

DTIC<sup>®</sup> has determined on *Oq Nov 2009* that this Technical Document has the Distribution Statement checked below. The current distribution for this document can be found in the DTIC<sup>®</sup> Technical Report Database.

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

© **COPYRIGHTED**; U.S. Government or Federal Rights License. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

**DISTRIBUTION STATEMENT B.** Distribution authorized to U.S. Government agencies only (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

DISTRIBUTION STATEMENT C. Distribution authorized to U.S. Government Agencies and their contractors (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

**DISTRIBUTION STATEMENT D.** Distribution authorized to the Department of Defense and U.S. DoD contractors only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

DISTRIBUTION STATEMENT E. Distribution authorized to DoD Components only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

DISTRIBUTION STATEMENT F. Further dissemination only as directed by (inserting controlling DoD office) (date of determination) or higher DoD authority.

Distribution Statement F is also used when a document does not contain a distribution statement and no distribution statement can be determined.

DISTRIBUTION STATEMENT X. Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoDD 5230.25; (date of determination). DoD Controlling Office is (insert controlling DoD office).

REPORT DOCUM		Form Approved OMB No. 0704-0188						
Public reporting burden for this collection of information is estimated to everage 1 hour per response, including the time for reviewing instructions, seerching deta sources, gethering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other espect of this collection of information, including suggestions for reducing this burden to Washington Heedqueters Service. Directorate for Information Operations and Reports, 1215 Jefferson Devis Highway, Suite 1204, Artington, VA 22202-4302, and to the office of Management and Budget, Pepervork Reduction Project (0704-0188) Weshington, DC 20503 PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS								
1. REPORT DATE (DD-MM-YYY)	2. REPORT TYPE			3. DATES COVERED (From – To)				
31-October-2009	Performance/Tec	hnical Repo	ort	22-Feb-2009 to 30-Jun-2009				
4. TITLE AND SUBTITLE		5a	5a. CONTRACT NUMBER					
INTEGRATED RECONFIGURABLE IN	TELLIGENT SYSTEMS	N	N00014-08-1-0107					
		5b	GR	ANT NUMBER				
		N/	A					
		5c	PR	OGRAM ELEMENT NUMBER				
		N/	A					
6. Author(s)	· · ·	5d	PR	OJECT NUMBER				
Dr. Dimitri N. Mavris								
		5e	TA	SKNUMBER				
		5f.	5f. WORK UNIT NUMBER					
			1					
	AND ADDRESS(ES)		NUMBER					
Georgia Institute of Technology			N/A					
Atlanta, GA 30332-0150	- ing							
9. SPONSORING/MONITORING AGENCY NA	ME(S) AND ADDRESS(ES	)		10. SPONSOR/MONITOR'S ACRONYM(S)				
Office of Naval Research 875 North Randolph Street			ONR					
			11. SPONSORING/MONITORING					
Arlington, VA 22203-1995			AGENCY REPORT NUMBER					
			N/A					
12. DISTRIBUTION AVAILABILITY STATEME	NT							
Unlimited Distribution	0	000	4	100117				
13. SUPPLEMENTARY NOTES	2	UUS		109147				
			_					
14. ABSTRACT The report details the research worl the period of February 22 to June 30 develop and refine the integrated mu of complex naval systems for improve to fulfill this objective. Models re architecture with inference engine to plain NN models and NN models with a surrogate models; a suitable framewow management for developing the HMI of proposed to conduct design space exp survivability and mission effective	thas been done by AS 0, 2009. The main obj odeling and simulatio ng the ship design a evised based on the n vas proposed and eval the block-structure w ork and a database en the simulation envi ploration in order to ness. The rest of the	DL in develop ective of the n environment nd operations otional YP we uated based of as performed gine has beer ronment; a ro obtain an ul report will	ing wor . Fi re i for sel bust tima expl	and applying the IRIS concept for ck for this period is to further order to investigate the behavior integrated and tested; a control sfined scenarios; comparison of evaluating accuracy of the lected to facilitate the information c and a resilient approaches were ate design with increased lain the accomplishments for each				
task in details.  15. SUBJECT TERMS								
Modeling & Simulation, Reconfigurability, Integrated & Intelligent, Naval Systems								
16. SECURITY CLASSIFICATION OF:           a. REPORT         b. ABSTRACT         c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19	9a. NAME OF RESPONSIBLE PERSON				

•

r

1

 A. REPORT
 b. ABSTRACT
 c. THIS PAGE
 ABSTRACT
 OF PAGES

 U
 U
 U
 SAR
 16

## TECHNICAL REPORT (February 22, 2009 – June 30, 2009)

# "Integrated Reconfigurable Intelligent Systems (IRIS) for Complex Naval Systems"

Contract #: N00014-08-1-0107

SUBMITTED TO: Mr. Anthony J. Seman, Office of Naval Research (email: Anthony\_Seman@onr.navy.mil)

> Office of Naval Research 875 North Randolph Street Arlington, VA 22203-1995

### **SUBMITTED BY:**

Georgia Institute of Technology School of Aerospace Engineering Aerospace Systems Design Laboratory (ASDL) Atlanta, Georgia 30332-0150

AWARD PERIOD: December 31, 2007 to June 30, 2009

October 31, 2009

#### Principal Investigator:

Professor Dimitri N. Mavris Director Aerospace Systems Design Laboratory School of Aerospace Engineering Georgia Institute of Technology Phone: (404) 894-1557 dimitri.mavris@ae.gatech.edu

## Summary

The following report details the research work has been done by ASDL in developing and applying the IRIS concept for the period of February 22 to June 30, 2009. The main objective of the work for this period is to further develop and refine the integrated modeling and simulation environment in order to investigate the behavior of complex naval systems for improving the ship design and operations. Five individual tasks were conducted to fulfill this objective. Models revised based on the notional YP were integrated and tested; a control architecture with inference engine was proposed and evaluated based on defined scenarios; comparison of plain NN models and NN models with the block-structure was performed for evaluating accuracy of the surrogate models; a suitable framework and a database engine has been selected to facilitate the information management for developing the HMI of the simulation environment; a robust and a resilient approaches were proposed to conduct design space exploration in order to obtain an ultimate design with increased survivability and mission effectiveness. The rest of the report will explain the accomplishments for each task in details.

# Task 1: Integration of Heterogeneous Systems

In order to represent a notional YP ship in a computer simulation environment, the models of sub-components must be created and integrated into a single model that represents the ship. The sub models used were created by several members of the GT IRIS team. They comprise of a power model for the representation of the electric system, a fluid model that represents the cooling fluid flow, three layers of controllers, a scenario script, and a Human-Machine Interface.

The power model is modeled in Simulink. It is set up modularly, and the number of loads is easily changeable. It is a physics-based model, and has the main physical components that an actual power model would include, such as controllers, busses, service loads, etc. The power model creates the thermal loads to be taken carc of by the fluid flow model. The cooling fluid flow is modeled in Flowmaster V7. It represents the cooling flow network, and interacts with the service load temperatures from the power model. The fluid model also has a damage mode modeled in. This is simply represented by two valves that open into the environment, thus simulating a pipe rupture, similar to what would be the outcome of a missile hit of the ship.

The opening of the rupture valves is controlled by the scenario script, which cssentially determines when a rupture happens, and where. Currently, ruptures are only implemented at one location in the fluid model, but more will be modeled in a next effort.

The three controllers are responsible to correctly react to ruptures within the fluid flow model and distribute the cooling fluid to the service loads. The controllers detect the rupture location, and shut down valves accordingly such that the rupture within the flow network is isolated, and cooling fluid loss is prevented.

The sub models were integrated using Phocnix Integration ModelCenter. A scheduler script was written, which takes care of running the models and exchanging the necessary

data between the respective models over the time. The script also allows for temperature unit conversions between some models. It further allows for arbitrarily scheduled execution of sub models, a feature that proved to be necessary for correct execution and communication between the low-level and mid-level controllers. Also, models can be exchanged and modified more easily when integrated with a script instead of the ModelCenter link editor. Lastly, the scheduler allows for adaptive time steps, a feature that will be implemented once the simulation runs correctly.

A makeshift interfaee for data output and visualization was included. The output is into an Excel worksheet. It allows for free choice and selection of variables to be displayed in standard Excel graphs. However, data is only stored until the next simulation run. More importantly, the interface is read-only, which means that the simulation cannot be influenced during the run, the Excel interface only helps to display variables. A much more sophisticated database driven Human Machine Interface (HMI) is currently under development. A preliminary version has been shown to work successfully with the simulation.

Previous investigations into general methods of optimizing simulation execution time and accuracy have shown that an adaptive simulation time step is beneficiary to both real world execution time and result accuracy. Related publications by Nairouz and Hoepfer have been eited previously. The approaches used for these publication was a crude rule of thumb algorithm, with no further evaluation regarding optimized parameters etc. Hence, further efforts were made to investigate this issue. This leads to the more general question of model evaluation and optimization. It must be kept in mind that in a real simulation, such as the one that will eventually model the actual YP ship, consists of dynamic sub models whose properties are unknown. Hence, it must be determined whether the outputs from the integrated co-simulation aetually represent the real world system output. Time step optimization will then be a sub problem to this general problem. First steps have been made to evaluate which approaches seem appropriate to determine the actual system output. Ideally, the applied algorithm would determine, from the eurrent simulation step, the next simulation step and the system states at that point. It also would determine an error bound within which it is assumed that the function accurately represents the real world system. If the system stays within this error bound, it is deemed to be accurate, and a next time step ean be evaluated. If the system goes beyond the error bound, then it is assumed that it does not accurately represent the real world system. A deviation from the error bounds might also indicate an external shoek input to the system. In either ease, eorrective actions must be determined which will need to be taken in order to get the simulation back into accurate real world representation. In the case of error deviation due to internal model behavior, a reduced time step may be an accurate and simple solution. In the ease of external shoek, such as sudden system alteration due to ruptures etc., the ease is different, since the simulation must be able to handle such instances, and reconfigure the system accordingly. Hence, the corrective action necessary will need to be more elaborate.

For simple simulation of an integrated model, mathematical models have been investigated that might help to determine the correctness of the simulation outputs. These

methods are based on numerical methods for the solution of differential equations. If a differential equation is to be solved on a digital computer, it may not be able to solve the equation and use the solution to determine the "path" of the equation variables. Hence, numerical methods are used to solve such equations. The main property that links these methods to the simulation of an integrated model is that, for both eases the actual equations are unknown and hence need to be approximated. First approaches and algorithms have been identified, and are currently being implemented on a simple test model. Further literature review showed that there are more sophisticated methods for evaluation, which represent predictor-corrector methods. These methods first estimate the next time step point, and then use corrective measures to determine the accuracy of the point. Also, some methods have adaptive time steps included, which keep the simulation within desired error bounds. However, these methods require large computational expenses, and an investigation may be necessary to determine the tradcoff between accuracy and computation time.

# Task 2: Multi-Agent Based Mid-level Control with Dynamic Inference Engine

### Introduction

Increasing societal demand for automation has led to considerable efforts for controlling large-scale complex systems, especially in the area of autonomous intelligent control methods. A control system of a large-scale complex system needs to satisfy four system level requirements: robustness, flexibility, reusability, and scalability. Corresponding to the four system level requirements, there arise four major challenges of controlling large-scale complex systems. First, it is difficult to get accurate and complete information. Second, the system may be physically highly distributed. Third, the system evolves very quickly. Fourth, emergent global behaviors of the system can be caused by small disturbances at component level. To deal with those challenges, Hybrid Multi-Agent Based Control (HyMABC) architecture with Multiple Sectioned Dynamic Bayesian Nctworks (MSDBNs) inference engine have been proposed.

## Hybrid Multi-Agent Based Control (HyMABC) Architecture

HyMABC architecture combines hierarchieal control architecture and module control architecture together to form a hybrid control architecture. First, it decomposes a complex system hierarchically; second, it combines the components at the same level as a module and then designs common interfaces for all of the components in the same module; third, a few replications are made for critical agents and are organized into some logical rings. It keeps clear guidelines for complexity decomposition and also reduces communication complexity of the distributed control system.

For an important control agent such as the highest level agent in the multi-agent system, if it is damaged or unavailable, the whole system will lose the global control even though the subsystems ean work according to their available information. In order to keep the whole system working and prevent a failure of significant control agent from occuring, a few replications are created and arranged in a robust and efficient way to insure

automatic reconfiguration when necessary. Similar to the idea of fault-tolerance with replicated main containers in Java Agent DEveleopment (JADE), it starts with many replications of a significant control agent as needed. The replicated agents are software based and modulated, thus it is easy to apply. All of the replications arrange themselves into a logical ring. Whenever one of the replications fails, the others will notice and act accordingly by using cross-notification. Agents connecting to the failed replication will be able to connect to some other replications and keep all of the information as the same as before the damage happens.

## Multiple Sectioned Dynamic Bayesian Networks (MSDBNs) Inference Engine

Multiple Sectioned Dynamic Bayesian Networks (MSDBNs), as a distributed, dynamic, probabilistic inference engine, can be embedded into the control architecture to handle uncertainties of general large-seale complex systems. MSDBNs decomposes a large knowledge-based system into many agents. Each agent holds its partial perspective of a large problem domain by representing its knowledge as a dynamic Bayesian network. Each agent accesses local evidence from its corresponding local sensors and communicates with other agents through finite message passing. If the distributed agents ean be organized into a tree structure, which satisfies running intersection property and d-sep set requirements, globally consistent inferences are achievable in a distributed way. By using different frequencies for local DBN agent belief updating and global system belief updating, it balances the communication cost and inference global consistency. In this research, fully factorized Boyen-Koller (BK) approximation algorithm is used for local DBN agent belief updating at the distributed is used for local DBN agent belief updating.

## Modeling and Simulation Environment

Multi-agent based control model with distributed multiple sectioned dynamic Bayesian network inference engine has been established for a simplified chilled water system. This simplified chilled water system includes one chiller-pump plant and two service loads.



Figure 1: The Entire Test Model in ModelCenter Analysis View

An integration environment shown in Figure 1 has been developed by using ModelCenter® of Phoenix Integration to test the proposed methods. The integration model includes five modules: Scenario. ABCtrl. CWS. TES and ScenarioDefinementAndResultCollection. Scenario module transforms the scenarios defined in ScenarioDefinementAndResultCollection module into the format which is compatible with CWS module created in Flowmaster. ABCtrl includes HyMABC and MSDBN, which consist of dozens of control agents and three Bayesian network agents. All of the agents are established in JADE which is completely implemented in Java language, while CWS simulates fluid network which balances energy, pressure and mass flow rate of fluid. TES is a thermoelectric model developed in MATLAB Simulink and it also includes low level feedback controllers. ScenarioDefinementAndResultCollection is implemented in Excel worksheet. It defines the scenarios, collects the simulated results and visualizes the results.

### **Result Analysis**

By using the integrated model, three scenarios have been tested and analyzed.

### • Scenario 1 (Nominal Conditions):

Assumptions: all of the components are not damaged; every flow rate point in the Bayesian network is observable; every component open degree is observable; resource eapacity is 0.8kg/sec; the initial temperatures of service load 1 and service load 2 are 317 Kelvin and 400 Kelvin respectively. For the nominal case, the control system can make the right decisions and distribute the resource to different service loads accordingly.

### • Scenario 2:

Assumptions: all of the flow rates listed in the Bayesian network are not observable; every component open degree is observable; resource capacity is 0.8kg/sec; valve7 becomes STUCKCLOSE at time t = 440 sec (the 11th iteration); valve11 becomes STUCKCLOSE at time t = 840 sec (the 21st iteration); resource capacity is 0.8kg/sec; the initial temperatures of service load 1 and service load 2 are 317 Kelvin and 400 Kelvin respectively. For this case, the results shows that without any flow rate observation and only with component open degree observations, the inference engine can detect component damages quick enough and the control system can reconfigure the whole system by switching from damaged components to their corresponding redundant ones to redistribute system resource accordingly.

#### • Scenario 3:

Assumptions: only the flow rates of the points located in the upstream of valves in service loads and listed in the Bayesian networks are observable; valve open degrees are observable only for valve1, valve2 and valve7. Pumps and chiller operation states are observable. Valve 7 is STUCKCLOSE at time  $t = 440 \sec$  (the 11th iteration). Valve 11 is STUCKCLOSE at time  $t = 840 \sec$  (the 21st iteration); resource capacity is 0.8kg/sec; the initial temperatures of service load 1 and service load 2 are 450 Kelvin and 400 Kelvin respectively. For this case, the results show that it is hard to detect component state changes by only using flow rate observations, which is due to cyclic characteristic

of the fluid network. Fortunately, for a smart valve, its open degree is one of the output signals.

## Task 3: Surrogate Modeling of Dynamic Systems

After the development of the surrogate model with a block structured NN and its successful implementation to simple nonlinear RLC modeling as proof-of-concept, the dynamic surrogate modeling method based on the block structured NN is eurrently being used for creating component models of the chilled-water model of the notional YP. In this report, one of the component models was ehosen in order to eonduet a performance eomparison between the surrogate model with plain single hidden layer (SHL) NN and the surrogate model with the block-structured NN. The model is shown in Figure 2 and its specification is listed in Table 1.



Figure 2: Model of a Heat Exchanger Unit

**Table 1: Model Specification** 

Pipeline	Heat exchanger	Valve
Tot. pipe length: 12 ft	Pipe area: 0.197 in2	Diameter: 0.5 in
Diameter: 0.5 in	Loss coefficient: 1.5	
	Hydraulic diameter: 0.5 in	

In Figure 2, Q is volumetrie flow rate,  $O_v$  denotes valve opening ratio, and  $\Delta P$  represents the pressure difference between the two ends of the system. The model was originally ereated using FlowMaster® V7.

Table 2 shows the differences in the structural configuration of the two NNs. The major difference is that the bloek-structured NN has two hidden layers instead of one, and the neurons at the first layer have the linear activation function. The number of nodes at the linear layer (i.e., the first layer) should be the same as the dimension of the system state variables (and the dimension of the final outputs from the NN in this formulation), so only one node was assigned. A graphical representation of the bloek-structured NN will be, as a result, very similar to that shown in Figure 3, in which a "bottleneck" structure ereated by the linear layer is elearly found. The same number of nodes was assigned in

the nonlinear layers of both NN structures so that the two NNs have at least the same potential capability of nonlinear function approximation.

	Plain NN	Block structured NN Double hidden layer			
37-4-4	Single hidden leven				
Net structure:	Single nidden layer –	Layer 1:	Layer 2:		
Activation functions:	Hyperbolic tangent	Linear	Hyperbolic tangent		
Input variables:	$Q(t-1), \Delta P(t-1), O_v(t)$	$Q(t-1), \Delta P(t-1)$	$O_{v}(t)$		
No. of hidden nodes:	10	1	10		
Degree of freedom:	51	45			
Output variables:	Q(t)	Q(t)			





Figure 3: Block Structured NN

The comparison was done in a following way. A training data set and two test sets were generated from original computer model, with arbitrary changes on both Ov and  $\Delta P$  over time. For each NN structure, five NN-based surrogate models were made using the same training set. For training of each model, 500 epochs and  $1 \times 10^{-6}$  MSE were set as the stopping conditions, and the training process stopped when any of the two conditions met. As performance is measured, MSE from one of the two test sets was measured to evaluate model approximation accuracy, and the training time and the number of epochs are used to assess training efficiency. All the NN implementation was made in Matlab®, and Levenberg-Marquardt method was chosen as the training algorithm. The results are shown in Table 3 and Table 4.

Table 3	: Tra	aining	Result	of	Plain	NN
---------	-------	--------	--------	----	-------	----

Trial No.	1	2	3	4	5	Average	Best
Training set MSE:	3.5609×10 <sup>-6</sup>	8.8035×10 <sup>-6</sup>	4.6208×10 <sup>-6</sup>	3.4656×10 <sup>-6</sup>	3.2052×10 <sup>-6</sup>	4.7312×10 <sup>-6</sup>	3.2052×10 <sup>-6</sup> (No. 5)
Test set MSE:	3.0385×10 <sup>-4</sup>	3.1160×10 <sup>-4</sup>	2.6629×10 <sup>-4</sup>	2.9953×10 <sup>-4</sup>	2.8758×10 <sup>-4</sup>	2.9377×10 <sup>-4</sup>	2.6629×10 <sup>-4</sup> (No. 3)
Training time (sec):	104.8	104.4	104.4	104.4	104.4	104.5	104.4
Epochs:	500	500	500	500	500	500	500
Training stopped by:	Max. epochs						

Trial No.	1	2	3	4	5	Average	Best
Training set MSE:	2.7326×10-6	1.7462×10 <sup>-6</sup>	9.9629×10 <sup>-7</sup>	9.9897×10 <sup>-7</sup>	9.9347×10 <sup>-7</sup>	1.4935×10 <sup>-6</sup>	9.9347×10 <sup>.7</sup> (No. 5)
Test set MSE:	2.1417×10 <sup>-4</sup>	7.2219×10 <sup>-5</sup>	9.4172×10 <sup>-4</sup>	9.0177×10 <sup>-5</sup>	1.1590×10 <sup>-4</sup>	2.8704×10 <sup>-4</sup>	7.2219×10 <sup>-3</sup> (No. 2)
Training time (sec):	37.2	37.3	15.9	28.1	6.2	24.9	6.2
Epochs:	500	500	212	371	80	332.6	80
Training stopped by:	Max. epochs	Max. epochs	Error criterion	Error criterion	Error criterion		

**Table 4: Traing Result of Block-Structured NN** 

Based on the results above, it indicates that the NN models with the block-structure seems to outperform the plain NN models in both training efficiency and model accuracy, except that the average values of the test set MSE almost tie though the best test set MSE of the block-structured NNs was about 3.7 times better than that of the plain NNs.

Thus, in order to further observe the performance in model accuracy, another test set was employed for simulation using the models from the two different structures. To have a better visualization of the simulation results, only the NN models with the best training set MSE and the best test set MSE were picked from each of the two groups to demonstrate the simulation tests. The plots of the simulation results are presented in Figure 4 and Figure 5.



Figure 4: Simulation Result of Plain NN Models, Using another Test Set



Figure 5: Simulation Result of Block-Structured NN Models, Using another Test Set

Unlike the previous results obtained through the simple RLC circuit model case, the plain NN models provided fairly good model stability and simulation accuracy, which may be because most fluid-system components have the inherent monotonic, fist order-like dynamics. However, as shown in Figure 4, the model also had strangely high errors in the period from about 1.2 seconds to 3 seconds of the simulation time, where the valve opening ratio, one of the inputs to the model, was relatively low.

On the other hand, the NN models with the block-structure delivered very good simulation accuracy over the entire simulation time, as can be seen in Figure 5. However there was one odd aspect in the result of the block-structured NN models too, which was the high overshoot of the model output at about 1.7 second of the simulation time where the valve was suddenly closed completely. With a few more manual tests, it was found that such a high pitch error could disappear or become negligible if the valve input was changed more gradually than a sudden step-type change to the complete closure. Expecting such an unrealistic step input is not applied in any actual simulation run, these models may be still valid enough to use. If still necessary, corrections can be made easily by many ways, one of which is just including a rule such as 'Q(t) = 0 if  $O_v(t) = 0$ ' that overwrites the result from the models.

## Task 4: Human in the Loop Control

After an investigation of several software frameworks a suitable framework has been selected and a migration plan put in place. Given the following criteria the most appropriate framework is a product from Adobe called Flex. The product is a mixture of the Adobe Action script programming language and a markup language call MXML. Flex applications can be complied into byte code compatible with the popular Flash Player.

- Built in cross platform compatibility.
- More advanced support for visual effects as an enabler of visual analyties.
- Additional support for a more modular software design.
- Capable of fast, responsive, and intricate interfaces.

In addition to the selection of the software framework, a database engine has been selected to facilitate the information management of the IRIS system simulation and design. MySQL has shown in a simple proof of eoncept that it is more than capable of handling the load demands of the simulation environment. It is simple to integrate with a variety of frameworks and platforms, and touts the strengths of the structured query language. Currently it is one of the most commonly used database engines. It is possible that its suitability can change over time as Oracle has recently acquired MySQL. PostgreSQL has been selected as a fall back option. As a measure to ensure compatibility with both choices, feature usage has been earefully selected as the HMI database libraries are being written.

# Task 5: A Methodology for Improving the Mission Effectiveness in Complex Systems Design

One of the main objectives of IRIS is to deliver a conceptual design methodology for more survivable and mission effective ships. There can be different underlying philosophies, based on which an improved design solution can be returned. Traditional design approaches would focus on performance, while modern approaches will seek for more robust solutions, either through enhancing safety or adding automation or intelligence. Beyond the development of methods that allow for the discovery of such solutions, studies can also be proposed to investigate the tradcoff of cost vs. effectiveness aeross solutions representing different underlying design philosophies.

Traditional design approaches are based on optimizing naval system architecture for performance, based on a very limited number of mission scenarios. However, such point solutions will only yield responses that maximize survivability mostly for scenarios similar to the ones that have been used for performance optimization.

While the traditional design approach is conceptually fairly simple and straightforward, it does not really address any issues regarding the uncertainty around naval system mission requirements, environmental condition or even the capability of the system to perform as designed under real operations. Moreover, it cannot guarantee that a feature that is an absolute best for a particular threat and operational situation, might not be the best against the range of threats and operational situations it may encounter. A robust solution will represent a system that in theory would be better prepared to perform multiple mission aets and withstand a larger spectrum of unexpected events. At the same time, preseribed design performance might not be optimal, in order to compensate for the multi-mission eapability (e.g., preferred extra weight for redundant systems over maneuverability).

It is quite certain that robust design can offer system designs that are capable of surviving under various mission tasks and hazardous threat environments. However, the question at this point is how exactly the multi-mission capability and the enhanced survivability are enabled. Typical survivability enhancement features, such as component redundancy, separation and shielding are immediate techniques that can be properly applied to the design based on conceptual sizing. Real time simulations of systems operations can be also available for the sizing and decision making on selecting system architectures. This is still a form of robust design, yet through a more reactive approach to how hazards and environmental uncertainty affect system effectiveness.

A new philosophy has been recently emerging and seeks to address the aforementioned concerns. Resilience engineering is a novel and relatively recent form of philosophy on understanding threats, accident and damage propagation, as well as how a system should be designed to conform to changes that occur around it, for the purpose of withstanding adverse effects and maintaining its mission effectiveness. In other words, a resilient system can adjust its functioning prior to or following changes and disturbances so that it can go on working even after a major mishap or in the presence of continuous stress, mainly by being able to be proactive on safety.

Resilience engineering can offer insight and research directions that may lead to answers regarding the design of more safe and survivable complex systems. According to the systemic view of how accidents occur, one can infer that a resilient response by the system would include the ability to efficiently adjust to non-favorable influences rather than to resist them. Such ability could be embedded as collection of internal functionalities and be the basis for certain active features for susceptibility/vulnerability reduction and recoverability increase. Automation and networks of sensing grids and information distribution might be possible enablers for enhanced reconfigurability and would lead to the essential functionality of a resilient system.

The overall problem though, relies on investigating possible methods for improving system and mission effectiveness. According to the Defense Appropriation Act of 2004, effectiveness can be improved by including survivability in the design process as a key performance parameter. The current United States Navy standard is primarily determined by the Survivability Design Handbook for Surface Ships (OPNAV P-86-4-99). According to this procedure, survivability is improved by focusing on vulnerability and applying standard design principles such as subsystem redundancy or separation. Other common tools that are employed are the deactivation diagrams that are similar to fault tree diagrams in reliability studies.

The fundamental rescarch question regarding this initiative would be how to improve the design the system, so that system effectiveness through survivability is maximized for a given set of scenarios, which will include system damage and/or restoration events. Moreover, it can extend to consider how the philosophy of resilience engineering can translate into a systems engineering method, involving various aspects, such as accident and damage modeling or system functionality and possible enablers, in order to fit into the bigger picture of more survivable systems in a highly uncertain mission environment.

Based on earlier work, it is assumed that there has been a clarification of how a robust solution would differ from the resilient solution. Most of current ship design methods are based on traditional design methods, yet robustness of solution is ensured through optimizing the architecture for multiple scenarios. However, nothing ensures that a robust solution is obtained with the use of a systemic accident model or, or that no significant excess of weight has been added due to the highly redundant subsystem components. But, more importantly, it is quite challenging to demonstrate how a robust architecture can be proactive through its embedded functionality, to be also recognized as a resilient system.

Furthermore, it is assumed that a first iteration of a method for resilient systems design will have been formulated. The main focus for this task should become the fine-tuning of this method and demonstrating its superiority if compared to the robust solution for the same mission scenario. At the same time, it is expected that there will be a costeffectiveness tradeoff that could be investigated through the assessment of safety and survivability improvements against any performance degradation for both design approaches.

Based on the objectives stated earlier, a central hypothesis can be formulated and the proposed subtasks will aim towards supporting it. The hypothesis states the following:

A more resilient system demonstrates improved survivability than a robust system, mainly by incorporating engineering system reconfigurability, if subject to the same intelligent or natural events that affect system operations.

Improved safety and survivability come at some expense in overall system performance, acquisition and maintenance costs.

Before providing the outline of tasks that is combined should support this hypothesis, the following implications can be extracted:

- Robust systems can be survivable mainly through reduced vulnerability, yet without significant potential in active susceptibility reduction and recoverability enhancement.
- Resilience engineering suggests a collection of modern concepts that could potentially improve active survivability, mainly through the development of reconfigurable systems.
- While resilient systems are expected to be more survivable than robust systems under the same threat environment, it might be that such benefit will be at the cost of degraded system performance and higher acquisition and maintenance costs

Optimize two system architectures, using robust and resilient systems design respectively

With this subtask, two different approaches are adopted for delivering two alternative optimized solutions, starting from the same baseline. The common baseline is a version of a Yard Patrol craft (YP) that will be augmented for survivability improvement through

susceptibility, vulnerability and recoverability reduction. A general template of the method adopted is presented with Figure 2.

The robust design approach will mostly focus on vulnerability reduction, through the usage of more reactive technologies and naval architecting enhancements. Some of them are involving redundant components, strategic placement, sophisticated zonal design, lighter materials and enhanced shielding. To a great extent, robust design is traditional naval architecting, with improved systems engineering to satisfy more stringent safety requirements and decision making for on selecting the optimal solution, based on multimission operations simulation.

The resilient design approach is a robust solution to a great extent, yet it requires the system to be more proactive for withstanding and recovering from a threat and its resulting events. There ean be various enablers that will offer this capability to the system. Reconfigurability seems to be the most feasible alternative for making a system more resilient. This can be achieved with controllers that will support a series of automated functions for sensing, analyzing and selecting an appropriate plan for withstanding and neutralizing the effects of the threat. Moreover, the implementation of a systemie accident and damage propagation model through real time modeling and simulation can be instrumental in identifying additional modes of failure and damage that ean be taken into account in the survivability assessment and improvement of the architecture.

The experimentation and design framework has been structured to support design space exploration, systemie damage and accident modeling, physics based simulation for eapturing system behavior and includes "placeholders" for importing different reconfiguration strategies through intelligent algorithms and selecting the most suitable for a given architecture.



Figure 6: General template of robust and resilient design methods

As a baseline, a notional naval ship design is required to be the starting point for the implementation of the method. A synthesis and sizing tool is used for generating the geometry and the inner systems distribution. Paramarine is the software that has been used to create this baseline. It would require a certain amount of information for the creation of a ship baseline, such as ship geometry, engineering subsystems, aequisition and operations cost breakdown, mission profiles, threats and hazards and local environmental conditions.

The damage prediction module is responsible for analyzing and visualizing the damage propagation throughout the particular architecture. Based on the three different types of aceident modeling, this module is more of a combination of a linear damage model and systemie. It is using DOMINO, a tool based on the theory of deactivation diagrams for initial damage prediction, based on the given single points of failure and subsystem connectivity. This module is linked to an M&S environment that simulates the operations of the ship's engineering plant, including the power generation and the cooling system with their corresponding controllers. In other words, the physics-based simulation represents the systemic model of failure prediction and is exchanging information with the deactivation diagrams at the end of every time step. Fire due to overheat and compartment flooding, are both expected to be part of damage modeling.









A set of responses at different operational levels have been defined and require data that will be provided by the output of the simulation. Such metries are the figures of merit for the particular design solution representing the corresponding architecture and will determine its performance based on survivability and mission effectiveness criteria. At the subsystem level, subsystem performance measures can be obtained (e.g. voltage outputs, coolant mass flow rates). Given a scenario per system configuration, system sensitivities and correlations of measures of performance (MoP) to scenario changes can be identified. Such measures are mostly conditional probabilities of achieving an outcome response, given events that occurred earlier, as defined by the scenario event tree analysis. By identifying the direction of improvement and exploring the design space, multiple iterations can be performed around the baseline to achieve a solution that satisfies the original design requirements.

Survivability mission effectiveness assessment is the next task that will enable the eosteffectiveness tradeoff for each solution. Despite the fact that some steps of a survivability assessment process have been already utilized for improving the solution at the design process, the objective of this task is to evaluate the complete solution. The template for the evaluation process is the Total Ship survivability Assessment Method (TSSA), an overview of which is provided at the Figure 8.

It should be expected however, that while the resilient solution demonstrates improvement in terms of safety and survivability, it might also incur increased development and maintenance costs. A similar case is expected for the robust solution, yet it should be investigated whether the extra costs for moving from a robust to a resilient system can justify the safety improvements and at what levels of mission performance degradation. For a mission with given outcomes, the integration of MoP to MoEs should look like Figure 9.



**Figure 8:** Total Ship Survivability Assessment Method (TSSA)



Figure 9: Probabilistic distribution of scenario outcomes as calculated by the TSSA method