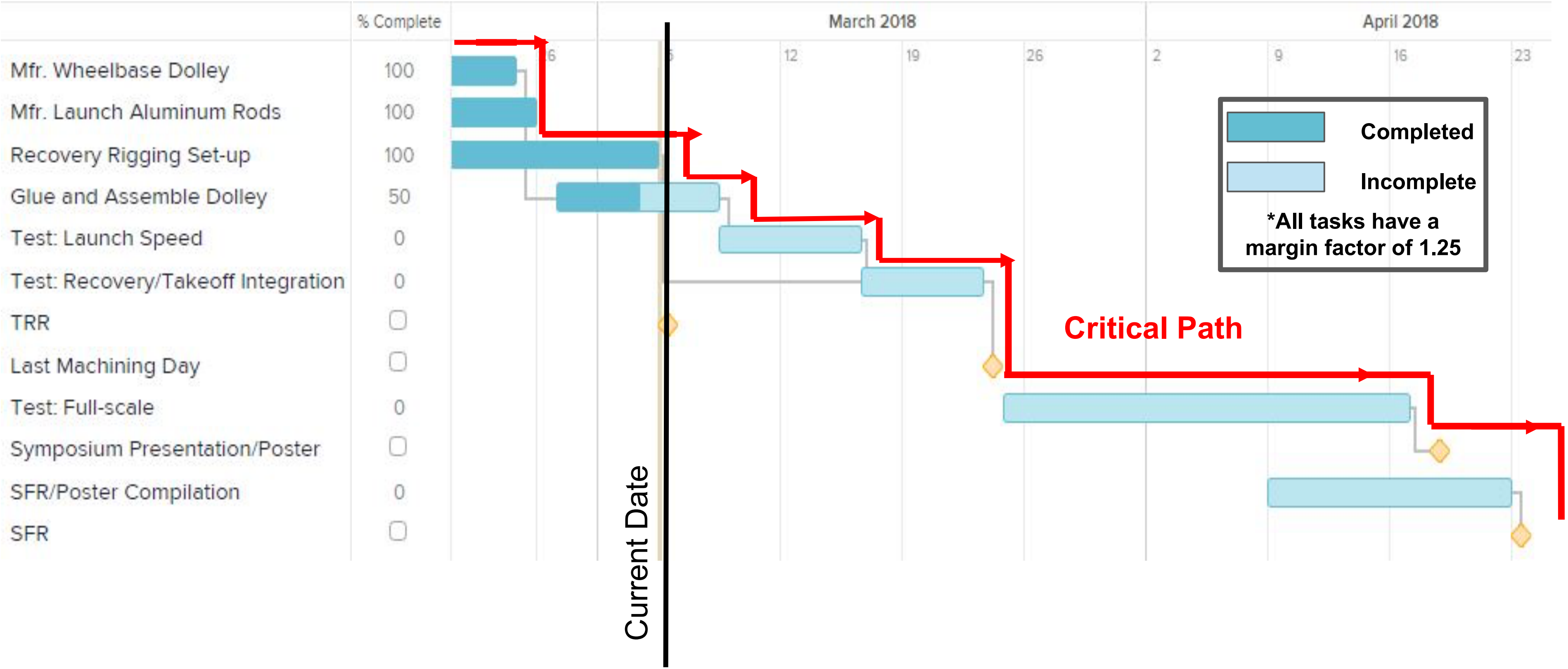
Critical Project Elements

CPE	Requirement Considerations
Communication with Ground Station	• **Communication range of 12 km** from ground station • Transmit images at **one per minute** • Piloted **manual control** • Transmit **updated flight waypoints** • Transmit **telemetry** to ground station
Flight Computer / Autopilot	• Collects **sensor data** for virtual cockpit • Autopilot keeps aircraft in **steady, level flight** • Accepts **flight waypoints** and **executes**

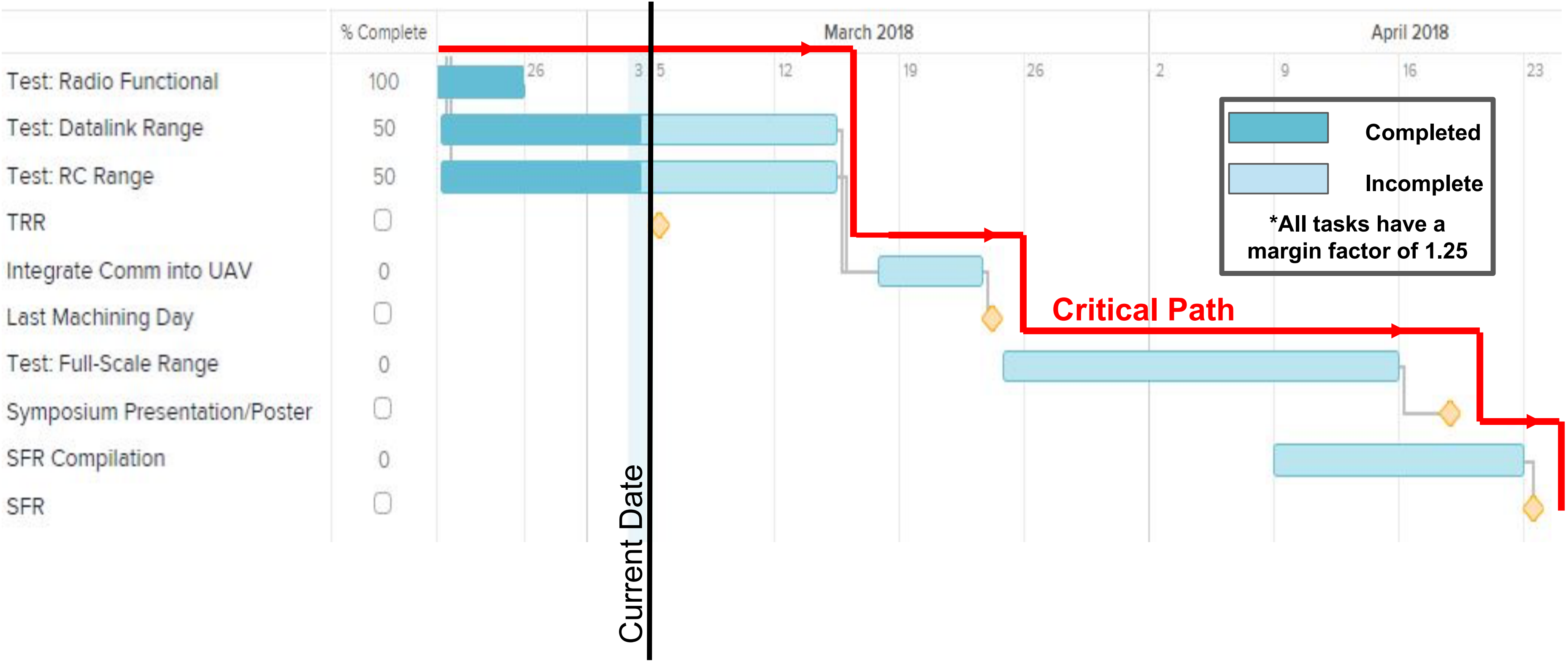
CPE: Aerial Vehicle Schedule



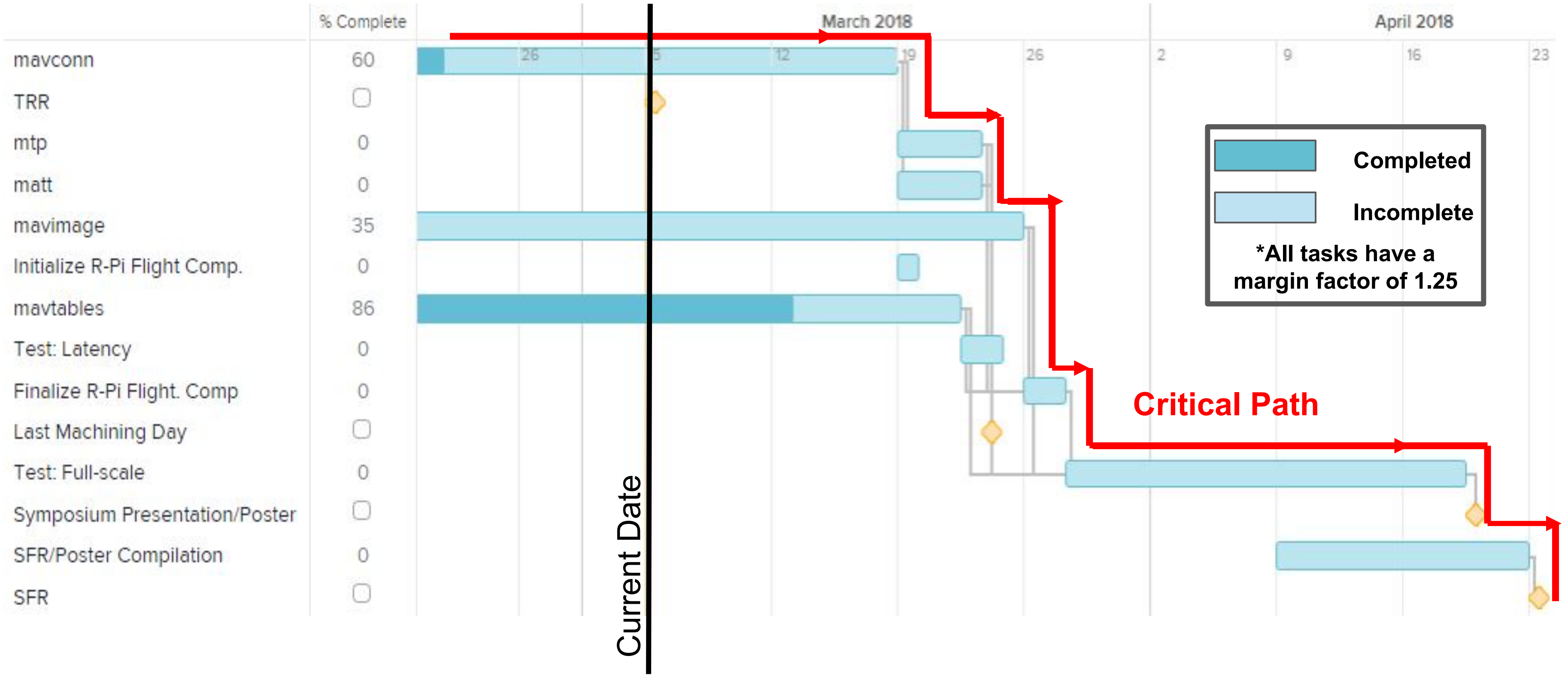
CPE: Takeoff and Recovery Schedule



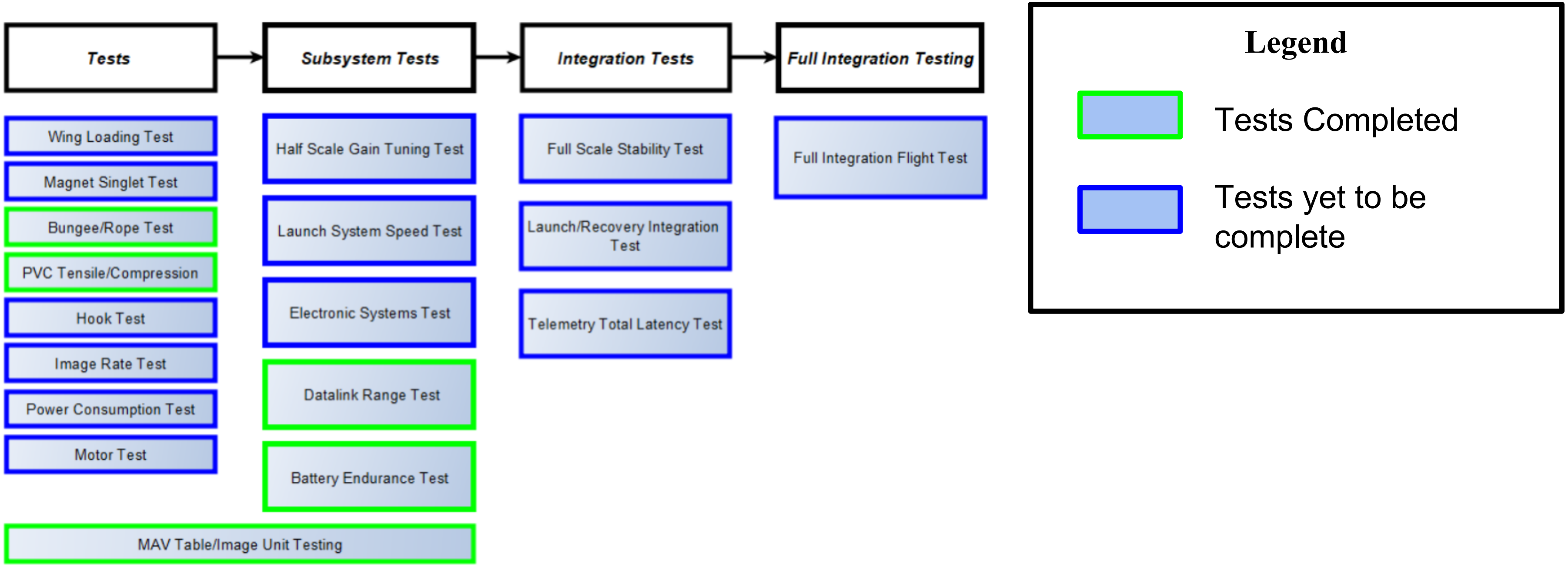
CPE: Radio Communication Schedule



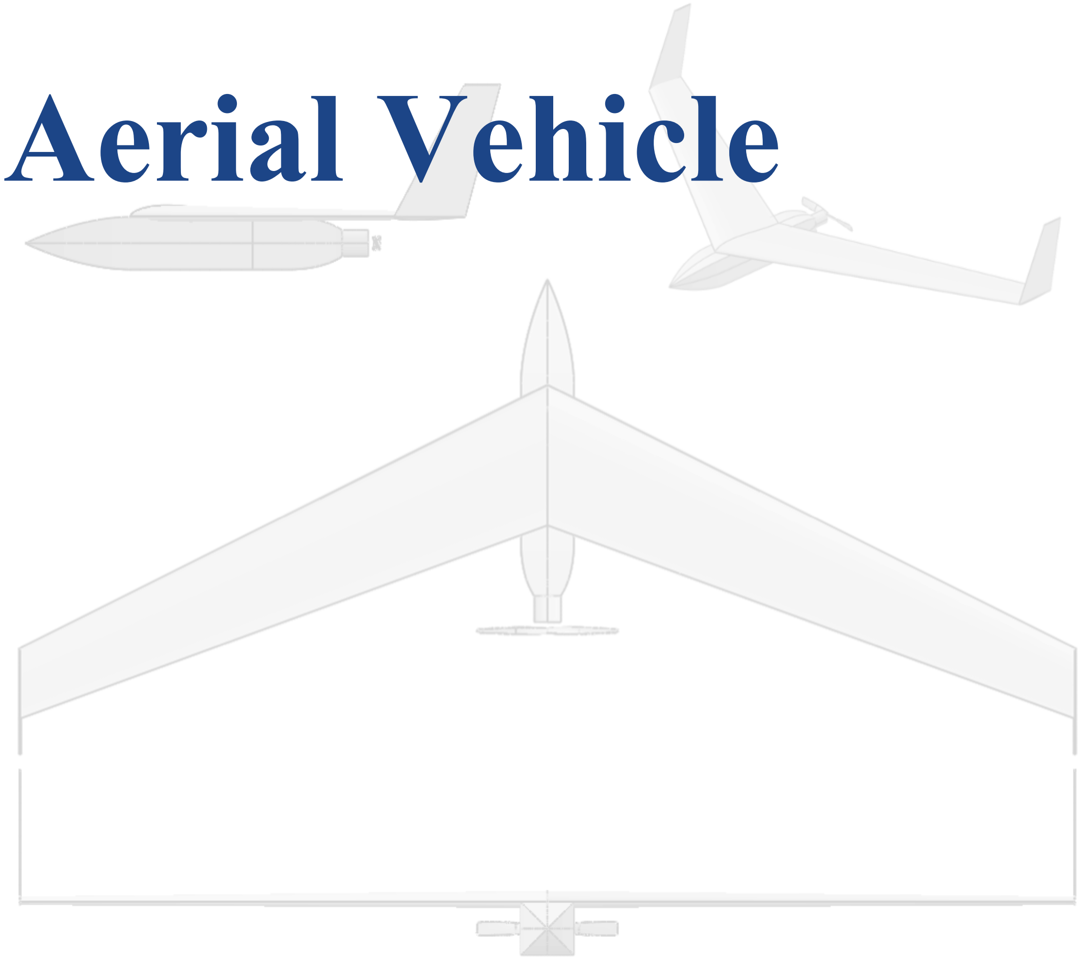
CPE: Software/Flight Computer Schedule



Test Plan


Tests
Subsystem Tests
Integration Tests
Full Integration Testing
Wing Loading Test
Magnet Singlet Test
Bungee/Rope Test
PVC Tensile/Compression
Hook Test
Image Rate Test
Power Consumption Test
Motor Test
Half Scale Gain Tuning Test
Launch System Speed Test
Electronic Systems Test
Datalink Range Test
Battery Endurance Test
Full Scale Stability Test
Launch/Recovery Integration Test
Telemetry Total Latency Test
Full Integration Flight Test
MAV Table/Image Unit Testing
Legend
Tests Completed
Tests yet to be complete

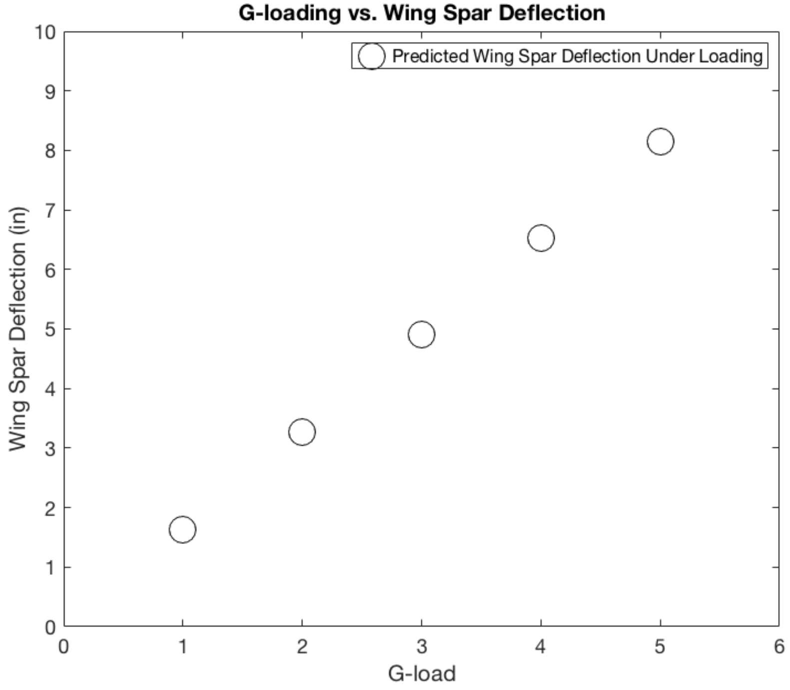
CPE: Aerial Vehicle



Wing Loading Test

Anticipated Date: March 8th **Expected Location:** Senior Projects Room

Requirement	The aircraft shall sustain **5 g forces upon landing**



Equipment	Expected Result According to Models
• Carbon fiber wing spar • Meter stick • ASEN 2001 Whiffletree setup • Vice	At 1 g loading: Carbon spar deflects 1.63 inches . . . At 5 g loading: carbon spar deflects 8.15 inches

Wing Loading Test Equipment

Test procedure
1. Secure test wing spar in vice. 2. Secure meter stick on adjacent table next to wingtip. 3. Position Wiffle Tree on spar. 4. Progressively and evenly load Wiffle Tree up to required 38 lb load.

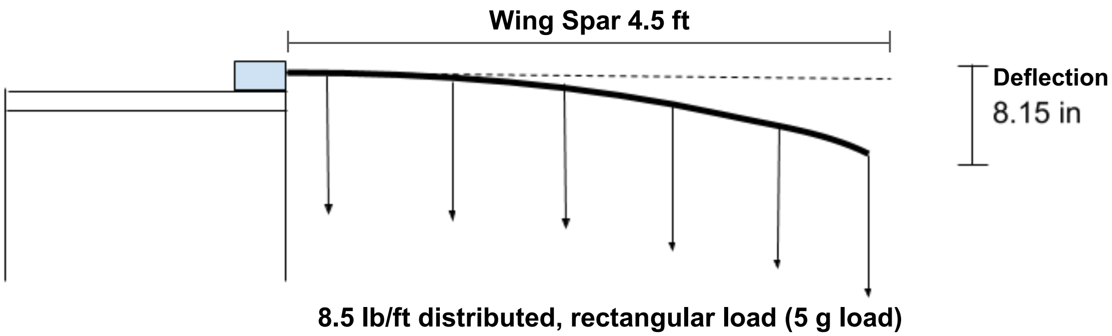


Diagram of Wing Loading Test Setup

Verification/Validation	Risk mitigation
Test validates **wing spar strength** and **loading requirement (5 g); deflection model**.	Ensures wing structural strength **withstands design limit loads** for flight.

Half-Scale Stability & In-Flight Gain Tuning Test

Anticipated Date: Week of March 12th **Expected Location:** Harlow Platts Park

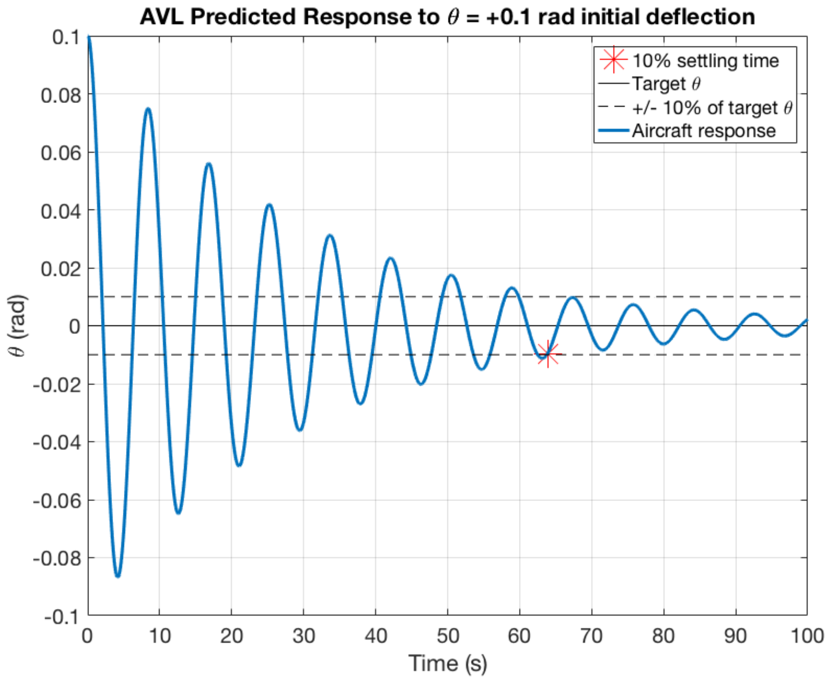
Requirement	The control system shall provide control surface deflections for aircraft **longitudinal and lateral stability** throughout all phases of flight.

Equipment

- Half-scale model
- Pixhawk 2.1
- RC radio
- RFD radio

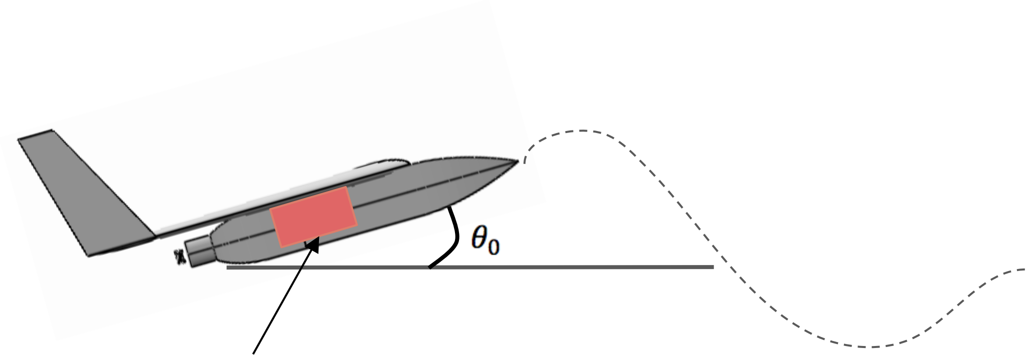
Expected Result According to Models

- **AVL model** predicts phugoid mode is **stable** with a **10% settling time** of approximately **1 minute**
- Tuned gains allow autonomous flight



Half-Scale Stability & In-Flight Gain Tuning Procedure

Test procedure
1. Launch aircraft 2. Perform manual flight maneuvers (PX4 procedure) 3. Remotely update control gains (PX4 procedure) 4. Pixhawk records flight parameters 5. Plot aircraft response 6. Compare with AVL model



Onboard: Pixhawk 2.1 with attitude sensors and radios

Verification/Validation	Risk mitigation
The recorded flight parameters will **verify** the **AVL predicted response** for all flight modes.	Half-scale **verifies stability models** used for both half-scale and full-scale. **Practice** for **in-flight gain tuning procedure**.

Battery Endurance Test

Anticipated Date: Week of January 17

Performed Date: February 19 @ RECUV Room

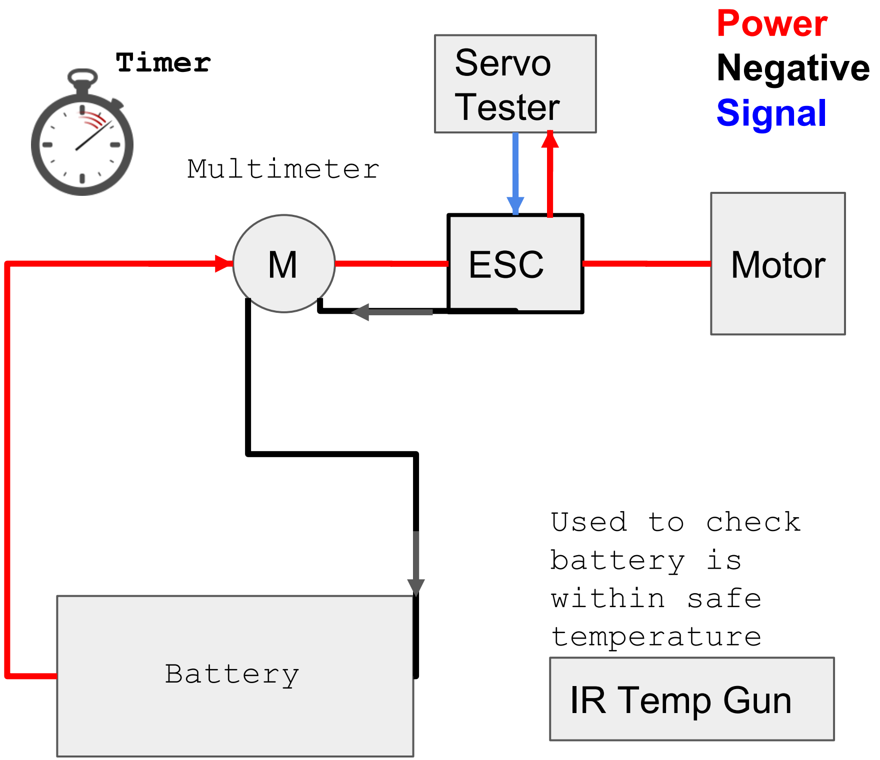
Requirement	Battery shall have **84 minute endurance.**

Motivation	**Expected Result According to Models**
• Validate Battery Capacity, • Reduces risk of: reduced range, loss of aircraft • Validates battery model	Battery holds charge for: 84 mins at 15A +/- 1 A.

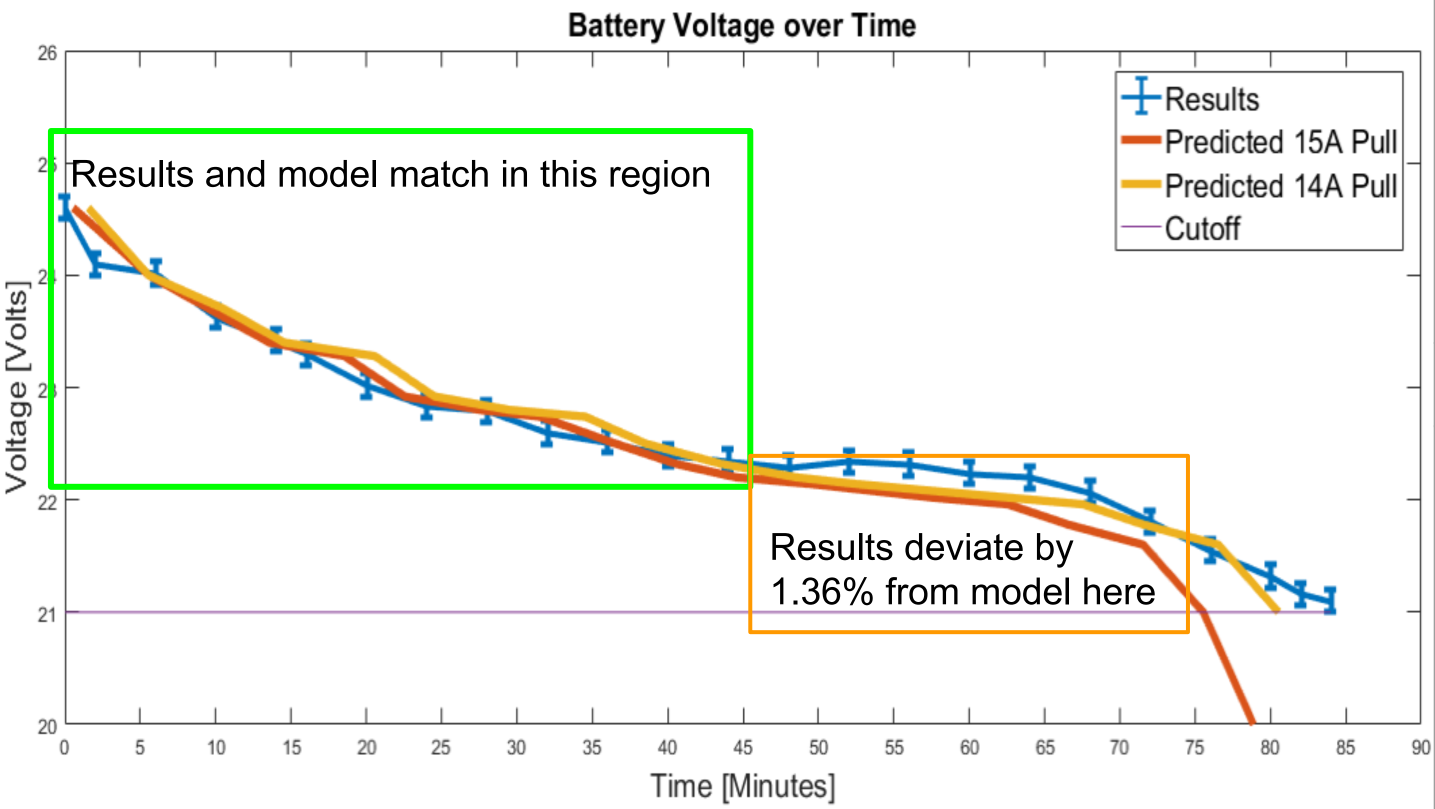
Procedure

1. Assemble Circuit
2. Turn on Motor until multimeter reads 15A
3. Record time from timer, voltage from multimeter, and maintain current at 15A +/- 1A every 4 minutes.
4. Stop when battery voltage reaches 21 V or when 84 minutes pass.

Test Setup



Battery Endurance Test Results



Error Sources

-Fluctuations in multimeter reading

-Multimeter voltage accuracy is within .01 V

-Multimeter error is 1%

Validation

- Deviation is within error within device.

- Result suggest valid battery model

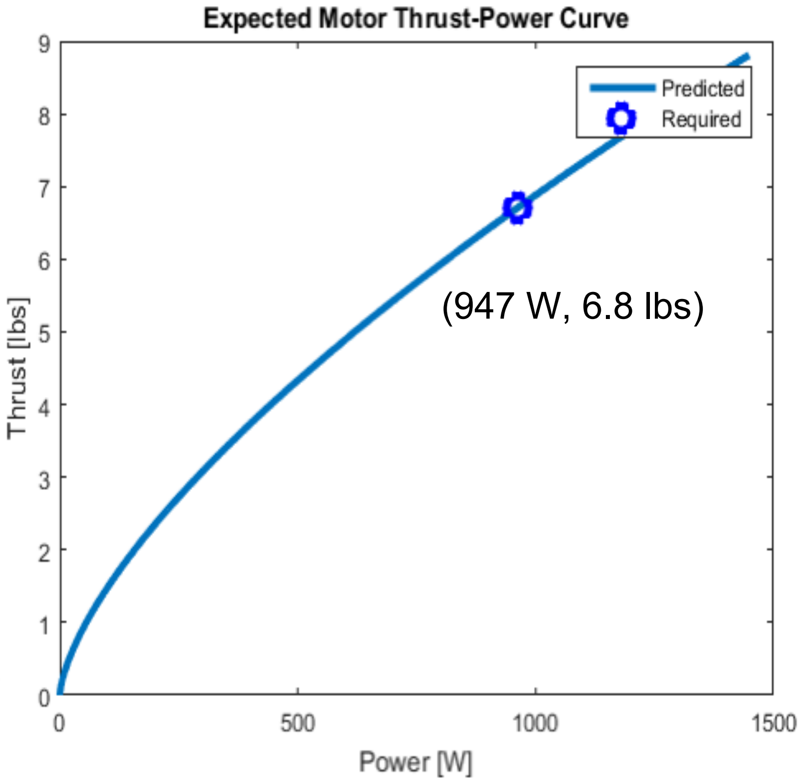
- Second test for repeatability and further validation of **84 minute endurance**

Motor Thrust Test

Expected Date: Week of March 5 in RECUV Room

Requirement	Aircraft shall **climb at 5 m/s**

Motivation	Expected Result According to Models
-Validate Motor Capability -Motor pulls expected power -Characterize thrust vs. power	-Motor can achieve 7.6 lbs of thrust at 1200 W -Meets needed 6.8 lbs at 947 W.



Motor Capability Model

$$T = C_t \rho n^2 D^3 \text{ lbf} \qquad P = 745.7\left(\frac{C_p \rho n^3 D^5}{550}\right) \text{ W}$$

Motor Thrust Test Procedure

Procedure

1. Assemble circuit
2. Step up motor by 0.2 lbs up to 7.0 lbs, record thrust, voltage, and amperage along the way.
3. Multiply amperage and voltage for power
4. Plot thrust and power against predicted model.
5. Check result curve at 6.8 lbs.

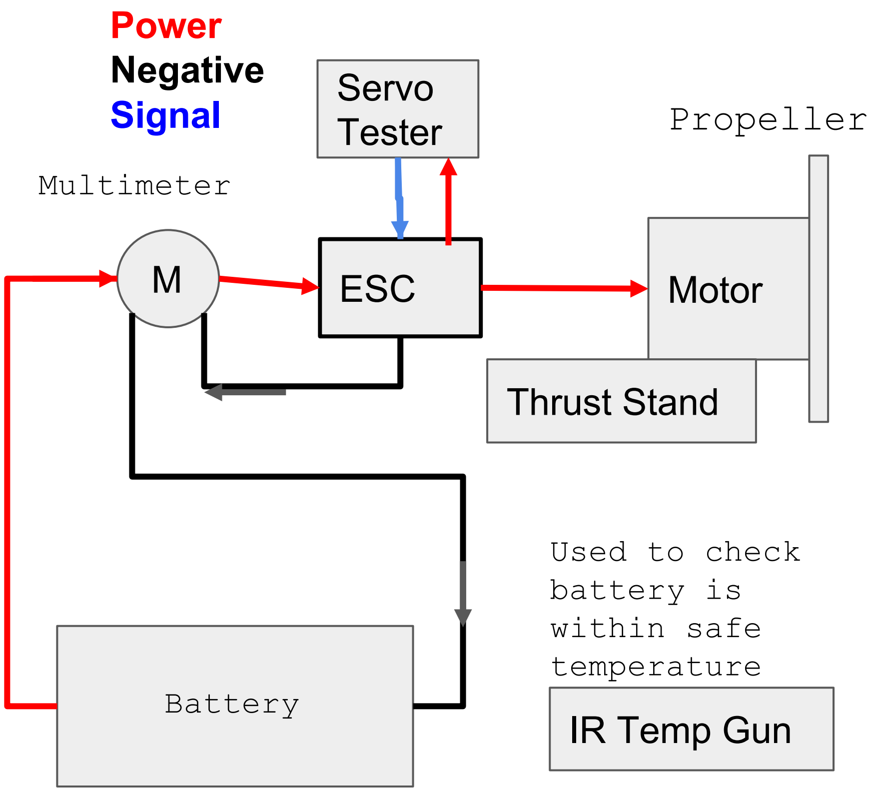
Verification and Validation

Plot results and model for alignment.

Risk Mitigation

Confirming characterization reduces risk of a reduced range, reduced climb rate, loss of aircraft.

Test Setup

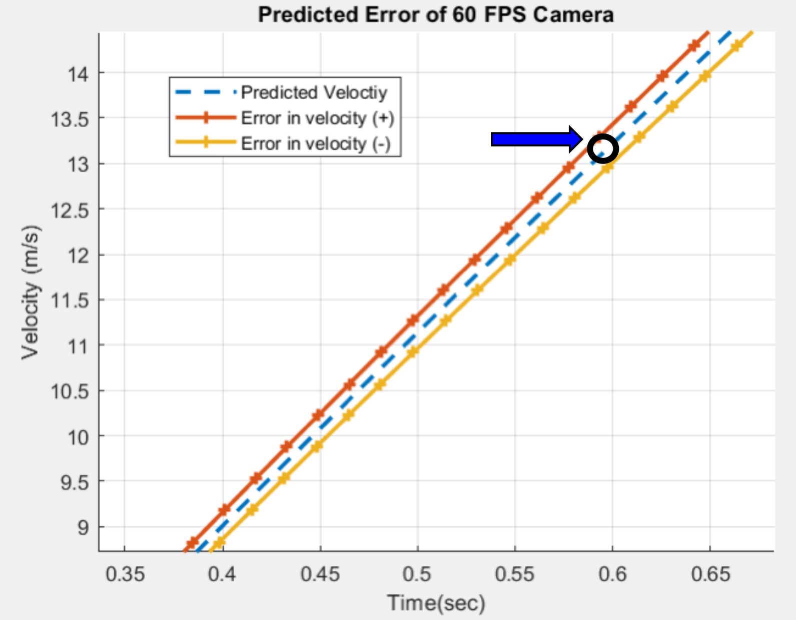


CPE: Launch/Recovery

Launch System Speed Test

Anticipated Date: March 8th **Expected Location:** Business Field

Requirement	The launch system shall **accelerate the UAV to 13.2 m/s** by the end of ramp.



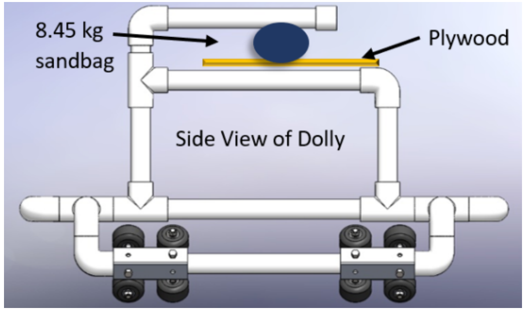
Equipment

- Launch system
- 60 FPS camera
- Logger Pro software
- Dummy weight (8.45 kg)

Expected Result According to Models

Dolly/Dummy mass system will reach **13.2 m/s at the end of the ramp** measured by **logger pro software**

Launch Speed Test Procedure

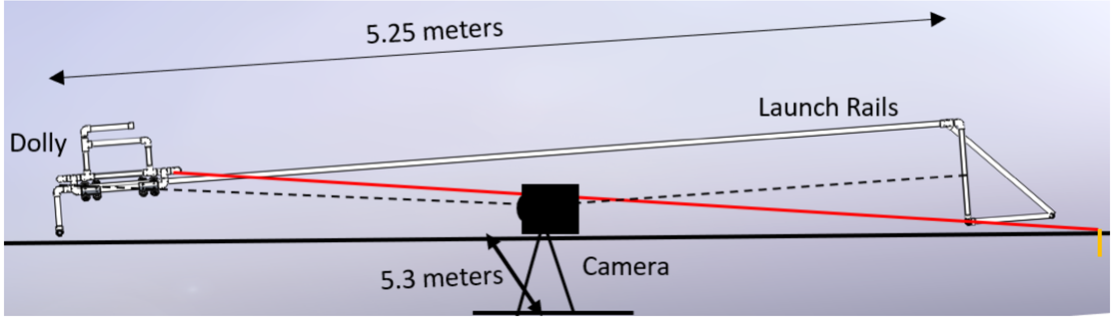


Test Procedure

1. Assemble Launch System
2. Insert sandbag
3. Pull back dolly system
4. Begin recording
5. Release dolly
6. Collect footage

Test Setup

- ❖ Dolly/Sandbag system
- ❖ Ramp System



Verification/Validation

The software will validate the UAV will reach 13.2m/s confirming the **MATLAB PE to KE launch model**

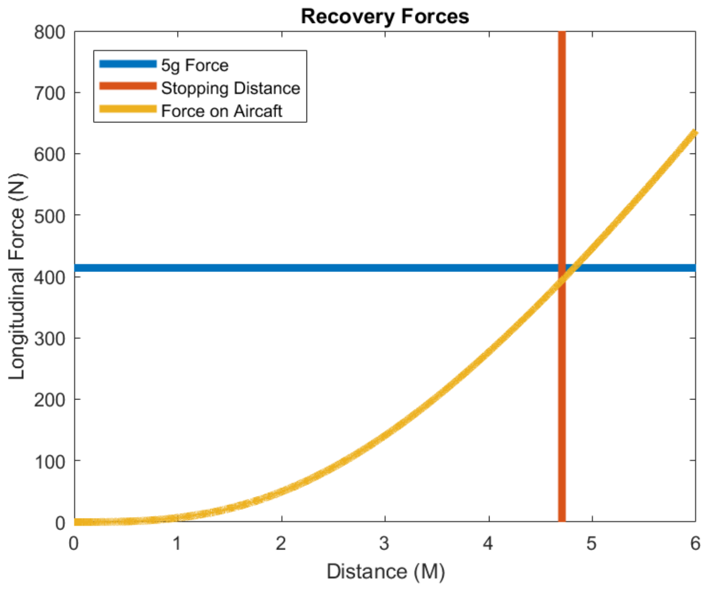
Risk mitigation

Launching **dummy mass** is **expendable**. **Prevents** full scale UAV crash.

Launch/Recovery System Integration Test

Anticipated Date: March 16th **Expected Location:** Business Field

Requirement	The recovery/launch system shall **exert forces** on the aircraft **under 5 g**.



Equipment

- Launch and Recovery Systems
- Dummy weight (8.45 kg)
- 120 FPS camera

Expected Result According to Models

Accelerometer on dummy mass will experience **less than 5 g** for launch and recovery

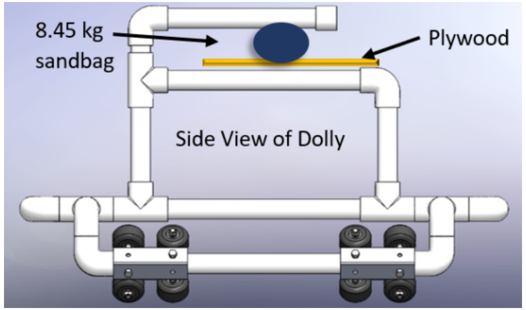
Launch/Recovery System Integration Test

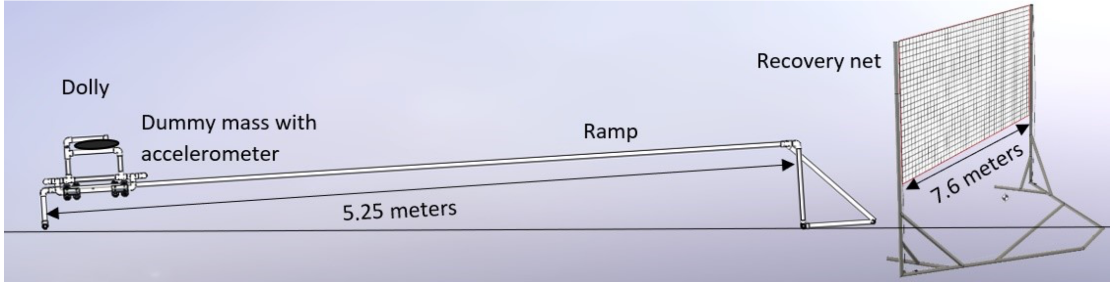
Test Procedure

1. Assemble Launch and Recovery Systems
2. Insert sandbag
3. Pull back dolly system
4. Begin recording
5. Release dolly
6. Collect data/footage

Test Setup

- ❖ Dolly/Sandbag system
- ❖ Ramp/Recovery System





Verification/Validation

Accelerometer data verifies UAV experiences under 5g during launch/recovery.

Risk mitigation

Launch/recovery **dummy mass** is **expendable**. **Prevents** full scale UAV crash.

CPE: Communications

Datalink Range Test

Requirement	Telemetry radio shall have a **range** of **12 km at 90+ kbps**.

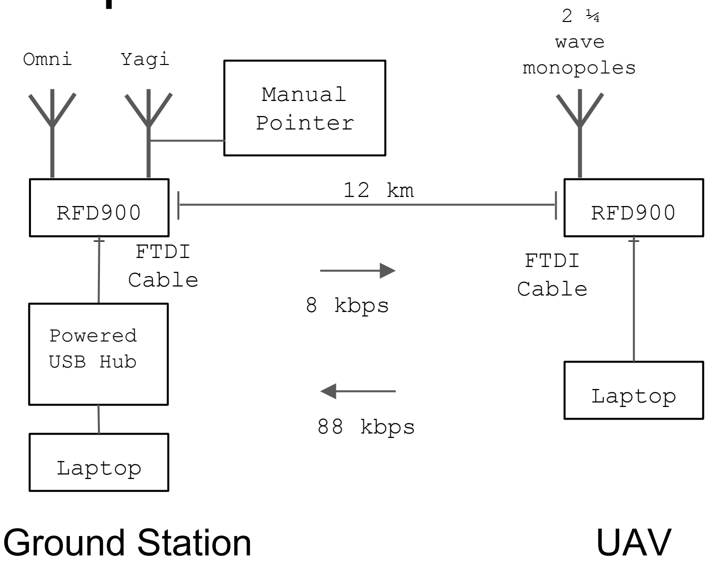
Test	Date	Location	Distance	Result
1	Feb 25	NCAR & Davidson Mesa	9.12 km	Link Established Software failure
2	Feb 28	Bear Peak & Open Space	12.06 km	Failed
3	March 2	NCAR & Davidson Mesa	7.44 km	Link established Max RSSI of 127

Equipment	**Expected Result According to Models**	**Risk Mitigation**
• Ground Station • UAV Datalink radio • UAV Antennas • Laptop (UAV PS)	Secure link with 3.59 dB margin (Target RSSI of >150)	Verify communication at 12 km to avoid losing communication with UAV

Datalink Range Test Setup

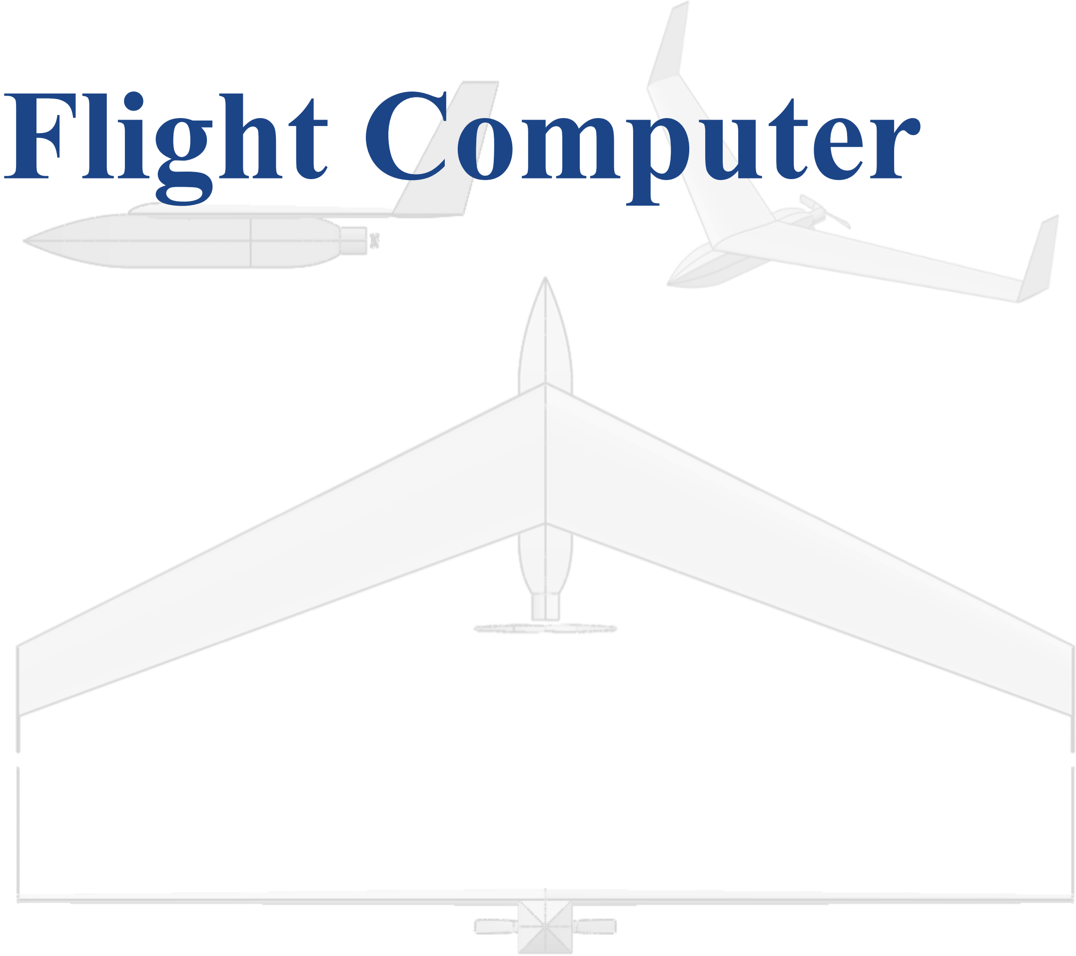
Test procedure

1. Team 1 hikes to the top of Bear Peak
2. Team 2 goes to point 12 km away in Boulder Open Space
3. Point antennas towards other team
4. Attempt to establish connection



Verification/ Validation	Experimental Results	Next Step
Verify the system can communicate at required range of 12 km	• Failed to establish adequate link • Dysfunctional directional antenna	• Upgrade directional antenna

CPE: Flight Computer



Software: Continuous Integration

Travis CI COVERALLS

Motivation	Procedure
Ensure software is: ● Functional ● Error free	1. **Commit changes** to main repository. 2. **Build and test software** on supported operating systems and compiler/interpreter combinations with Travis CI. 3. **Compute the test coverage percentage** with coveralls.io. 4. **Automatically mark pull requests** with ✔ or ✖ to **indicate the testing results** of the requested changes. 5. **Badges indicating current test status** of master branch are embedded into the online README files.

Component	Status (from CI)	# Test Cases
MAVLink Interface Library	build passing coverage 99%	6
Image Capture/Transmission	build passing coverage 70%	19
MAVLink Router/Firewall	build passing coverage 98%	837

Telemetry Total Latency Test

Anticipated Date: March 20th

Requirement	Aircraft shall **transmit telemetry** to ground station with **less than 200 ms latency**.

Motivation
Must **verify required latency** before the UAV can be flown **beyond line of sight**.

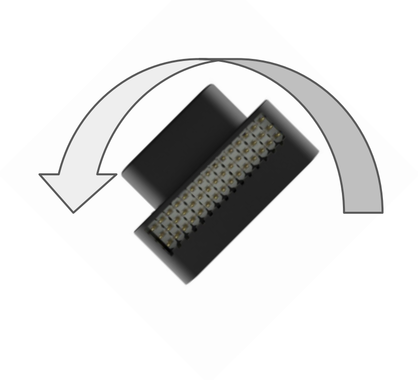
Equipment & Software	Expected Result According to Models
• Telemetry Radios • Autopilot & Flight Computer • 60 FPS Camera (17ms res) • mavtables • Ground Control Software	Latency of 160 ms for largest packet (267 bytes) based on worst case analysis.

Telemetry Total Latency Test

Anticipated Date: March 20th

Test procedure
1. **Assemble communication system** from autopilot to ground station. 2. **Record** autopilot and ground control software in same frame. 3. **Rotate autopilot ~90 degrees**. 4. Collect footage.

Pixhawk

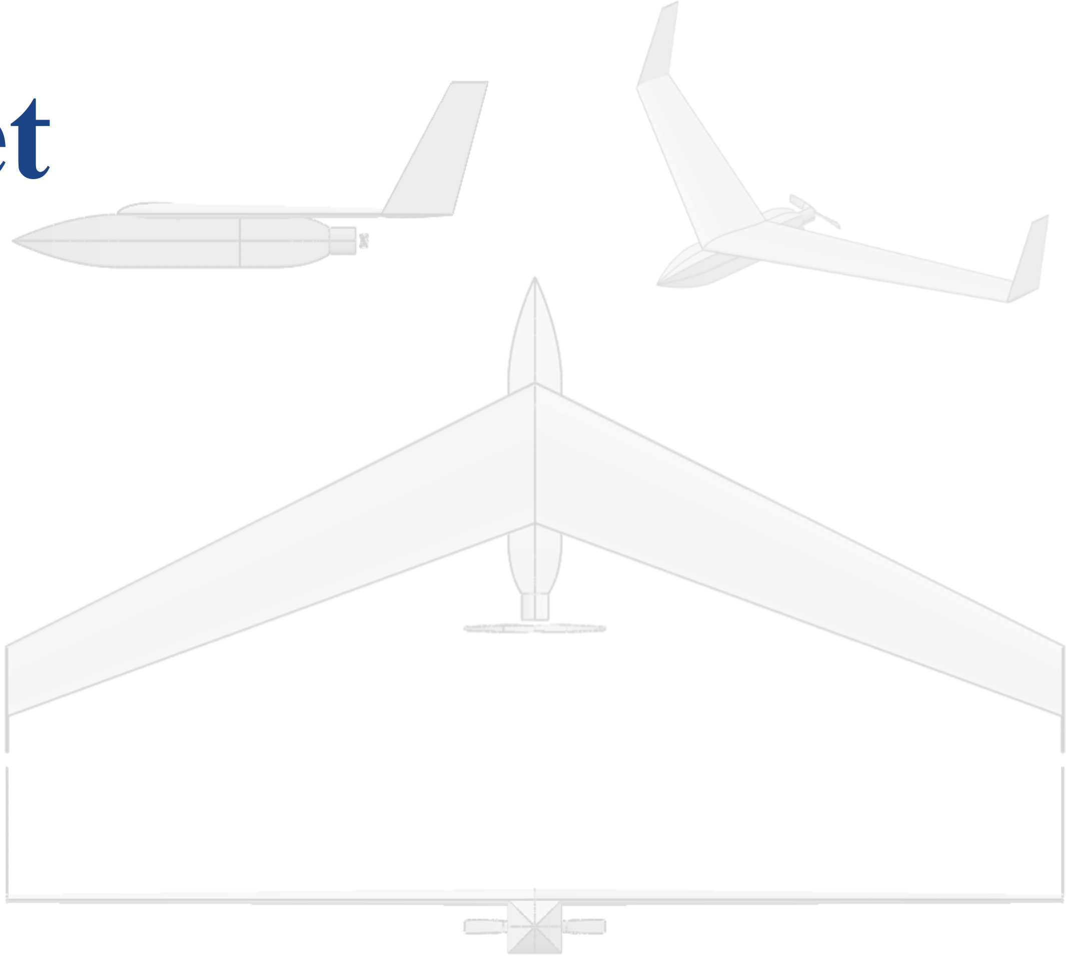


Comm latency

QGroundControl

Verification/Validation	Risk mitigation	Next Step
Verify requirement: rotation indicated on virtual cockpit within **200ms** of rotation of autopilot.	Need under 200ms latency for **safe operation** of the aircraft **beyond line of sight.**	If measurement is within 17ms (camera margin) of 200ms then a higher frame rate camera will be required.

Budget



SHAMU Budget

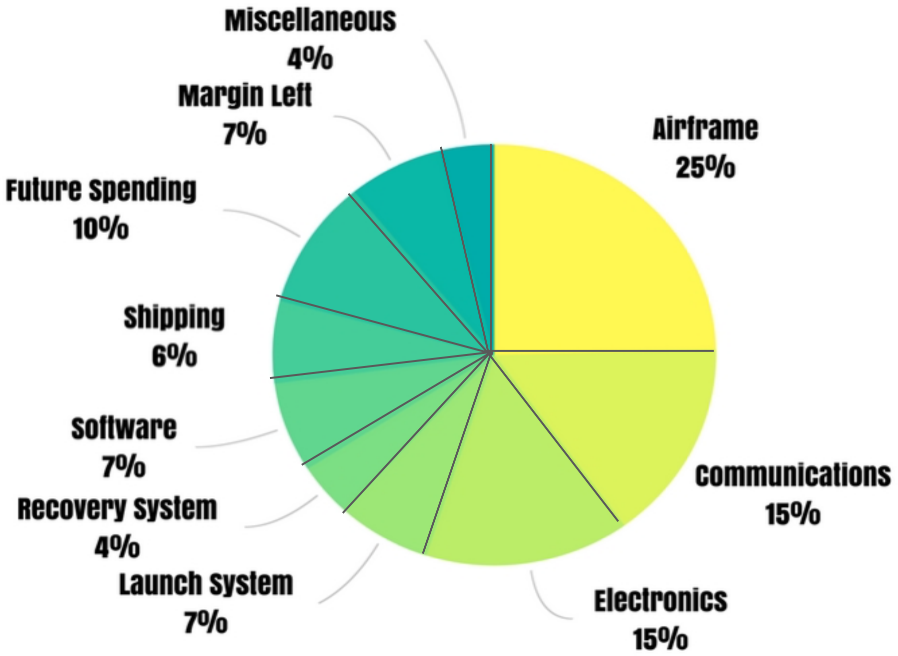
Total Spent: $4703.40

98% of Parts Received

99% Parts Ordered

Remaining Purchases	Cost
Poster/Printing	~$200

Predicted Margin
~$100*

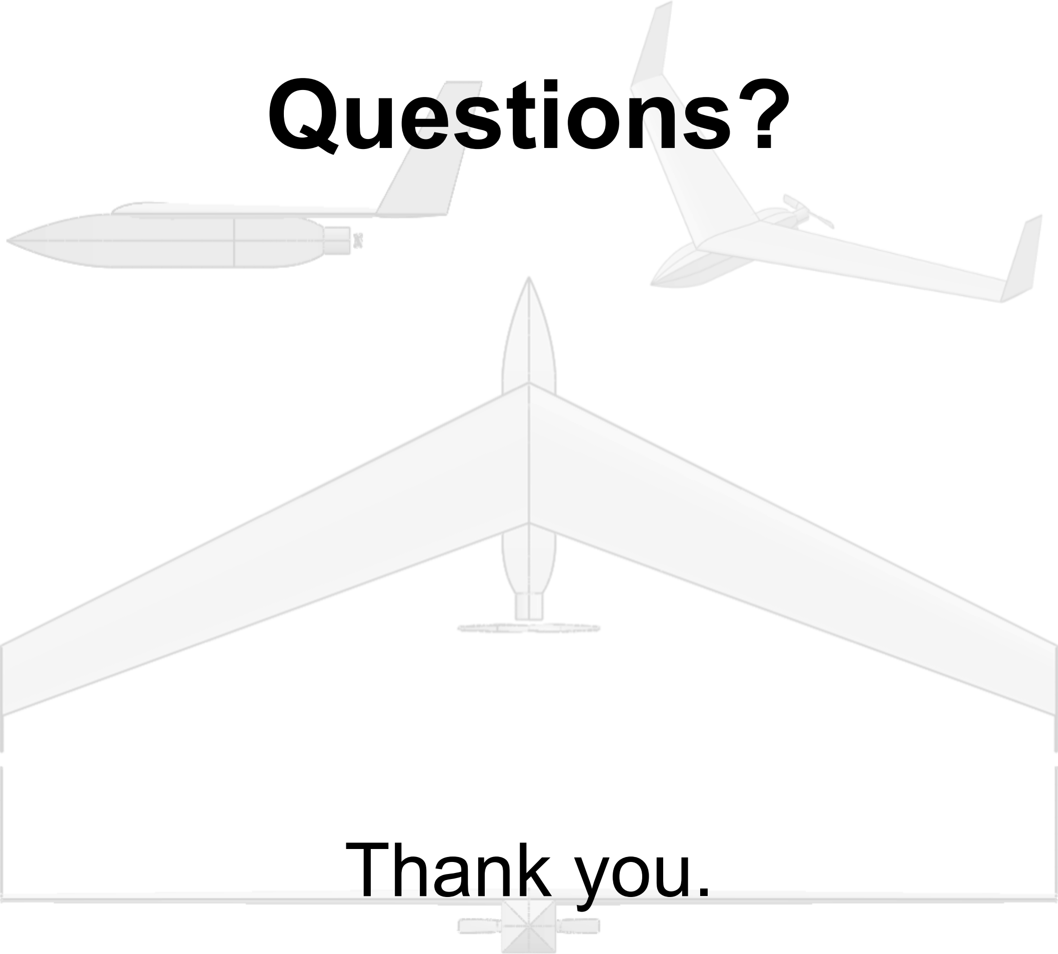


*Out of $5,000. Crowdfunding currently grants addtl. $2,259 margin.

Acknowledgements

The SHAMU team would like to thank our crowdfunding supporters, Dr. Gerren, Dr. Koster, Dr. Lawrence, Trudy Schwartz, Bobby Hodgkinson, Matt Rhode, PAB, Tim Kiley, Lee Huynh, Mr. Norman, Trimble Inc., James Nestor, and David Gruber.

Questions?



Thank you.

Navigation

Backup Slides

PVC Tensile Test

Requirement	Capture system shall sustain 5 g aircraft recovery forces.

Anticipated Date: **COMPLETED**

Motivation	Expected Result According to Models	Off-Ramp
Verify PVC structure will be able to sustain landing forces	Accelerometer on dummy mass will experience < 5g	Change pipe size/type

Equipment	Availability	Capabilities	Requirements	Satisfied?
Instron Machine	ITLL	Range: 0-1100 lbf	300 lbf	Available ✔

Results: Average Failure Stress 46.9 kPa >14.5 kPa expected bending stress, 3.25 factor of safety

Unit Test Sample

packets for testing

mock Filter class

mock Socket class

```
TEST_CASE("UDPInterace's 'send_packet' method.", "[UPDInterface]")
{
    // MAVLink packets.
    auto ping = std::make_shared<packet_v2::Packet>(to_vector(PingV2()));
    auto heartbeat =
        std::make_shared<packet_v2::Packet>(to_vector(HeartbeatV2()));
    auto encapsulated_data =
        std::make_shared<packet_v2::Packet>(to_vector(EncapsulatedDataV2()));
    // Filter
    fakeit::Mock<Filter> mock_filter;
    fakeit::When(Method(mock_filter, will_accept)).AlwaysDo(
        [&](auto & a, auto & b)
    {
        (void)a;
        (void)b;
        return std::pair<bool, int>(true, 0);
    });
    auto filter = mock_shared(mock_filter);
    // Socket
    std::multiset<std::vector<uint8_t>> send_bytes;
    std::multiset<IPAddress> send_addresses;
    using receive_type =
        IPAddress(std::back_insert_iterator<std::vector<uint8_t>>,
                  const std::chrono::nanoseconds &);
    using send_type =
        void(std::vector<uint8_t>::const_iterator,
             std::vector<uint8_t>::const_iterator,
             const IPAddress &);
    UDPSocket udp_socket;
    fakeit::Mock<UDPSocket> mock_socket(udp_socket);
    fakeit::When(OverloadedMethod(mock_socket, send, send_type)
                ).AlwaysDo([&](auto a, auto b, auto c)
    {
        std::vector<uint8_t> vec;
        std::copy(a, b, std::back_inserter(vec));
        send_bytes.insert(vec);
```

Unit Test Sample

mock Socket class

spy on ConnectionFactory class

construct UDPInterface

mock will_accept method

mock receive method

```cpp
        send_addresses.insert(c);
    });
    auto socket = mock_unique(mock_socket);
    // Connection Factory
    ConnectionFactory<> factory_obj(filter);
    fakeit::Mock<ConnectionFactory<>> spy_factory(factory_obj);
    fakeit::Spy(Method(spy_factory, wait_for_packet));
    // Interface
    UDPInterface udp(
        std::move(socket),
        std::make_shared<ConnectionPool>(),
        mock_unique(spy_factory));
    std::chrono::nanoseconds timeout = 1ms;
    SECTION("Multiple connections with targeted packet.")
    {
        // Mocks
        fakeit::When(Method(mock_filter, will_accept)).AlwaysDo(
            [&](auto & a, auto & b)
        {
            (void)b;

            if (a.name() == "PING")
            {
                return std::pair<bool, int>(true, 0);
            }

            return std::pair<bool, int>(false, 0);
        });
        fakeit::When(OverloadedMethod(mock_socket, receive, receive_type)
                    ).Do([](auto a, auto b)
        {
            (void)b;
            auto vec = to_vector(HeartbeatV2());
            std::copy(vec.begin(), vec.end(), a);
            return IPAddress("127.0.0.1:4000");
        }).Do([](auto a, auto b)
```

Unit Test Sample

mock receive method

actual test

mid test verifications

continue test

final test verifications

```cpp
        }).Do([](auto a, auto b)
        {
            (void)b;
            auto vec = to_vector(EncapsulatedDataV2());
            std::copy(vec.begin(), vec.end(), a);
            return IPAddress("127.0.0.1:4001");
        }).Do([](auto a, auto b)
        {
            (void)b;
            auto vec = to_vector(PingV2());
            std::copy(vec.begin(), vec.end(), a);
            return IPAddress("127.0.0.1:4002");
        });
        // Test
        udp.receive_packet(timeout);
        udp.receive_packet(timeout);
        udp.receive_packet(timeout);
        udp.send_packet(timeout);
        // Verification
        fakeit::Verify(Method(spy_factory, wait_for_packet).Using(1ms)).Once();
        fakeit::Verify(Method(spy_factory, wait_for_packet)).Once();
        fakeit::Verify(
            OverloadedMethod(mock_socket, send, send_type)).Once();
        REQUIRE(send_bytes.size() == 1);
        REQUIRE(send_bytes.count(to_vector(PingV2())) == 1);
        REQUIRE(send_addresses.count(IPAddress("127.0.0.1:4000")) == 1);
        // Test
        udp.send_packet(timeout);
        // Verification (no futher operations)
        fakeit::Verify(
            Method(spy_factory, wait_for_packet).Using(1ms)).Exactly(2);
        fakeit::Verify(Method(spy_factory, wait_for_packet)).Exactly(2);
        fakeit::Verify(
            OverloadedMethod(mock_socket, send, send_type)).Once();
    }
}
```

DR 3.6 - Image Rate Test

Anticipated Date: March 24th

Requirement	Aircraft shall capture and transmit 1920x1080 images to ground station at 1/60 Hz.

Equipment & Software	Expected Result According to Models
Concurrent to full system integration and flight test. (no extra equipment or setup)	Images will be received by ground station at a rate of at least 1/60 Hz.

Verification/Validation
Verify timestamps of images received at the ground station to be within 1 minute of each other.

Tests Completed for Launch/Recovery

Test	Motivation	Expected Result According to Models	Results
Bungee Test	Verify the spring constant of the bungee from launch/recovery models	Bungee has a spring constant of K = 86 N/m and does not snap under expected loads	Bungee tensile strength exceeded 68.5N with a safety factor of 2.5. Bungee slipped out of the grips at this point.