SÛRETÉ DE FONCTIONNEMENT

« OU COMMENT S'ASSURER QU'UN SYSTÈME EST SÛR ? »

B. Monsuez
ENSTA PT

BUILDING SAFE ARCHITECTURE BASED ON MICROPROCESSOR

MANAGING FAULTS

- Fault avoidance : Keep the fault out of the design
  - Goal of the ARP 4754, DO-178 & DO-254
  - Regarding microprocessor based architecture
    - No control on the design
    - Fault may always be present in the microprocessor and may occur
- Fault removal : Remove the fault before the system enters the service
  - Goal of the ARP 4754, DO-178 & DO-254
  - Regarding microprocessor based architecture
    - Test extensively the complete architecture functionally and electrically.
    - Proceed with fault injection
    - Use the monitoring feature provided by the microprocessor to test the architecture.

MANAGING FAULTS (Cntd)

- Fault Detection
  - Detect the fault during service
  - Take the adequate counter-measures to prevent the fault from manifesting itself as an error of failure
  - In case of a microprocessor based system
    - Monitor all the components, for instance the internal registry of the microprocessor to determine if the system is operating correctly
    - Monitor the output of the system to determine if the computed value are corrected
      - Values can be compared to pre-calculated table
      - Values can be compared to output of other systems
- Fault Tolerance
  - Capacity of the system to continue to operate correctly despite the occurrence of a fault.
    - ECC for internal RAM or Buffer
    - ECC for internal & external bus

DETECTING FAULTS

- Functionality Checking
  - Detect the wrong operation of hardware components using routines to check their functionality
    - Routines to check memory
    - Routines to check processor operations
    - Routines to check network communication
  - Done periodically
  - Performs checksums and compare to pre-calculated results
- Consistency Checking
  - Compare the output of software with expected results
    - Range of value
    - Deviance from pre-calculated values.
  - Execute routines to verify the data integrity & consistency
    - Periodical verification of a file system

DETECTING FAULTS: CONSISTENCY CHECKING

- Signal Comparison
  - Compare different signals in redundant systems that are assumed to be equal.
- Instruction and Bus Monitoring
  - Check the operation code and operand for each instruction
    - The processor must allow monitoring of the instruction and operand
  - Check the bus for
    - illegal access (address corruption)
    - Illegal data
- Information Redundancy
  - Parity checks
  - ECC

DETECTING FAULTS: SIGNAL COMPARISON

- Loopback Testing
  - Verify that a signal has reach his destination unchanged
  - An independent path connect the destination to the source so that the signals can be compared at the source
  - Used to test communication lines but also communication network
- Watchdog and Health Monitoring
  - A timer is loaded with a value and get decremented
    - The processor must periodically reload the value
    - If the timer reaches zero, a non-response fault has been detected

FAULT TOLERANT ARCHITECTURE

- Fault Tolerance
  - Capacity of the system to operate properly on the hypothesis of the failure of one (or more) of its component
  - Fault Tolerance is required for all the systems with high availability requirement not only safety critical systems
    - Ex: Internet is a fault-tolerant system

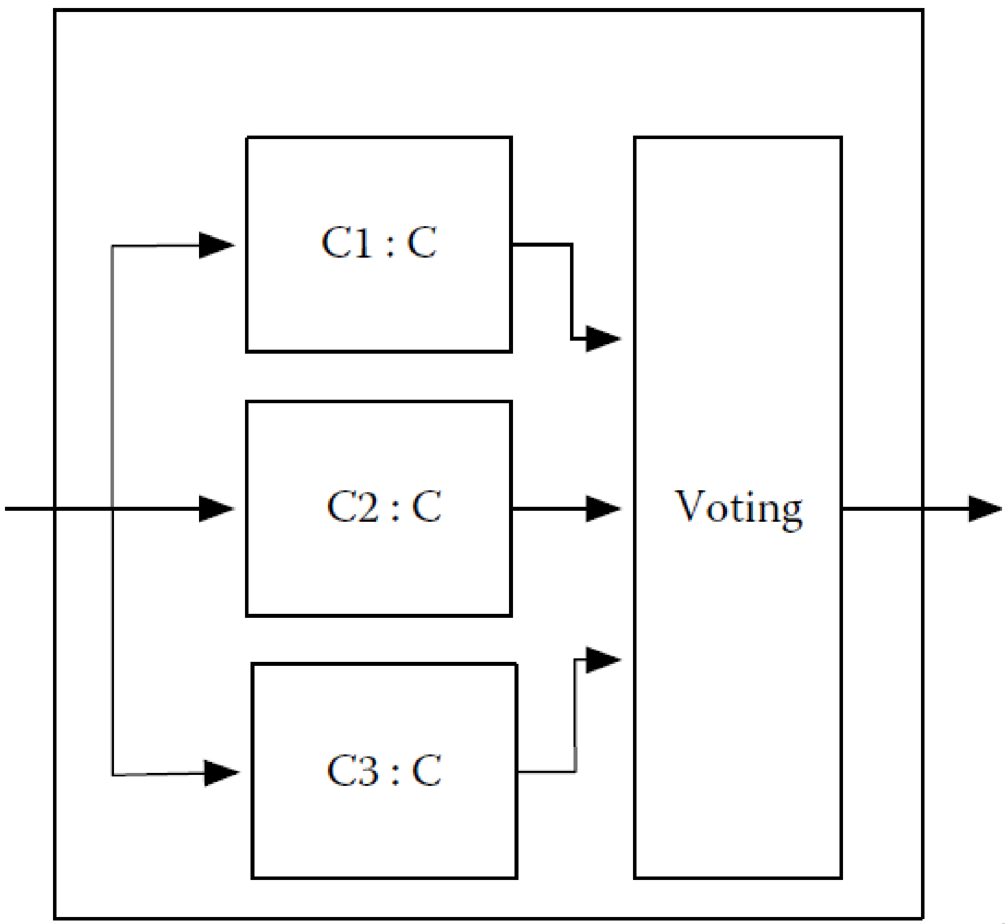
DETECTING FAULTS AT SOFTWARE LEVEL

- Requires that software monitors
  - The current state of the hardware
    - Do health monitoring by reading the different registers that describe hardware states
    - Do health monitoring by handling interruption generated by hardware when a specific operation appends
  - The current state of the executing processes
    - Tests if the processes are alive
    - Tests if the processes complete in the expected tome
    - Tests if the output values of the functions are valid
- Requires that software covers all the faulty state

MANAGING FAULTS AT SOFTWARE LEVEL

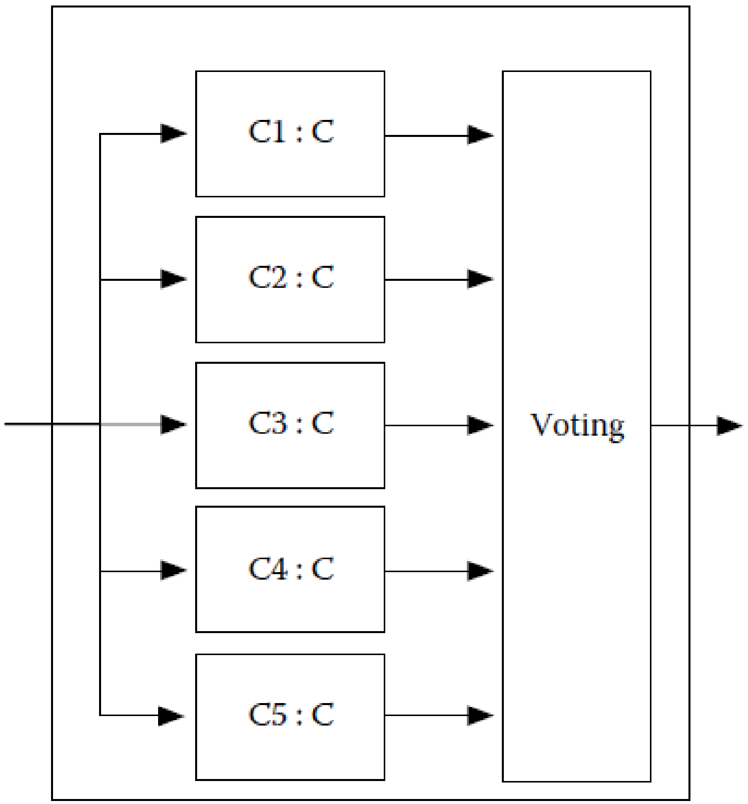
- When fault is detected
  - Determine the fault gravity
    - Determine if the equipment is compromised or not
  - Proceed with the appropriate steps
    - Disconnect or reconfigure a faulty unit
    - Stops and relaunches a faulty process
    - Goes into a degraded mode to pursue the operation
    - Cuts non critical operations if required.

TRIPLE MODULAR REDUNDANCY



- At most one replicated component fails
- The voting mechanism does not fail !
- There are no systematic failure
- Isolation of the failed component is not an issue

MULTIPLE FAILURES



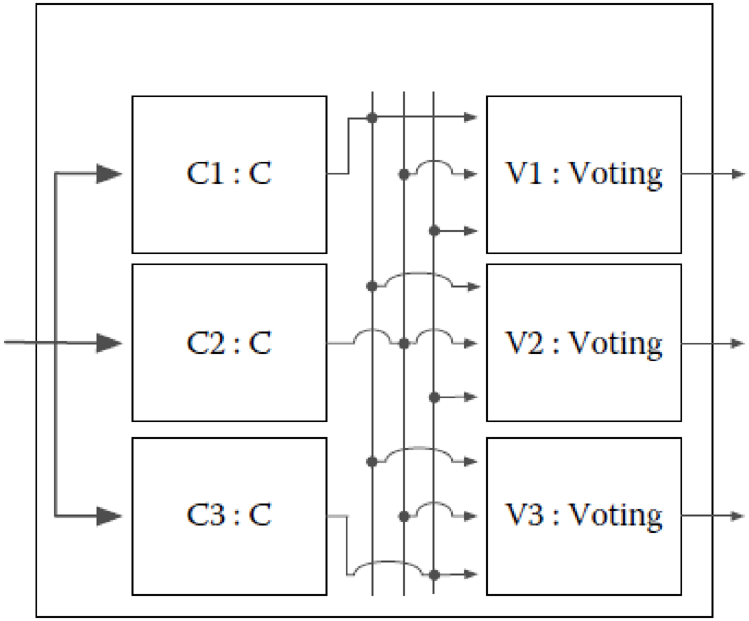
N-Modular redundancy where

- N is an odd number
  - Majority voting is required to determine the output
- Up to (N-1)/2 redundant component may fail
- The voting mechanism is assumed not to fail.
- No protection along systematic failure

MULTIPLE FAILURES OF THE VOTING COMPONENT

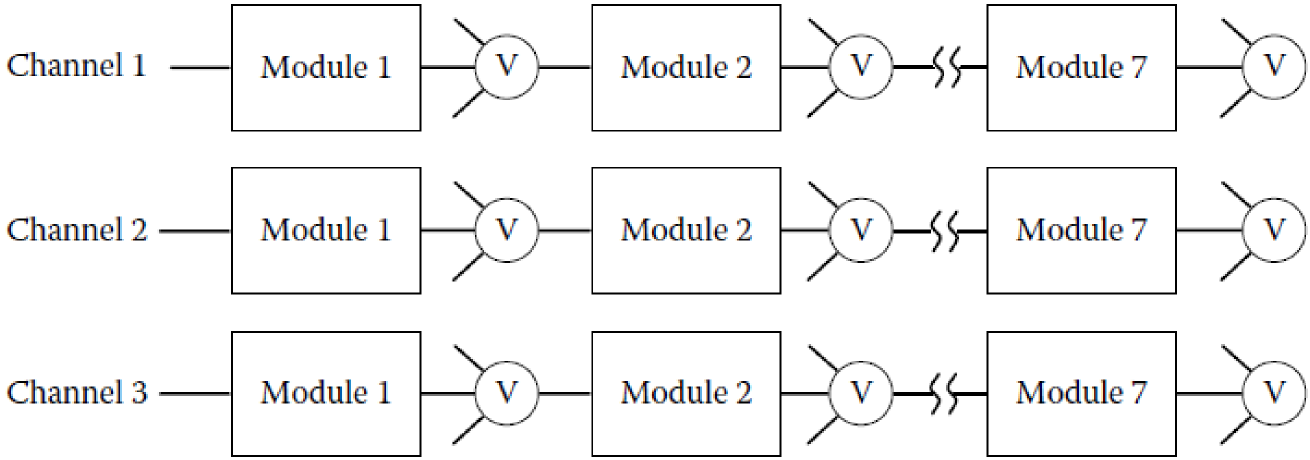
- About reliability of the voting component
  - Voting component is a relatively simple component
    - Four nand gates per bits to be compared
- The level of confidence may not be adequate
  - Was not considered safe enough for the launch vehicle digital controller of the Saturn V rocket
- Solution : triplicate the voting component

TRIPLE VOTING ARCHITECTURE



- Voting component is triplicated
  - Robust to one failure of a voting component
  - However, we have to handle three outputs
- Can only be used in cascading Triplicated Redundant architecture
  - Connection from on stage to another stage of the architecture

TMR MULTIPLE STAGES TRIPLE VOTING ARCHITECTURE



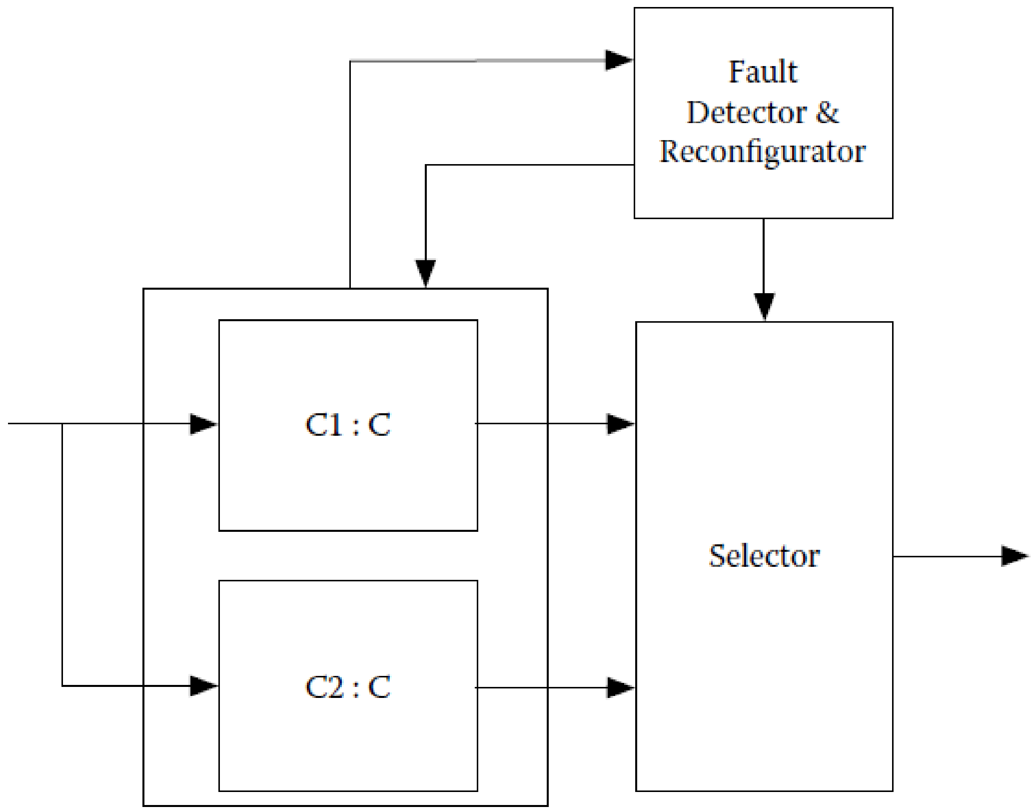
SYSTEMATIC FAILURES

- Systematic failures
  - Errors in the specification or the design of the replicated components
  - All the components will fail the same way in the same context.
  - Voting component will not detect any error.
- Use Component Diversity
  - Use of 3 different microprocessors
- Use Temporal & Spatial Diversity
  - Use delayed inputs
  - Use different execution contexts
- Diversity may not always be the solution
  - May introduce new sources of errors

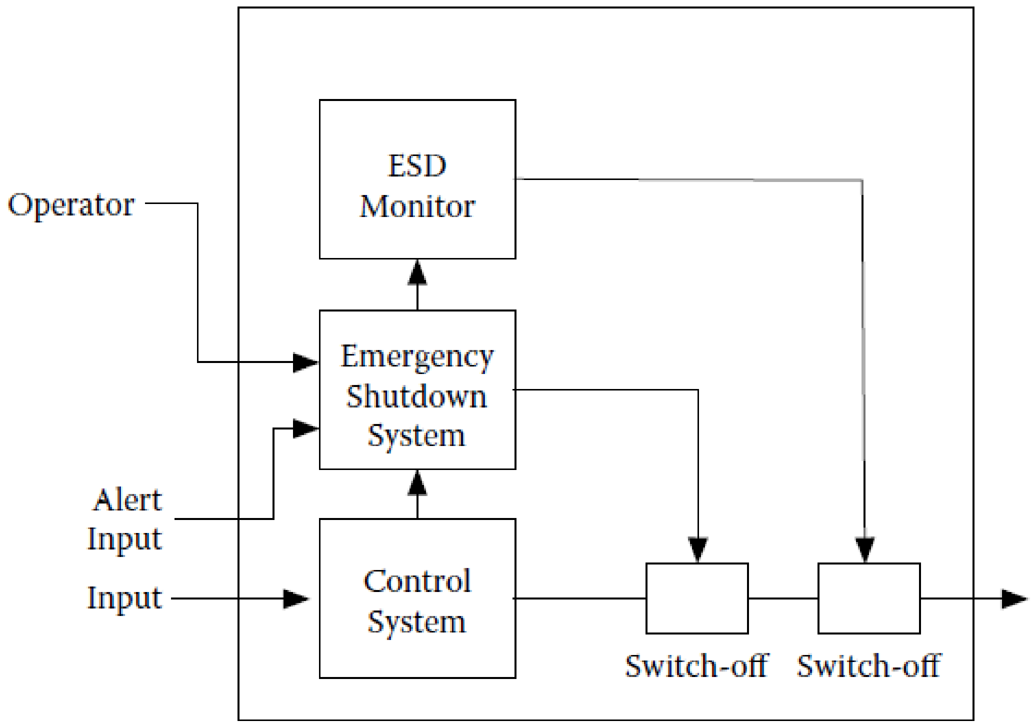
FAULT TOLERANCE & FAULT DETECTION

- Faulty Component
  - May interfere with the system
  - Requires to be isolated from the system
  - Example : an Ethernet port of a micro-controller is polluting the network.
- Requires
  - To identify the faulty component
  - To switch off or to restart the faulty component
  - To switch to a stand-by component if available.
- Stand-by components
  - Hot stand-by : runs in parallel to the standard component.
  - Cold stand-by : turned off and must be switched on.
  - Warm stand-by : turned on but must be synchronized.

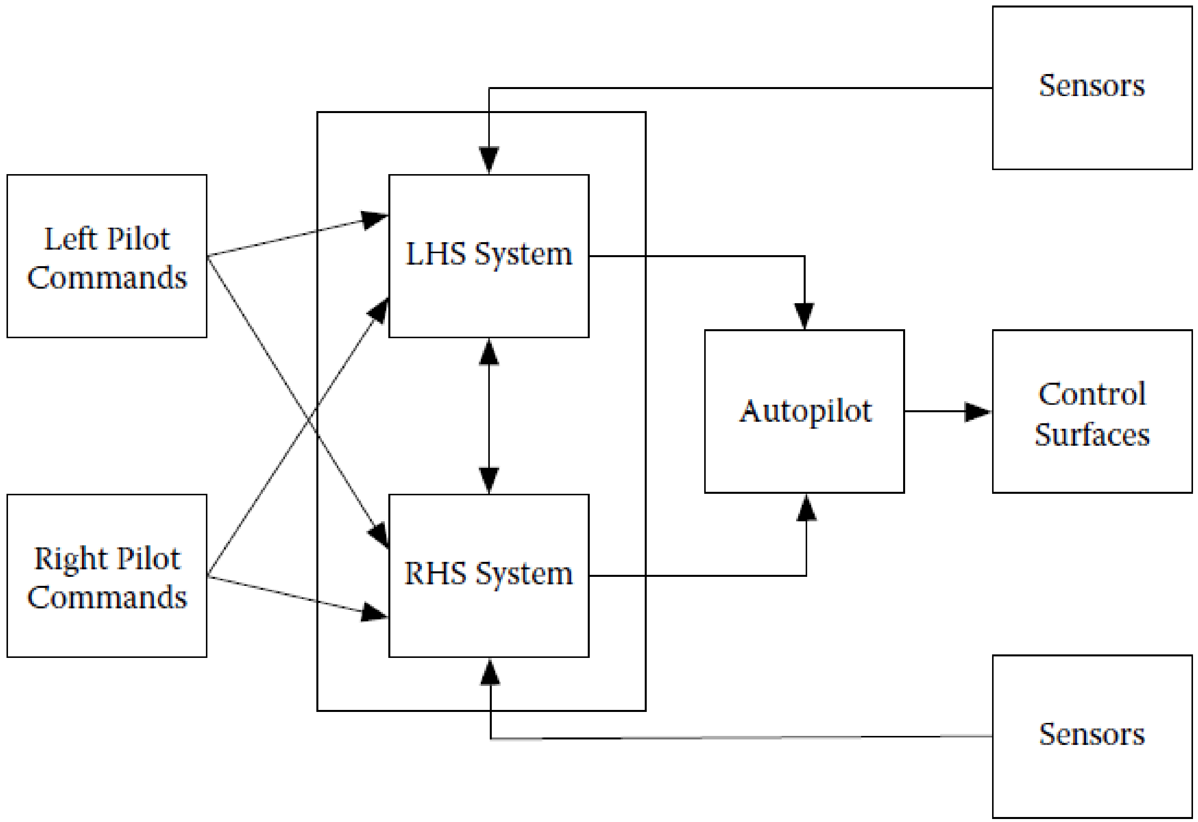
ACTIVE REDUDANCY



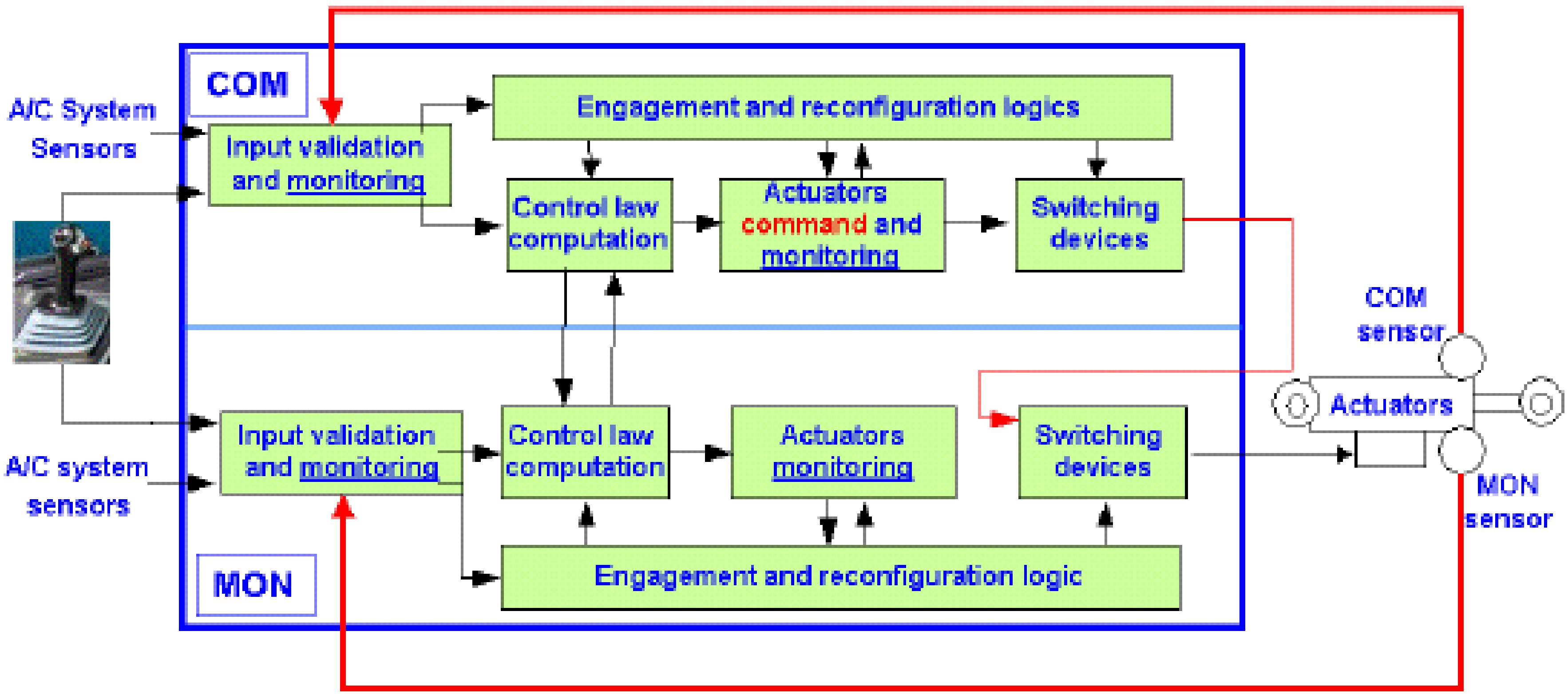
EMERGENCY SHUTDOWN SYSTEM



REDUNDANT FBW ARCHITECTURE



REDUNDANT FBW ARCHITECTURE



SOFTWARE FAULT TOLERANT ARCHITECTURE

- Software Architecture may also be include fault tolerant process
- Dual-Time
  - The process is executed twice
  - If an external non systematic perturbation occurs, probability that it will occur twice is very low
  - Take the most probable value as a result of the function
- Software Run-time Checking
  - Monitor compares results to pre-calculated values
  - Value not in the variance of the outputs
    - May substitute its own output
    - May invoke a piece of code to return in a safe state