

GN&C Fault Protection Fundamentals

Robert D. Rasmussen

Topics

⊕ Covered

- ★ Fault protection
- ★ Related system control functions

⊕ Not covered

- ★ Identifying hazards and evaluating risks
- ★ Redundancy, cross-strapping, and fault containment
- ★ Sensor selection and placement
- ★ Verification processes and methods
- ★ Preventative maintenance
- ★ etc.

Why GN&C FP Fundamentals?

- ⊕ A recurring issue
- ⊕ A growing challenge
- ⊕ A warning of things to come

- ⊕ To do the next hard thing we have to get fault protection fundamentals right

What's Fundamental?

*“A designer knows
he has arrived at perfection
not when there is
no longer anything to add
but when there is
no longer anything to take away.”*

Antoine De Saint-Exupery (1900-1944)
French Aviator, Writer

Simplicity

- ⊕ Not the fewest number of elements
- ⊕ Rather a few essential *concepts* connected in a few recurring *patterns* adhering to a few key *principles*
- ⊕ Something you can understand, analyze, and build with confidence

Foundations of Good Architecture

- ⊕ Basic concepts
- ⊕ Regular patterns of design
- ⊕ Well-founded principles

... applied rigorously

- ⊕ Loss of architectural integrity →
 - ★ Brittle design
 - ★ Growing incidental complexity
 - ★ Surprising emergent behavior

Typical Fault Protection Notions

Concepts

- ⊕ Fault Tree, Failure Modes & Effects Analysis
- ⊕ Error, Fault, Failure
- ⊕ Threshold, Event, Persistence
- ⊕ Detection, Monitor, Isolation, Response
- ⊕ Priority, Level
- ⊕ Critical Period, Mark & Rollback
- ⊕ Safing

etc.

Patterns

- ⊕ Monitors trigger responses
- ⊕ Every monitor and response can be disabled
- ⊕ Responses terminate command sequences

etc.

Principles

- ⊕ Respond only to unacceptable conditions
- ⊕ Avoid hair triggers and retriggering
- ⊕ Tolerate false alarms
- ⊕ Make parameters commandable
- ⊕ Corroborate before severe responses
- ⊕ Ensure commandability and long term safety
- ⊕ Preserve consumables and critical data
- ⊕ Log events and actions

etc.

Fundamental?

⊕ Not Really

- ✦ Imprecise and numerous concepts
 - ✦ Weak patterns and principles
 - ✦ Exceptions and omissions
 - ✦ Cluttered with incidentals
-
- ✦ Part of an even larger collection of interrelated notions in system management

⊕ **No concise “Theory of Fault Protection”**

GN&C Fault Protection

- ⊕ Broadly coupled across the system
- ⊕ Large portion of vehicle fault protection

- ⊕ Good GN&C requires
good fault protection

Is the reverse also true?

What Does Fault Protection Do?

- ⊕ **Observes the system** (*measurements...*)
- ⊕ **Uses models** (*failure modes...*)
- ⊕ **Estimates system state** (*health, hazards...*)
- ⊕ **Coordinates actions** (*conflicts, resource use...*)
- ⊕ **Directs the system** (*commands...*)
- ⊕ **Meets system objectives** (*safety, viability, critical events...*)

Fault Protection is a Control System

Control Fundamentals

Concepts

- ⊕ Objectives on state
- ⊕ Models of state behavior
- ⊕ Knowledge of state
- ⊕ Closed control loops on state

Patterns

- ⊕ Each system state is assigned a cognizant control system
- ⊕ Control systems interact via explicit state knowledge and coordinated objectives
- ⊕ Knowledge and control designs exploit models

Principles

- ⊕ Make objectives explicit, complete and clear
- ⊕ Uniquely assign responsibility for all objectives on a state
- ⊕ Make model usage apparent and consistent
- ⊕ Explicitly coordinate concurrent objectives
- ⊕ Keep state estimation independent of state control
- ⊕ Represent state knowledge uncertainty openly and objectively
- ⊕ Strive for a single source of truth for state knowledge
- ⊕ Make control decisions based only on state knowledge and objectives

Transparency is the Key

- ⊕ Close the semantic gap
 - ★ Map every design element unambiguously to basic concepts and patterns
- ⊕ Exercise architectural discipline
 - ★ Adopt design methods and frameworks that help to enforce principles
- ⊕ Every exception weakens architectural integrity

An Observation

- ⊕ Typical fault protection implementations,
even in GN&C,
are full of exceptions to control transparency

A Sample Mapping Issue

- ⊕ Persistence threshold value:
 - ★ Appears in monitoring functions, but is it...
 - ★ Likelihood, transient duration, system error tolerance, response delay, false alarm avoidance, or what?

- ⊕ Role depends on assumed meaning
 - ★ Detection in state estimation
 - ★ Branching in control decisions
 - ★ Precedence among objectives
 - ★ etc.

Differences in Perspective

- ⊕ “fault protection” detects and responds to faults
- ⊕ ***Fault tolerant control systems*** achieve important system objectives, even when faults happen
- ⊕ “fault protection” is verified by testing all monitors and responses
- ⊕ ***Fault tolerant control systems*** are verified by showing how well they guard expectations of system performance
and so on

Emergent Questions

- ⊕ Why should estimation errors and control errors both trigger fault responses?
- ⊕ Why are normal control functions and fault protection for the same state implemented separately?
- ⊕ What is the objective of “safing”, and how do you know whether or not it’s failing?
- ⊕ ...

Companion Challenges

- ⊕ System operability & ops cost
- ⊕ Autonomy in advanced missions
- ⊕ Complex system-system interoperations

Necessary Steps

⊕ Reassertion of control fundamentals

- ✦ Principled architecture
- ✦ State- and model-based design
- ✦ Goal-driven operation

Conclusion

- ⊕ GN&C and Fault Protection are both fundamentally about *control*
- ⊕ Control works best when its concepts, patterns, and principles are applied *transparently*
- ⊕ GN&C Fault Protection is best when treated as an *integral* part of a *unified* approach to system control