CASE STUDY: Test Asset Protection Logic

Protecting Seven-Figure programs and Six-Figure Test Articles Through Smart Emergency Shutdown Design

Project Snapshot

Industry: Aerospace / Combustion Testing
Challenge: High-value combustion test articles (hundreds of thousands of dollars each) were exposed to multiple failure scenarios with potential for catastrophic damage.
Result: A structured, automated protection logic that sequences dozens of control elements during emergencies, dramatically improving safety, preserving assets, and stabilizing test operations.

---

1. Opening Hook – The Challenge

In advanced combustion testing, the real challenge is not just running a test; it is **protecting a very expensive test article** every second the flame is on. A single adverse event can melt metal, crack components, or destroy a unique prototype in minutes.

The lab environment is full of what-ifs: loss of air, control-valve failure, acoustic overload, fuel spikes, or quench-system faults. The question we set out to answer was: **How do we guarantee that when something abnormal happens, the test article survives undamaged?**

---

2. Problem Definition

We were working with a complex combustion rig whose safe operation depended on many interacting subsystems, including:

- Multiple **fuel valves** (main, pilot, staging, recirculation)
- A **high-pressure fuel pump** with VFD control
- **Metering valves** on the pumping skid
- **Fuel isolation valves**
- **Nitrogen purge solenoids** and their set pressure
- **Back-pressure valve** setpoints
- Additional air and auxiliary controls

vidually, each control element was understandable. Together, they formed a **potentially otic emergency environment** if something went wrong. The core problem was the absence of a **clear, unified “test asset protection logic”**:

- Which devices should move,
- To what state,
- In what **priority and sequence**,
- Triggered by which **emergency conditions**.

---

3. Our Approach – From Complexity to Clear Logic

We tackled the problem in two major stages.

Stage 1 – Define Safe States and Priorities

First, we **enumerated every relevant control element** and defined its correct “safe state” in an emergency. For example:

- Fuel valves: which ones must **close immediately**, which can ramp down.
- Fuel pump and VFD: how quickly to **de-rate or stop** the pump.
- Metering valves: how to back out to avoid surges.
- Nitrogen purge system: when to **open purge solenoids** and at what relative pressure.
- Back-pressure valve: how to move to avoid over- or under-pressurizing the system.

Equally critical, we established the **execution priority and timing** of each action. Some responses must be **instantaneous** (hard isolation of fuel), while others should follow in a controlled sequence (purge, depressurization, auxiliary shutdowns).

Once these target states and priorities were clearly defined, they could be encoded into the test controller as a **single, coherent emergency shutdown sequence** rather than a patchwork of independent trips.

Stage 2 – Define the Trigger Scenarios

Next, we identified and formalized the **scenarios that should invoke this protection logic**, including:

- **Acoustic limits exceeded**: When dynamic pressure amplitudes surpass what the mechanical design can tolerate, risking structural cracking.
- **Fuel-pressure or flame-temperature spikes**: Indicating risk of thermal overload beyond material ratings.

- **Loss of quench or cooling systems**: Threatening turbo-machinery or hot-section hardware.
- **Loss or malfunction of a key control valve**: Creating uncontrolled or unbalanced fueling.
- **Loss of air**: Effectively similar to a fuel spike, as flame speed can exceed flow velocity when air drops suddenly.

For each scenario, we tied a **clear detection rule** to the same asset-protection sequence, so that any qualifying event would reliably trigger the logic and safely shut the system down.

---

4. Results and Impact

Implementing this protection logic had a **huge impact** on safety, cost, and test reliability:

- **Safety:**
  - Systematic handling of worst-case events dramatically reduced the risk of flame-induced damage.
  - The test article remained within its safe operating envelope, even when multiple parameters went wrong simultaneously.
- **Cost avoidance:**
  - Each protected test article represented **hundreds of thousands of dollars** in hardware, plus weeks or months of schedule.
  - The protection logic effectively **insured** these assets by preventing failures that could have destroyed them in a single incident.
- **Operational impact on testing:**
  - Engineers gained confidence to explore more aggressive or advanced test conditions, knowing the system would **react correctly in emergencies**.
  - Instead of catastrophic failures, unexpected events became **controlled shutdowns**, followed by root-cause analysis and safe restarts.
  - Downtime associated with damage repair, hardware replacement, and re-qualification was greatly reduced.

---

5. Takeaway & Forward Value

This project illustrates how we approach **high-risk combustion testing**:

- We start by **mapping every control element** and defining its safe state.
- We **codify emergency scenarios** into clear, programmable logic.

We implement an asset-protection strategy that turns “chaotic failure” into **predictable, safe shutdown**.

The result is a test environment where **six-figure assets are protected by design**, not by luck—enabling more ambitious testing, fewer expensive surprises, and a step-change in how safely and confidently the lab can operate.