



ISWHM: Tools and Techniques for Software and System Health Management

Kick-off meeting

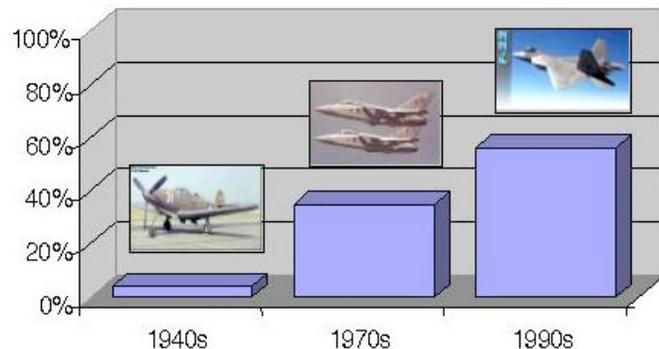
10/10/08

Johann Schumann, RIACS

NASA Ames

IVHM in Aircraft

- Modern aircraft
 - Have IVHM for major electrical/mechanical subsystems
 - Important for safety, reliability, environmental impact, economical considerations
 - Rely heavily on SW
 - but: no health mgmt system for SW



Avionics costs = SW costs

Many Software problems

K.I.S.S.: Attach SW to IVHM

not that easy...

- Software problems don't develop over time
 - come in during all phase of SW life cycle
 - “don't go away”
- SW failures mostly occur instantly–HW often fails gradually (e.g., an oil leak)
- SW problems occur due to problematic interoperation with HW
- SW IVHM is a piece of software

HW-SW Interaction

- HW (e.g., sensors) can behave differently than expected (and thus cause a SW failure)
 - on purpose: use same SW for different HW
 - Ariane V failure
 - accidentally during development
 - DART: new GPS system just before launch
 - HW failure
 - broken cable
 - disabled sensor (e.g., Deep Space I)
 - gradual degradation
 - increase of sensor noise



SW IVHM is SW

Quis custodiet ipsos custodes?

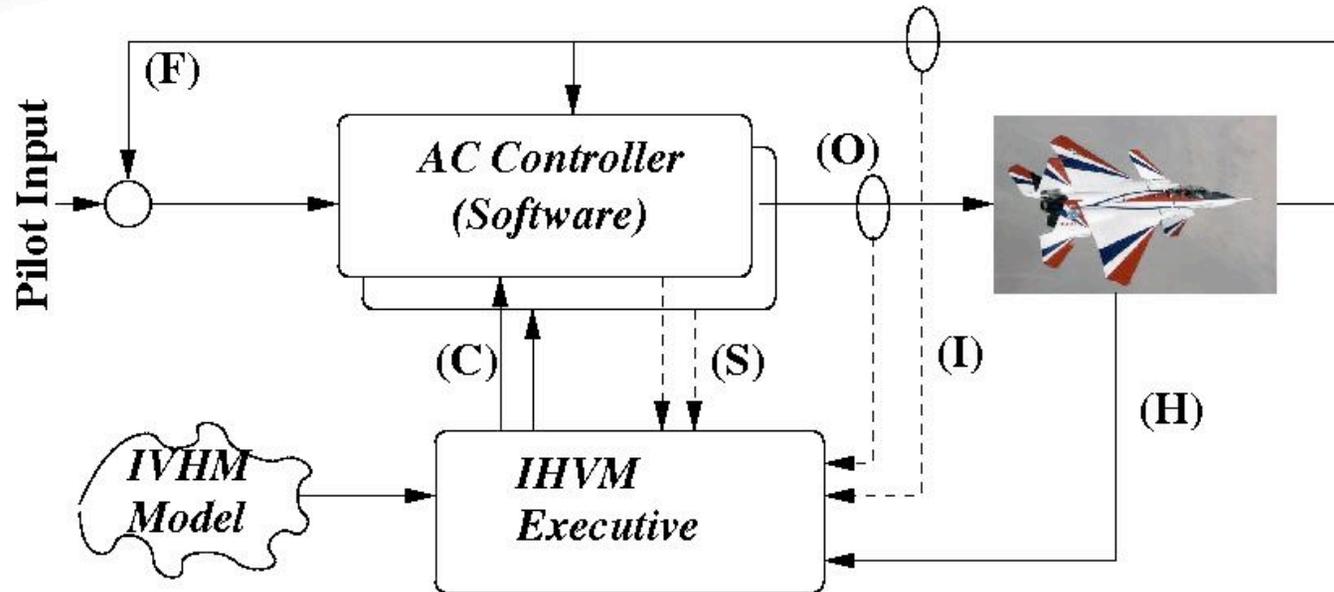
Juvenal

- The IVHM system that monitors the SW system must be at least reliable as the SW under scrutiny
 - false alarms are not an option
 - un-detected failures are a safety hazard

ISWHM

- I. Software Health Management must be integrated seamlessly into IVHM*
- II. Exceptions and Error Messages are not enough*
- III. All IVHM systems must be verified*
- IV. All flight-critical software must be certified*

Proposed ISWHM Architecture



- Controller provides probabilistic quality metric
- Advanced IVHM system (compiled version of Bayesian IVHM model) *Adnan, Ole*
- IVHM executive/reasoner subject to V&V

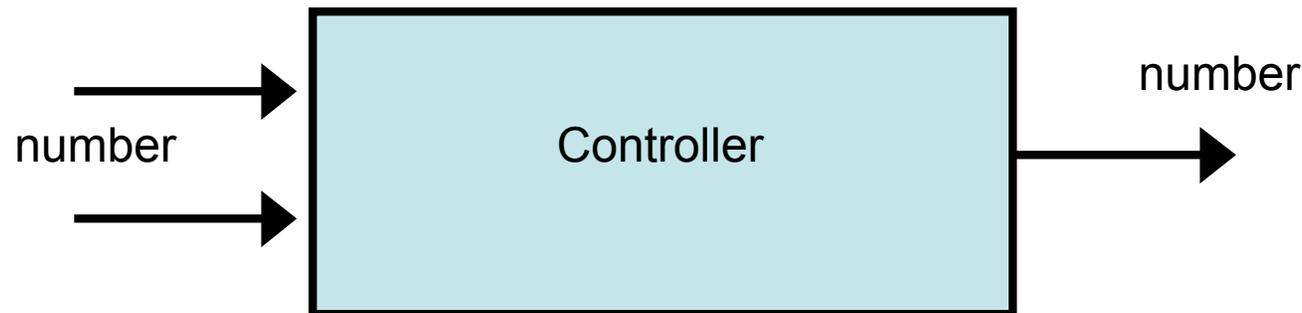
Hybrid Systems

- Most flight SW systems are hybrid systems
 - discrete components (e.g., Stateflow, finite state machines, mode logic, gain schedules)
 - continuous components (e.g., control loop, matrix operations, RLSQ, ...)
- System must be designed with ISWHM in mind
 - combination of before-deployment techniques
 - dynamic monitoring techniques
 - when properties could not be fully checked during V&V
 - new/changed properties (e.g., damage, env. change)
 - dynamic properties on system status

For the discrete components, we will use a combination of advanced MC techniques and Runtime Verification

Health Metric

- Traditionally, an algorithm (e.g., control system) takes numerical data and produces numerical data.

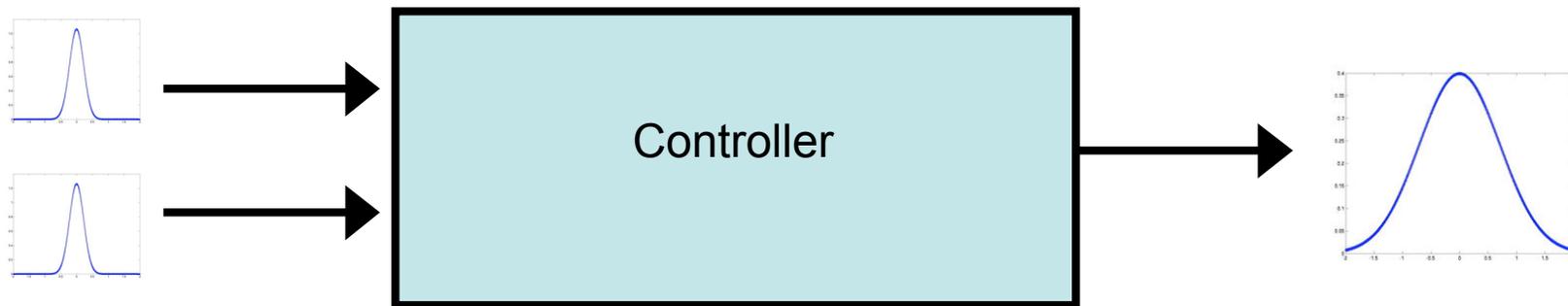


The output produced does not contain any notion of

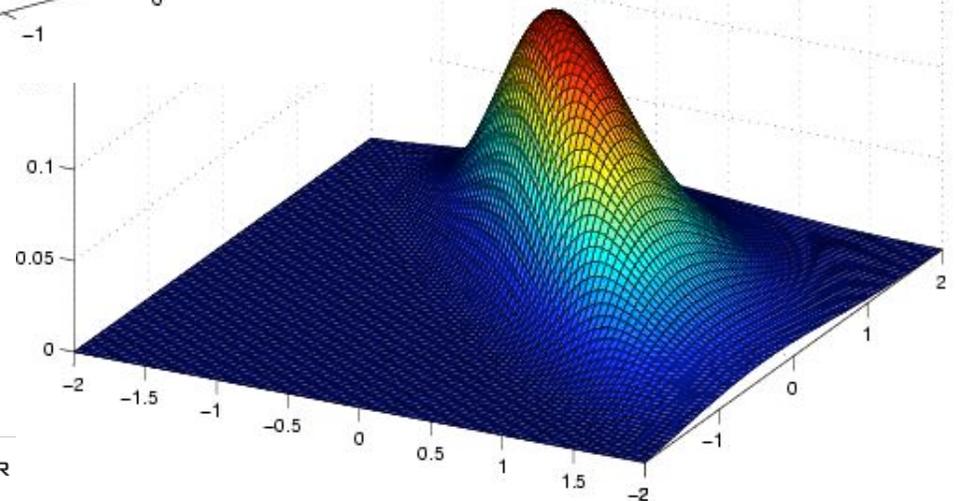
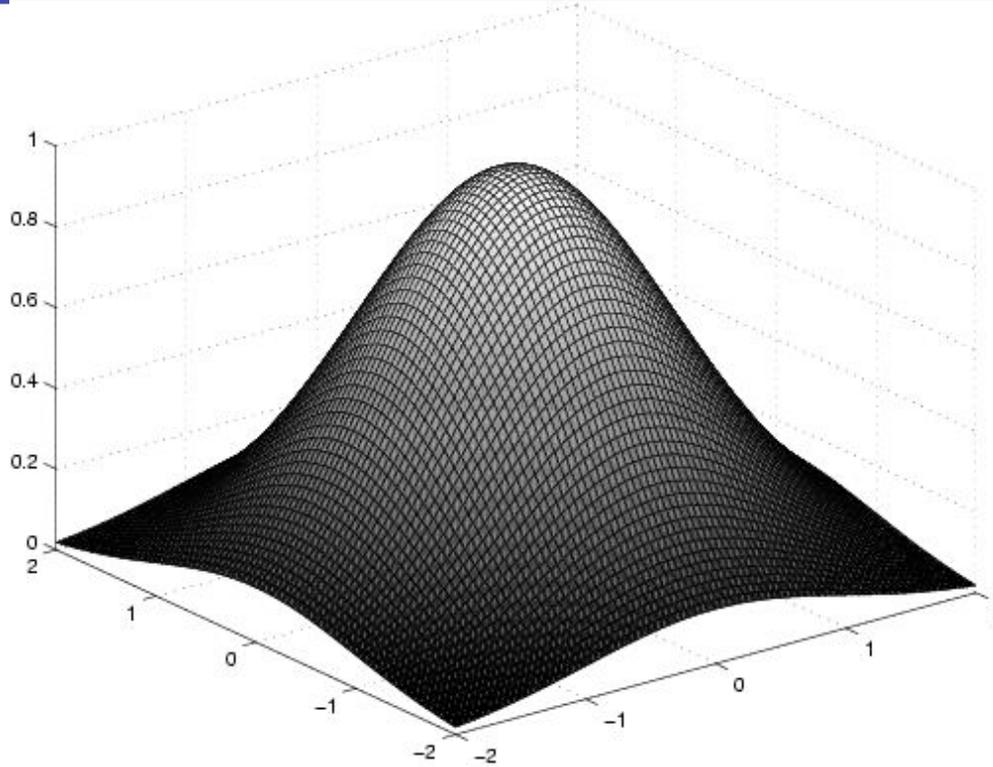
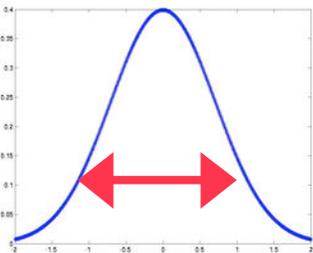
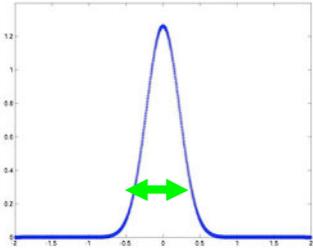
- quality of input data (e.g., are the sensor data OK or noisy?)
- quality of calculation (big round-off errors?)
- quality of internal parameters (are we at the stability limit of the controller?)

Health Metric

- An algorithm with built-in health metric takes probability variables as inputs and outputs.
- Shape and width of the Probability density function comprises the health metric



Health Metric



Narrow Gauss curves = good quality/health
Wide gauss curves = bad quality/health

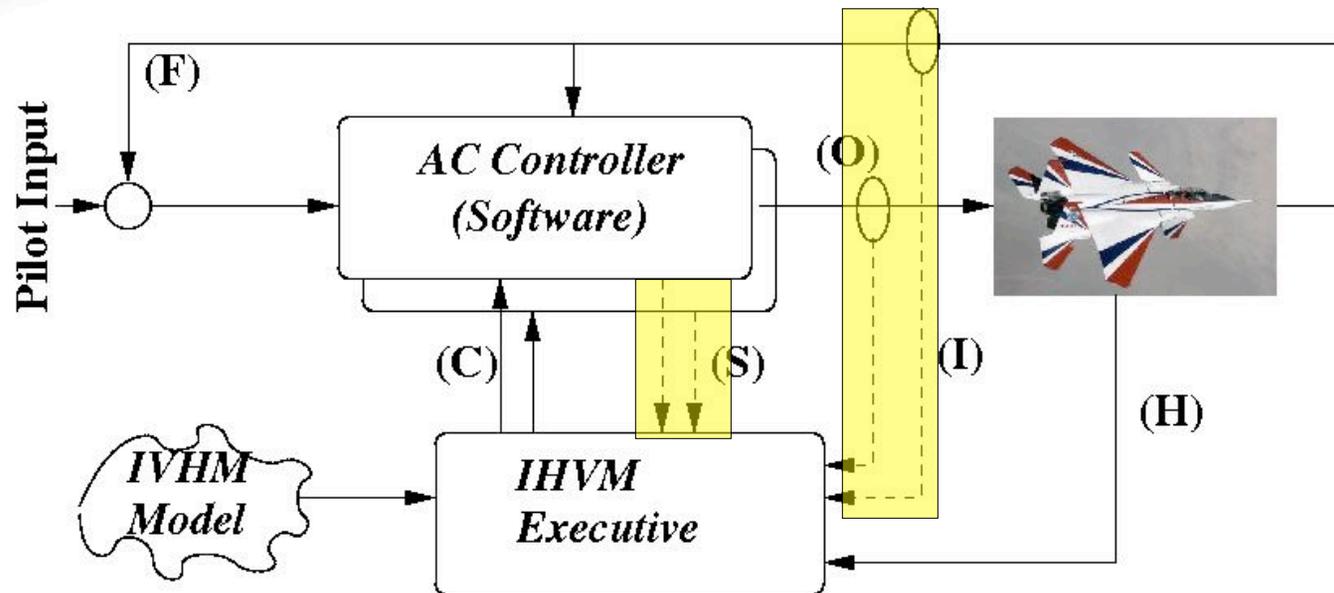
Health Metric

- Bayesian statistical theory provides a solid formal basis for calculations with probability distributions

Kalman filters and the Confidence Tool use a Bayesian approach. Our technology is based on similar concepts and will be developed toward selected algorithms in the area of GN&C.

- Traditionally, Kalman filters provide quality of estimates, based upon quality of the input signals. In most cases, this metric is used to “reset” the filter, if the quality gets too poor, but the current quality of the estimates is seldomly used for other purposes. Other issues, like numerical stability and round-off errors have been analyzed, but are not handled explicitly by the algorithms
- The Confidence Tool, developed within the IFCS project dynamically calculates the quality of the neural network using a Bayesian statistical approach. As the NN is trained during the flight, the Confidence tool output can be used to early detect poor performance and detect diverging NN learning.

ISWHM



- Bayesian quality metric, Runtime Verification, and additional SW monitor results are fed into advanced IVHM reasoner
- IVHM model and FDIR technology for combined SW/HW system

Verified IVHM

- Modern IVHM systems use advanced and complex algorithms for diagnosis, prognostics, root-cause analysis
 - Logic based reasoning systems, model compilation, Bayes Nets, Arithmetic Circuits
 - Multivariate optimization (search machine learning, fuzzy, neural networks, ...)
- Current V&V techniques are not sufficient to show safety and reliability of such algorithms and their implementation

Here, we don't talk about V&V of the *model* (correctness, consistency, etc)

V&V Aspects

- For the V&V of the IVHM reasoning, we need to address
 - correctness and completeness of model compilation
 - Does Arithmetic Circuits produce same results as BN?
 - functional correctness for IVHM correctness
 - runtime and memory limitations

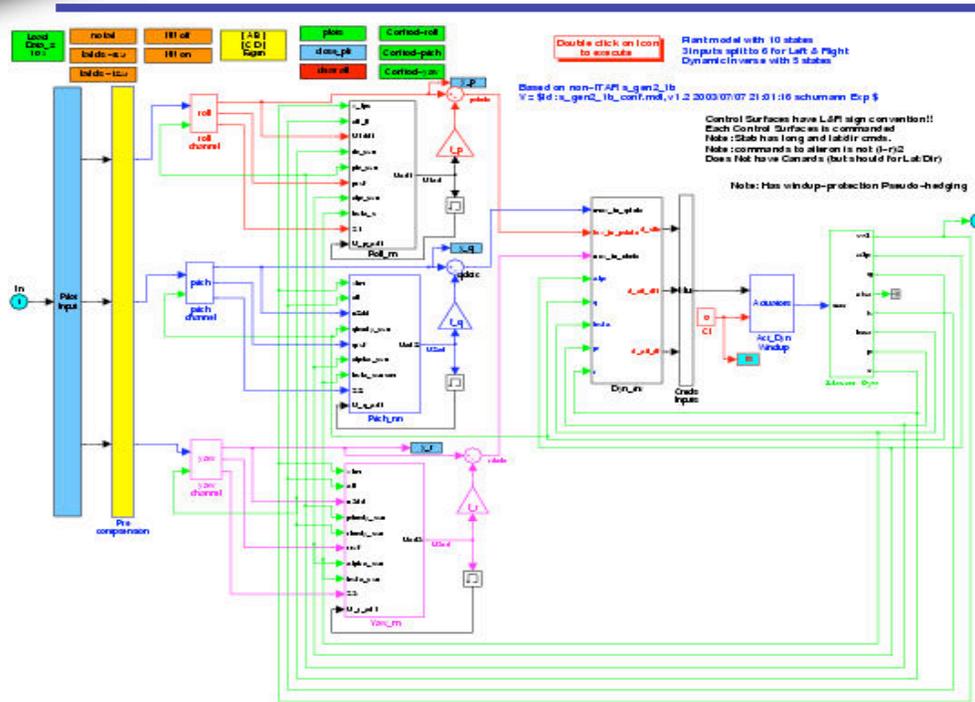
IVHM V&V Tools

- We will draw upon advanced V&V techniques and methods
 - Model checking
 - Static analysis
 - Formal proofs (e.g., model compilation correctness)
 - Test-case generation with test data analysis

Many of these techniques have been developed within the RSE group (e.g., Java PathFinder, Livingston SMV model checker). However, they need to be adapted toward the specific algorithms. Together with formal arguments, a process needs to be developed to *demonstrate* safety, reliability and performance of the IVHM algorithms.

With demonstrated quality/correctness of the IVHM models (for both physical systems and software), *safety* and *dependability* cases for the entire system can be developed.

Testbed Candidates I



- IFCS Gen-II controller
- non-ITAR Simulink model (and experience) available
- PI controller with non-linear adaptive component

Testbed Candidates II



- SmallSat demonstrator
- GN&C system
 - INS sensors
 - implemented in Simulink/Stateflow
 - developed at ARC

Testbed Candidates III



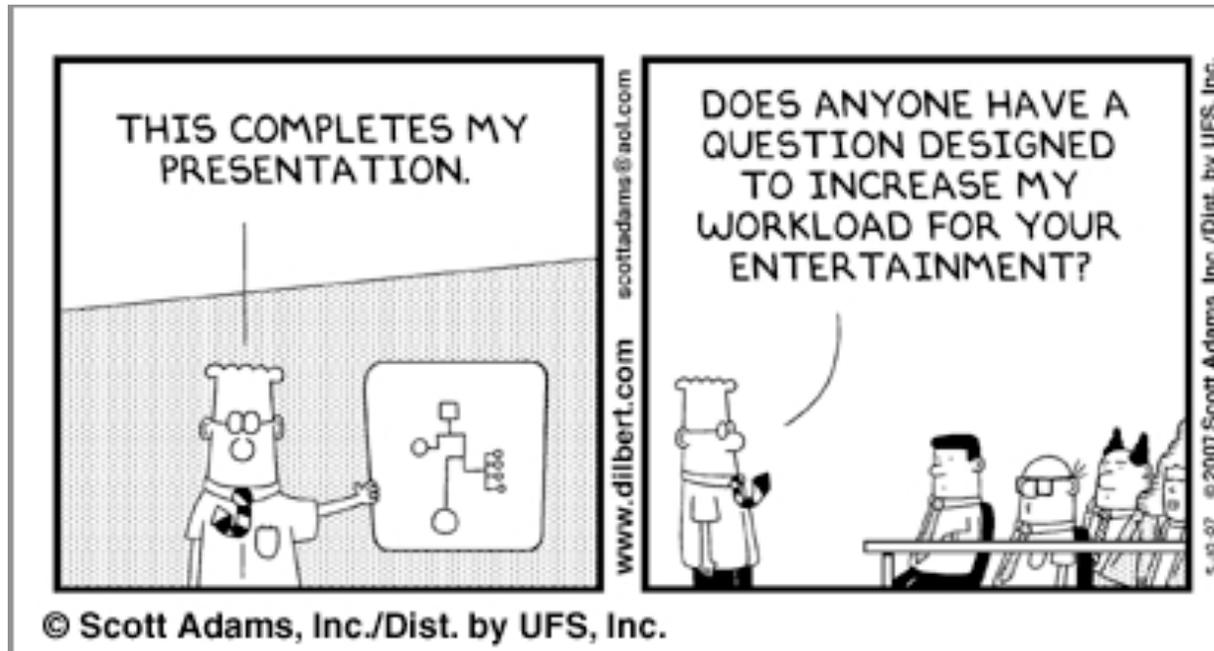
- IVHM testbed
- power distribution
- multiple sources (batteries)
- multiple loads
- multiple power routing
- developed at ARC

Conclusions

Exciting times lie ahead!

- I. Software Health Management integrated seamlessly into IVHM*
- II. Statistical Quality Metric for continuous components combined with Runtime Verification/Monitoring of discrete SW*
- III. IVHM reasoner/executive verification*
- IV. All flight-critical software must be certified: Dependability and Safety Cases*

Conclusions



BACKUPS

- Data acquisition
 - Dynamic monitoring of software artifacts with runtime assertions
 - based on RSE work by K. Havelund et al
 - Assertions coming from requirements and out of formal methods tools and model checking
 - Dynamic monitoring of software systems (e.g, CPU load, memory consumption,...)
 - Algorithms with built-in health metric

Diagnosis, Prognostics

- Use system data (from Hardware *and* Software systems)
- Use *system model*
- Fault detection, root cause analysis,...

Within the proposed work, a state-of-the-art IVHM system/algorithms will be used. Important requirements include:

- hybrid for continuous and discrete data
- verifiability (see V&V of IVHM algorithms)
- powerful modeling capabilities
 - modeling of software characteristics with advanced monitoring data
 - modeling of hardware system wrt. Software interaction

ISWHM: Mitigation

- Many techniques exist for mitigation of software faults, once they are detected/isolated
 - Redundant computation
 - Automatic SW reconfiguration, SW adaptation
 - SW re-juvenating, self-healing SW...
- Wide research range: fault tolerant, redundancy, self-healing, adaptive SW, dynamic reconfiguration,...

In the proposed work, we will survey existing techniques. For the testbeds and demonstration, we will use a suitable, simple way of fault mitigation.

Schedule

- Y1: Survey on state of the art SW IVHM
 - Taxonomy, approaches, tools and techniques
 - V&V of IVHM: state of the art and gaps
- Y2:
 - Develop Bayesian approach for selected algorithm (GN&C) area and perform initial experiments on simulation-only testbed
 - Develop specific V&V for selected IVHM algorithm
- Y3:
 - Implement ISWHM on selected test-bed. Run experiments and demonstrate system.
 - Report on approach toward ISWHM with initial draft of ISWHM V&V process
 - Prepare journal publication