Shilpa B. Ganesh, Karl Schoder, Hong-Jian Lai, Amer Al-Hinai, Ali Feliachi

Energy Management System with Automatic Reconfiguration for Electric Shipboard Power Systems

ABSTRACT

The automatic reconfiguration of electric shipboard power systems is an important step toward improved fight-through and self-healing capabilities of naval warships. The improvements are envisioned by redesigning the electric power system and its controls. This research focuses on a new scheme for an energy management system in form of distributed control agents. The control agents' task is to ensure supply of the various load demands while taking into consideration system constraints and load and supply path priorities. A self-stabilizing maximum flow algorithm is investigated to allow implementation of the agents' strategies and find a global solution by only considering local information and a minimum amount of communication. A case study using the distributed agents within a multi-layer system architecture to function as energy management system is presented.

INTRODUCTION

The success of increasingly all electric commercial ships has sparked interest to incorporate the advantages of improved electric power systems in naval warships. Design of shipboard power generation, distribution, communication, propulsion, and control is dictated by the demand to reduce manning and costs and is step-by-step replacing mechanical-hydraulic systems by electric solutions. This process will ultimately also improve the survivability of the envisioned all electric warship.

The Naval Combat Survivability (NCS) generation and propulsion and DC distribution testbeds have been developed as a common base for modeling the electric warship (Pekarek et al. 2003 and Sudhoff et al. 2003). The remaining control challenge requires advanced control strategies to ensure system stability and to take advantage of the system's capabilities (Zivi and McCoy 1999.) The controls to be developed include local controls for the various power electronic devices and the automation of processes to help system operators in coping with routine operations, maintenance, and emergency situations to optimize life-time and cost performance. The necessity of improving warships is the bottom line of several incidents in which the electric power was lost through only a single incident (Tucker 2001). The answer to this problem is envisioned in dependable automation strategies (US Navy ONR/NSF EPNES 2002) that for example extend nonlinear control theory, apply analytic redundancy, utilize distributed intelligence to form robust networks, and allow reconfiguration based on situational awareness. The realization of these goals is hindered by several technological shortcomings including incomplete theoretical basis and limited situational information. The challenge of providing a new and distributed answer for a decentralized energy management system of the electric shipboard power system is addressed here.

The energy management system to be developed here should help solve the problem of providing the various loads with electric power. This problem is usually known as the power flow problem and can be solved at a central place numerically using various techniques, for example Newton-Raphson algorithm to solve the non-linear set of equations and it can also be formulated as linear programming problem to allow incorporation of limits and priorities of loads and supply paths. These approaches have been investigated by several authors and can be found in Butler, Sarma, and Prasad (2001) and Butler-Purry and Sarma (2004). A different approach in solving the power flow problem for the energy management system is taken here by using spatially distributed agents. These agents make local decisions to reach a globally acceptable solution with limited amount of communication.

The use of software agents gained popularity by recent advances in software engineering to construct distributed systems that cooperate to reach a common goal (Weiss 1999). This new philosophy to implement decentralized control algorithms is discussed in this paper and envisioned improvements and advantages of a multi-agent based control framework are applied to build the energy management system.

The distributed agent concept needs to be complemented by an appropriate framework for implementing automatic reconfiguration. Graph theory provides a formal basis to represent the distributed control system and to develop algorithms for a decentralized energy management solution. Maximum flow algorithms have been investigated to find an answer to this challenge.

The following sections present an introduction to the electric shipboard power system as the domain of application for the energy management system, multi-agent systems, the maximum flow problem, design of the energy management system, a case study to demonstrate the distributed and agent-based concept, and conclusions.

SYSTEM DESCRIPTION

The system used in the case studies has been described in detail in Pekarek et al. (2003) and Sudhoff et al. (2003). A simplified representation of its components has been implemented to avoid excessive simulation times due to unnecessary details concerning the power electronic components. An overview of this system in form of a block diagram is given in Figure 1. The distribution testbed is fed by two AC sources represented by gas turbines, synchronous generators, and exciters. The propulsion systems are connected to the AC system using power electronics for a flexible drive system. The AC is converted to DC for distribution of the electric power by two DC buses, the port-bus and starboard-bus. To improve the survivability, the loads on board are grouped into zones and each zone is supplied from both of the DC buses by additional DC/DC converters. This allows a certain redundancy as well as graceful performance degradation as the converters are used to actively limit currents and can be used to isolate faulty loads or entire parts of the system. Also, the converters can be operated to supply loads from either or both buses and to switch continuously between them. The loads represent typically encountered power demands including induction motors, three-phase AC-loads, and constant power loads.

All components are modeled by transfer functions of first- and second-order to represent input-output relationships concerning active power and voltage values. The components' transfer functions and parameters are given in the appendix. The following sections discuss the agent based framework and the maximum flow algorithm used to create the energy management system.

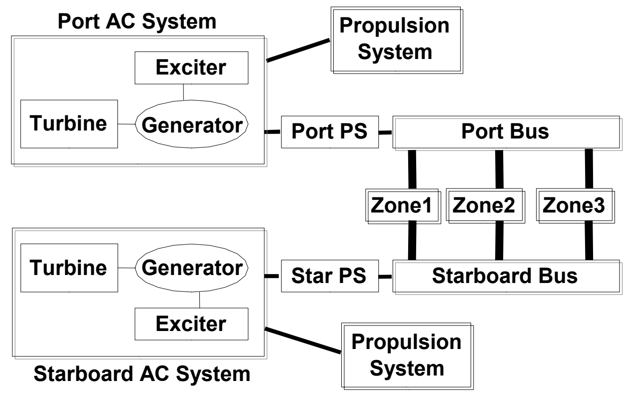


Figure 1. Shipboard power system

MULTI-AGENT SYSTEMS

A single agent is a system that is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objectives (Weiss 1999). Agents are capable of flexible autonomous actions through reactivity - perceive and respond in a timely manner, pro-activity - goal-directed behavior by taking initiative, and social ability - interacting. A multi-agent system is a system composed of a population of agents, which interact with each other to reach common objectives, while simultaneously each agent pursues individual objectives (Ferber 1999).

The agents form a loosely-coupled network of problem solvers and work together to solve problems that are beyond their individual capabilities. Together the agents increase fault-tolerance as inherently distributed mechanism and thus a system made of autonomous agents will not collapse when one or more of its components fail. The modular approach creates a scaleable architecture and agents become powerful entities because of the factorization of the problem they provide. Each agent can be identified as an entity (e.g., a machine, a plant, a tool or a part) and thus help in incremental growth and flexible expansion. The advantage of scalability is provided as each agent can join a system, start working with other agents, or just leave a system once it has finished a plan it was engaged in without effecting the operation of the system.

Agents have the capability to reconfigure themselves. This is an important advantage for systems that must respond to a wide range of different conditions. Because each agent is in close contact with the real world, the system's computational state tracks the state of the world closely and without the need for a centralized database. As the overall system behavior emerges from local decisions, the system readjusts itself automatically to changes in the environment or the removal of other agents. Thus a fully functional self-configuring system can be effectively implemented by merely networking agent-based resources.

The communication among agents can be achieved by different means. The work presented in this paper makes use of the blackboard architecture as described in Weiss (1999). The blackboard is, for example, used to notify other agents of their current status with respect to being active making local changes or running idle.

MAXIMUM FLOW PROBLEM

A fundamental problem in graph theory is the maximum flow problem. Many different sequential and parallel algorithms have been developed (Ahuja, Magnanti and Orlin 1993). All the parallel algorithms but the one in Gosh, Gupta, and Pemmaraju (1997) present implementations that do not run in polylogarithmic time on a polynomial number of processors. Other advantages of the Gosh, Gupta, and Pemmaraju's algorithm are: simple, passive, self-stabilizing, and local checking and corrections only. The algorithm has been investigated here to find a solution to the power flow problem of the electric shipboard system. The following briefly defines the maximum flow problem and notation used. More details can be found in Gosh, Gupta, and Pemmaraju (1997).

An directed graph (digraph) $G = (V,E)$ with $n = |V|$ number of nodes and $e = |E|$ number of edges is shown in Figure 2. The graph should not contain any directed cycles of length 2: if $(i,j) \in \mathrm{E}$ then $(j,i) \notin E$. The nodes s and t are the distinct source and sink nodes, respectively. The source node s has no incoming edges, the sink node t has no leaving edges. Every edge (i,j) is associated with a real-valued *flow* $f(i,j)$ and a non-negative real-valued *capacity* $C(i,j)$. Any feasible flow f in G obeys the flow conservation constraint, i.e., the *incoming flow* $I_f(i) = \Sigma_{(j,i)\in E} f(j,i)$ equals the *outflow* $O_f(i) = \Sigma_{(i,k)\in E} f(i,k)$:

$$\text{for all } i \in V - \{s,t\}: I_f(i) = O_f(i).$$

Flows are also satisfying the skew symmetry property of $f(i,j) = -f(j,i)$. The maximum flow problem is to maximize the outflow of the source node $O_f(s)$.

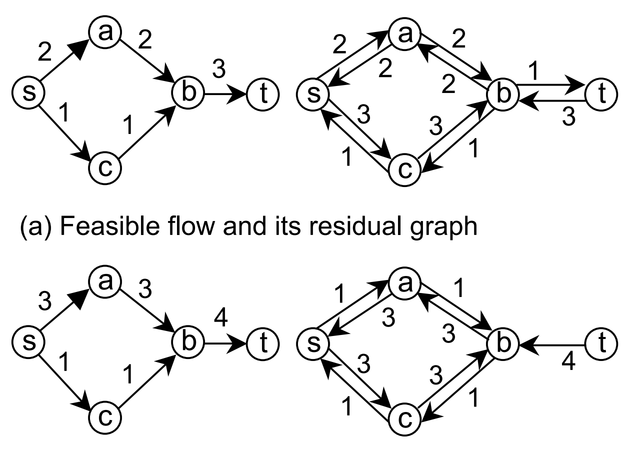
Definition 1. For any flow f and for each pair of nodes $(i,j) \in V \times V$, the *residual capacity* $r(i,j)$ is equal to $C(i,j) - f(i,j)$.

Definition 2. For any flow f in G, the *residual graph* G_f of G with respect to f is defined as the weighted digraph $G_f = (V_f,E_f)$, where $V_f = V$ and $(i,j) \in E_f$ if and only if $r(i,j) > 0$.

Definition 3. A directed path in the residual graph G_f from s to t is called an *augmenting path*.

Example: A feasible flow of a graph and its residual graph are given in Figure 2a. All edges have a capacity of 4 and $C(s,a) = C(a,b) = C(s,c) = C(c,b) = C(b,t) = 4$. The number on an edge indicates the flow along an edge, e.g., the flow from node b to the target node t $f(b,t) = 3$. The residual graph for the flow gives the remaining capacity, for example the residual flows $r(b,t) = C(b,t) - f(b,t) = 4 - 3 = 1$ and $r(t,b) = C(t,b) - f(t,b) = 0 - (-3) = 3$.

A maximum flow and its corresponding residual graph are shown in Figures 2b, respectively. It has been achieved by increasing the flow on the augmenting path s-a-b-t by its minimum residual capacity of $r(b,t) = 1$. Note that after increasing the flow the updated residual graph contains no augmenting path and, therefore, the flow from the source to the target cannot be increased further.



(a) Feasible flow and its residual graph

(b) Maximum flow and its residual graph

Figure 2. Example graph with edge capacities of 4.

```
Nodes' algorithm: for agent i, i ≠ s
begin
    {GS1} : d(i) ≠ min(D(i) ∪ {n})      ? d(i) := min(D(i) ∪ {n})
    {GS2} : demand(i) < 0               ? Reduce_Inflow(i)
    {GS3} : ∃ j ∈ INP(i): pull(j,i)     ? f(j,i) := f(j,i) + min(demand(i), r(i,j))
    {GS4} : push(i)                     ? Reduce_Outflow(i)
    {GS5} : ∃ j ∈ LIMIT(i)              ? f(j,i) := f(j,i) - (f(j,i)-C(j,i))
    {GS6} : 1                           ? Update_Priority(i)
    {GS7} : change(i)                   ? Notify(i)
 end
```

```
Procedure Reduce_Inflow(i)
 begin
   Find (k,i) ∈ E such that f(k,i) > 0
   f(k,i) = f(k,i) - min(-demand(i), f(k,i))
 end
```

```
Procedure Reduce_Outflow(i)
 begin
   Find (i,k) ∈ E such that f(i,k) > 0 ∧ k ∈OUTP(i)
   f(i,k) = f(i,k) - min(demand(i), f(i,k))
 end
```

```
Procedure Update_Priority(i)
 begin
   (i ∈ LN) ∧ demand(i) = 0 : p(i) =  -1
   (i ∈ LN) ∧ demand(i) > 0 : p(i) =  P(i)
   (i ∈ TN) ∧ (j|(i,j)∈E)   : p(i) = max(p(j))
 end
```

```
Procedure Notify(i)
 begin
   Blackboard_Send(status(i))
   history(i) = status(i)
 end
```

```
Node s: idle with d(s) =0, Node t: same but demand(t) = ∞, initially history(i) = 1
```

```
Notation:

 demand(i) = Of(i) - If(i)
 IN(i)   = {j | (j,i) ∈ Ef
 INP(i)  = {j | (j ∈ IN(i) ∧ p(j)=max(IN(i))}
 OUT(i)  = {j | (i,j) ∈ E }
 OUTP(i) = {j | (j ∈ OUT(i) ∧ p(j)=min(OUT(i))}
 LN      = {j | (I,t) ∈ E}
 TN      = {j ∈ V - {s,t,LN}

D(i)      = (d(p) + 1 | p ∈ IN(i)}
pull(j,i) = (demand(i) > 0) ∧ (d(i) < n) ∧
            (d(j) = d(i) - 1)
push(i)   = (demand(i) > 0) ∧ (d(i) = n) ∧ (i ≠ t)
LIMIT(i)  = (j | f(j,i) > C(j,i) ∧ (j,i) ∈ E)
p(i)      = (i ∈ V - {s,t} : max(p(OUT(i))))
status(i) = (G1 ∨ G2 ∨ G3 ∨ G4 ∨ G5)
change(i) = status(i) ≠ history(i)
```

Figure 3. Maximum flow algorithm

THE SHIPBOARD's POWER FLOW AS MAXIMUM FLOW PROBLEM

The electric shipboard power system is modeled as an acyclic directed graph and its residual graph. The transformation of the block diagram into the acyclic graph is achieved by representing physically linked components by neighboring nodes in a graph. Each of the edges between nodes can be associated with a direction because the electric power flows only from the AC generators toward the loads. The multiple generators (sources) and loads (sinks) can be accommodated by introducing supersource and supersink nodes with additional edges. This is a common step in maximum flow algorithms and does not introduce any limitations (see for example Ahuja, Magnanti, and Orlin 1993). The edges' capacities from the supersource and to the supersink nodes represent the upper bounds on generation capabilities and load demands, respectively. Each agent should perform locally by observing its environment and taking corrective actions to satisfy its goals, i.e., to request power for a load, to route the power flow according to load priorities, etc. The moves themselves are implemented by the agents without synchronization. By making local and asynchronous moves, tolerance to transient faults is achieved. The algorithm published by Gosh, Gupta, and Pemmaraju (1997) provides a framework for the desired agents' behavior. It allows the design of agents that adjust to dynamic changes in network topology and edge capacity. Its computational model and changes made to accommodate requirements of the energy management system are presented next.

Computational model

Each edge (i,j) corresponds to a physical directed link and each node i in graph G is represented by an agent i. Every agent contains a finite set of local variables that can be read and written by itself but only read by its neighbors. Agent i executes a program (algorithm) asynchronously that can be expressed by the conditional execution of actions:

```
do
  G1 ? A1 (G ... guard, A ... action)
  G2 ? A2
  ...
end;
```

Each action A_i modifies the agent's local variables. The guards G_i are boolean functions of agent i's local variables. If more than one guard is activated then only one is randomly chosen and its corresponding action executed. Afterwards the cycle begins again by evaluating the guards before an action is chosen.

Agents' maximum flow algorithm

The agents represent the nodes in the acyclic digraph G. The flow on the edges between nodes i and j is stored in variable $f(i,j)$. Two agents are neighbors in case an edge (i,j) exists. Only neighboring agents can communicate. Each agent can update the flow by directly modifying $f(i,j)$ or by adjusting a local variable in case the flow is represented by a difference of two positive values. Each agent keeps track of its believed distance value $d(i)$:

Definition 4. The *distance value* $d(i)$ of a node denotes the "believed" length of the shortest directed path from s to i in the residual graph G_f. The value of $d(i)$ is restricted to the range [0 ... n].

Every node in the graph is allowed to make moves by changing its distance value or flows. The algorithm executed by the agents at their nodes i ($i \neq s$) is given in Figure 3. The algorithm is based on seven guarded statements GS1 – GS7 that consist of their respective guard G_i and action A_i. The first six actions represent local steps taken to find a solution to the power flow problem. Guarded statement GS7 notifies other agents by publishing to the blackboard its status of making local changes or convergence to a local solution.

The algorithm is continuously executed by every agent $i \neq \{s, t\}$ in order to restore the $demand(i) = O_f(i) - I_f(i) = 0$. The two distinguished agents s (source) and t (sink) differ in their behavior from other nodes: Agent s is idle; agent t performs the same program as the other nodes but with a $demand(t) = \infty$. The agents' actions perform changes to increase or decrease their inflows or to reduce their outflows. Breadth-first search is used to update the believed shortest distance value $d(i)$. Agents with the belief that no direct path from the source to itself exists push back the demand on their outgoing edges. The following describes the guarded statements in more detail:

Guarded statement GS1: Every node i, $i \neq s$, updates its $d(i)$-value by determining $d(k) = \min\{d(j) \mid (j,i) \in E_f\}$. If $d(k) < n$ then node k is a predecessor of node i and $d(i) = d(k) + 1$, otherwise $d(i) = n$ and no direct path from the source to node i exists. The distance value of the source node is set to $d(s) = 0$.

Guarded statement GS2: Every node i, $i \neq s$, reduces the total flow along incoming edges in case $demand(i) < 0$.

Guarded statement GS3: Every node i, $i \neq s$, with $demand(i) > 0$ increases the flow on the highest priority and available incoming path $(j,i) \in E_f$ by the minimum of $demand(i)$ and $r(j,i)$.

Guarded statement GS4: Every node i, $i \neq s$, with $demand(i) > 0$ and $d(i) = n$ reduces the outflow on the lowest priority path $(i,j) \in E_f$.

Guarded statement GS5: Every node i, $i \neq s$, checks the flows on incoming edges $(i,j) \in G$ and reduces flows in case of capacity violations. Note that the above guarded statements never lead to this condition in case a valid initial flow existed. This statement rather allows changing system conditions including the loss of a link.

Guarded statement GS6: Every node i, $i \neq \{s,t\}$, updates its priority $p(i)$: nodes with all local demand (nodes connected to the supersink) satisfied set $p(i) = -1$, nodes with unsatisfied local demand set $p(i)$ to the respective load's priority value $P(i)$ where $0 \leq P(i) \leq 1$, and nodes without any local demand set $p(i)$ equal to the highest priority value of unsatisfied load demand along outgoing edges (i,j). This statement ensures that higher priority loads will be preferred throughout the system.

Guarded statement GS7: Every node i, $i \neq s$, publishes its status to the blackboard in case it has changed.

The target node with $demand(t) = \infty$ leads the agents towards a maximum flow solution.

IMPLEMENTATION OF THE ENERGY MANAGEMENT SYSTEM

The energy management system for the electric shipboard power system has been accomplished following the agent-based framework. The different system parts have been grouped into layers to form a logical architecture according to their functionality (see Figure 4.) The top layer represents the human-machine interface for the command and control center. This layer displays important system information and allows the system operator to communicate with system components, e.g., request for desired changes in propulsion power. It can also be used to simulate disturbances, e.g., loss of generation.

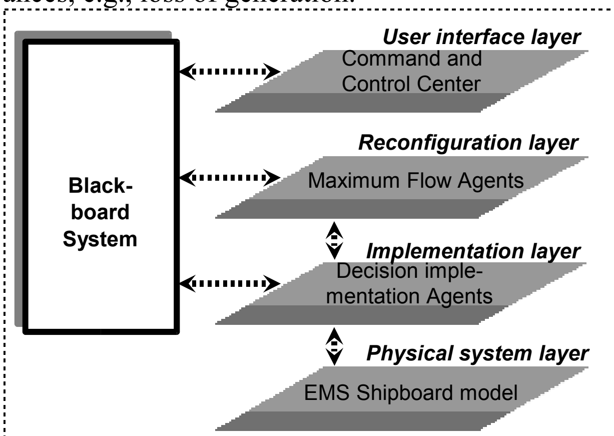


Figure 4. Layered architecture for the energy management system

The decentralized power flow solution using the maximum flow algorithm is part of the reconfiguration layer. The agents execute their programs asynchronously and locally but can communicate status and results (internal variables) using the blackboard. The implementation layer represents the decision implementation agents (interface points of these agents with the physical system are identified as "IA" in Figures A4 and A8.) These agents are notified when a new power flow solution has been agreed on by the agents of the reconfiguration layer. These agents access locally their respective maximum flow agents to inquire the solution to be implemented. Details concerning the code of the implementation of the human-machine interface and implementation agents have been omitted as we are primarily interested in the feasibility of the autonomous system reconfiguration. The lowest layer represents the mathematical model for the physical system. The shipboard model described earlier simulates the various components of the ship and their interactions. The components have local controls, e.g., proportional-integral controllers and their reference and parameter values, which are modified and adjusted by the decision implementation agents.

CASE STUDY

The energy management system has been tested for different system conditions, e.g., initial startup, changes in load demand, changes in edge capacity, loss of agents (nodes), increases in demand beyond generation and power transfer capabilities. Statistics concerning the number of agents' moves necessary for the "startup" scenario where initially no power is provided to any load is shown in Figure 1. Because the case study was performed on a single processor computer, all the moves made by agents have been executed in series. To be able to test convergence characteristics, each agent was randomly chosen from the list of available agents to perform once in one "negotiation round." This arrangement ensures that every agent gets a fair chance. The figure shows the minimum, average, and maximum number of actions executed by the agents in each of these "negotiation rounds" before convergence was achieved (for 10 runs). The command and control center used to supervise the energy management system is shown in Figure 6. The average number "negotiation rounds" required to determine a global solution is 14. The interface allows access to the reconfiguration and decision implementation layers and displays important information. It represents a convenient way for the system operator to, e.g., change system conditions. The current status of the reconfiguration layer and its agents is displayed using the graph concept and an example is shown in Figure 7. It represents the power flow solution for the "startup" scenario. The agents represent the following system components: "AC" – synchronous generator, "PS" – AC/DC converter, "CM" – Zonal DC/DC converter, "PROP" – propulsion, and "L" – load.

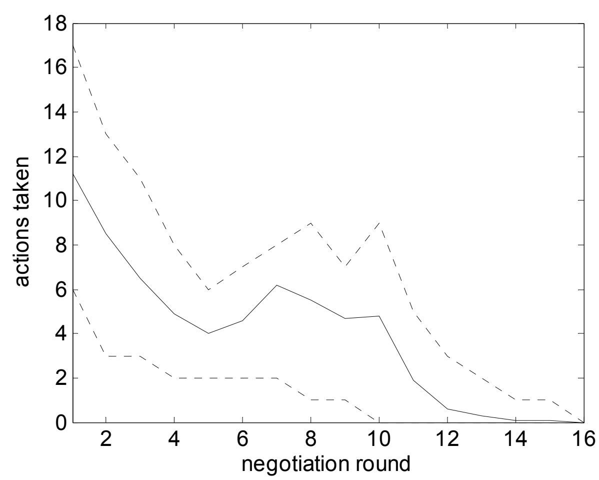


Figure 5. Number of moves for the "startup"-scenario

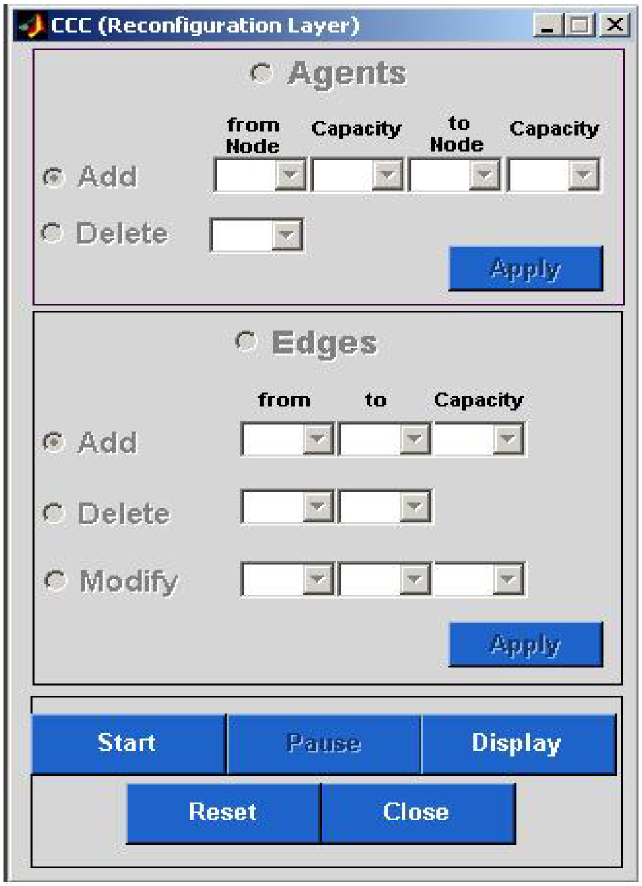


Figure 6. Command and control center (CCC)

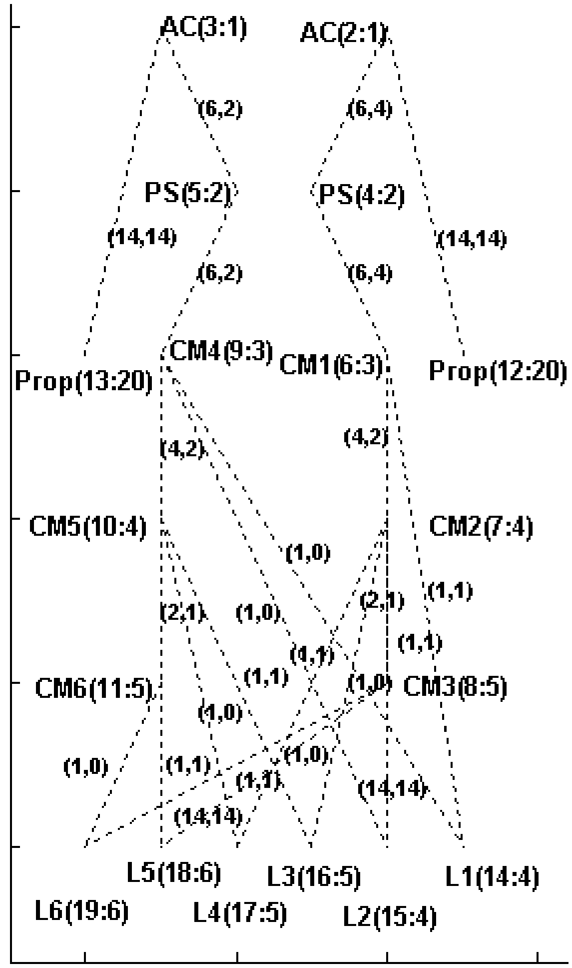


Figure 6. Graph representing the reconfiguration layer (nodes - number:d-value, edges – capacity:flow)

CONCLUSION

A new agent-based concept for the energy management system of shipboard power systems has been presented. A multi-layer architecture has been used to accomplish the functionality of a human-machine interface, automatic reconfiguration using a maximum flow algorithm, agents to implement reconfiguration decisions, and the mathematical model of a shipboard power system. Most of the actions necessary for the operation of the energy management system can be performed locally and, therefore, the energy management system requires only a limited amount of communication. The case study demonstrated the feasibility and flexibility of the concept and results are promising. The agents are able to adjust to changes in the system and allow the automation of actions usually performed by humans. Further research will focus on two different aspects: First, to broaden the application of agents to automate the shipboard operation, and second, to investigate tools that allow simulation of more details of the electric power systems while guaranteeing real-time performance.

REFERENCES

Ahuja, R.K., T.L. Magnanti, J.B. Orlin, *Network Flows – Theory, Algorithms, and Applications*, Prentice Hall, 1993.

Butler-Purry, K.L. and N.D.R. Sarma, "Self-Healing Reconfiguration for Restoration of Naval Shipboard Power Systems," *IEEE Transactions on Power Systems*, 19 (2): 754-762, May 2004.

Butler, K. L., N. D. R. Sarma, and V. R. Prasad, "Network reconfiguration for service restoration in shipboard power distribution systems," *IEEE Transactions on Power Systems*, 16: 653–661, November 2001.

Ferber, J., *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, Harlow, UK, 1999.

Gosh, S., A. Gupta and S.V. Pemmaraju, “A Self-stabilizing Algorithm for the Maximum Flow Problem,” *Distributed Computing*, 10: 167-180, 1997.

Pekarek, S.D, J. Tichenor, S.D. Sudhoff, J.D. Sauer, D.E. Delisle, and E.J. Zivi, “Overview of Naval Combat Survivability Program,” *Proceedings of the 13th International Ship Control Systems Symposium*, Orlando, Florida, USA, 2003.

Sudhoff, S.D., S.F. Glover, S.H. Zak, S.D. Pekarek, E.J. Zivi and D.E. Delisle, “Stability Analysis Methodologies for DC Power Distribution Systems,” *13th International Ship Control Systems Symposium*, Orlando, Florida, USA, April 7-9, 2003.

Tucker, A. J., “Opportunities and Challenges in Ship Systems and Control at ONR,” *IEEE Conference on Decision and Control*, December 4, 2001.

US Navy ONR/NSF EPNES, “ONR Control Challenge Problem (white paper),” http://www.usna.edu/EPNES/Challenge Problem.htm, 2002.

Weiss, G., *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.

Zivi, E.L. and T.J. McCoy, “Control of a Shipboard Integrated Power System,” *Proceedings of the 33rd Annual Conference on Information Sciences and Systems*, Baltimore, Maryland, USA, 1999.

ACKNOWLEDGMENTS

This research is sponsored in part by a 2003 grant from the US DEPSCoR and ONR (DOD/ONR N000 14-031-0660).

APPENDIX

The transfer functions and their respective parameters of the mathematical model of the shipboard power system are given in the following. The various transfer-functions represent input-output models for power generation, power demand, and voltage. The blocks' transfer functions and parameters have been determined through simulation and identification of the NCS systems as presented in Pekarek et al. (2003) and Sudhoff et al. (2003). The only modifications made to the system concern the loads: Only the constant-power demand type is used here as it adequately represents all load types for the power flow solution. Also, two loads instead of only one are modeled within each of the three zones.

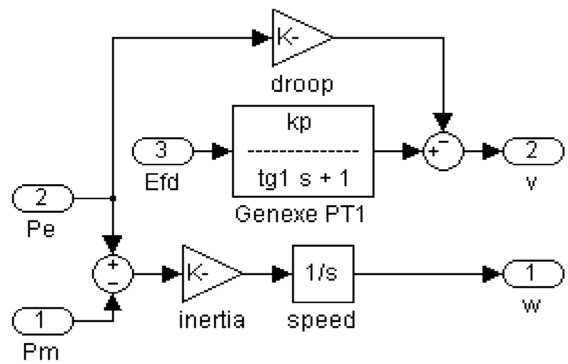


Figure A1. Synchronous generator model

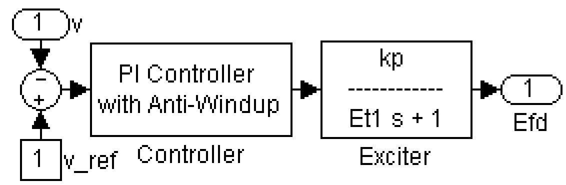


Figure A2. Exciter model and its controls

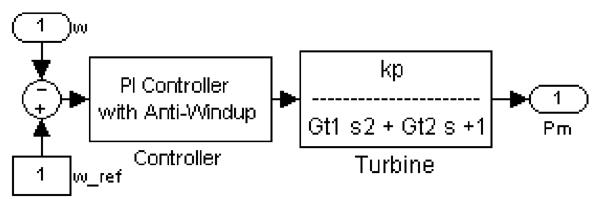


Figure A3. Turbine model and its controls

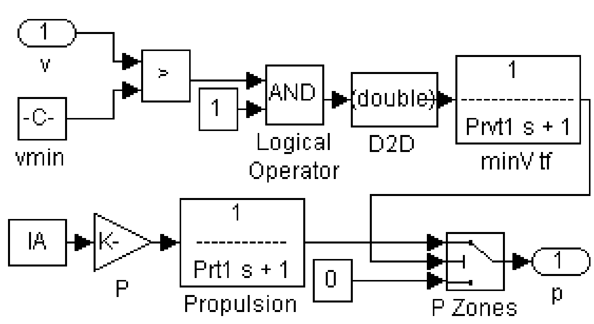


Figure A4. Shipboard propulsion system model

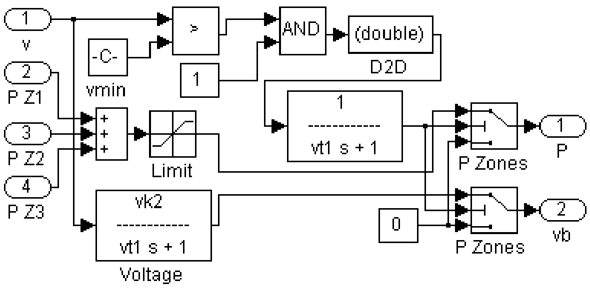


Figure A5. AC/DC converter model and controls

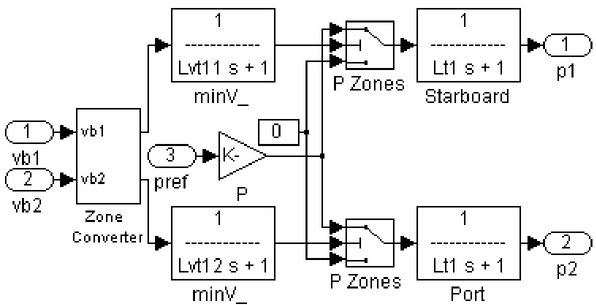


Figure A6. Zone model and its controls

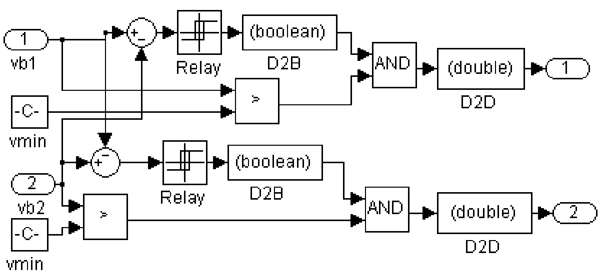


Figure A7. Zone converter model (part of Zone model, Fig. A6)

Table A1. Parameters

Generator		
Parameter	Value	Unit
V_{droop}	0.1	p.u.
$H_{inertia}$	1.0	p.u.
tg_1	0.2	sec.
kp	2.0	p.u.

Exciter		
Parameter	Value	Unit
Et_1	0.1	sec
kp	1.0	p.u.
PI: ki	2.0	p.u.
PI: kp	4.0	p.u.
PI: max. limit	1.02	p.u.
PI: min. limit	0.0	p.u.

Table 1. Parameters (continued)

Turbine		
Parameter	Value	Unit
Gt_1	0.0625	sec
Gt_1	0.4500	sec
kp	0.98	p.u.
PI: ki	1.0	p.u.
PI: kp	2.0	p.u.
PI: max. limit	1.02	p.u.
PI: min. limit	0.0	p.u.

PS (AC/DC) Converter		
Parameter	Value	Unit
vt_1	0.2	sec
$vt_{1\ Voltage}$	0.2	sec
vk_2	0.95	p.u.
vmin	0.8	p.u.

Propulsion Load		
Parameter	Value	Unit
$Prvt_1$	0.01	sec
Prt_1	0.10	sec
P	0.55	p.u.
vmin	0.8	p.u.

Zone Converter and Load		
Parameter	Value	Unit
Lvt_{11}	0.01	sec
Lvt_{12}	0.01	sec
$Lt_{1\ Standboard}$	0.10	sec
$Lt_{1\ Port}$	0.10	sec
vmin	0.8	p.u.

Shilpa B. Ganesh *completed her B.S. degree in Electrical Engineering from Bangalore University, India, in 2000 and has been a software consultant for L&T Komatsu for 3 years. Currently, she is a graduate research assistant at the Advanced Power & Electricity Research Center (APERC) at West Virginia University and pursuing her Master's degree in Software Engineering. She is interested in applying agent technology to control issues of electric power systems.*

Karl Schoder *(Ph.D.) received his Diplom-Ingenieur in Electrical Engineering from the University of Technology in Vienna in 1997 and his Ph.D. from West Virginia University in 2002. He has been working at APERC on modeling and simulating electric power systems and FACTS devices. His research interests include modeling, analysis and control of electric transmission and distribution systems.*

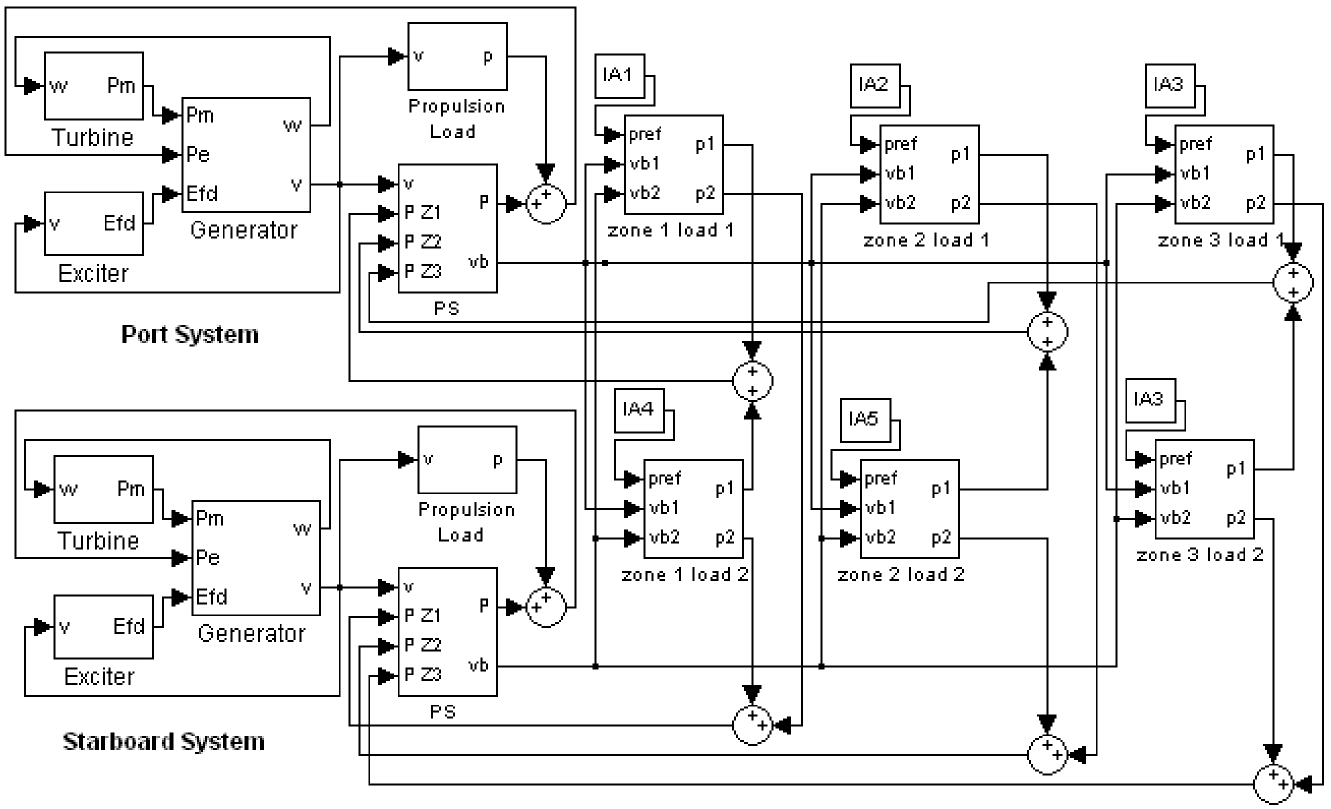


Figure A8. Top-level view of the shipboard model

Hong-Jian Lai *(Ph.D.) received his PhD in Mathematics from Wayne State University in 1988. He is a full Professor of the Department of Mathematics at West Virginia University. He has been working on discrete mathematics, algorithm and optimization for 15 years.*

Amer Al-Hinai *received the BSEE from Sultan Qaboos University, Muscat, Oman in 1997 and his M.S. degree in Electrical Engineering from West Virginia University in December 2000. He is currently a Ph.D. student and graduate research assistant at APERC. His areas of interest are power system dynamics and control, power distribution systems, and dynamic modeling.*

Ali Feliachi *(Ph.D.) received his Diplome d'Ingenieur en Electrotechnique from Ecole Nationale Poltytechnique of Algiers in 1976, and MS (1979) and Ph.D. (1983) in EE from Ga Tech. He is a Full Professor and Director of the Advanced Power & Electricity Research Center at West Virginia University. He has been working in the field of large scale and power systems for 25 years.*