1-1-2008

Multi-Agent Systems For Reconfiguration Of Shipboard Integrated Power System Including Ac-Dc Zonal Distribution System

Qiuli Yu

MULTI-AGENT SYSTEMS FOR RECONFIGURATION OF SHIPBOARD

INTEGRATED POWER SYSTEM INCLUDING AC-DC

ZONAL DISTRIBUTION SYSTEM

By

Qiuli Yu

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Electrical Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

December 2008

MULTI-AGENT SYSTEMS FOR RECONFIGURATION OF SHIPBOARD

INTEGRATED POWER SYSTEM INCLUDING AC-DC

ZONAL DISTRIBUTION SYSTEM

By

Qiuli Yu

Approved:

Noel N. Schulz Professor of Electrical and Computer Engineering (Major Advisor and Director of Dissertation)	Burak Eksioglu Assistant Professor of Industrial and Systems Engineering (Minor Advisor and Committee Member)
Herbert L. Ginn Assistant Professor of Electrical and Computer Engineering (Committee Member)	Anurag K. Srivastava Assistant Research Professor of Electrical and Computer Engineering (Committee Member)
James E. Fowler Professor and Interim Director of Graduate Studies, Electrical and Computer Engineering	Sarah A. Rajala Dean of the Bagley College of Engineering

Name: Qiuli Yu

Date of Degree: December 12, 2008

Institution: Mississippi State University

Major Field: Electrical Engineering

Major Professor: Dr. Noel N. Schulz

Title of Study: MULTI-AGENT SYSTEMS FOR RECONFIGURATION OF SHIPBOARD INTEGRATED POWER SYSTEM INCLUDING AC-DC ZONAL DISTRIBUTION SYSTEM.

Pages in Study: 124

Candidate for Degree of Doctor of Philosophy

Future all-electric warships with an integrated power system (IPS) are capable of unlocking large amounts of power dedicated to propulsion and redirecting this power for service loads, weapon loads, and other loads. The IPS for all-electric ships combines the power generation system, electric propulsion system, power distribution system, and power control and management system all together. The move to IPS design will significantly improve efficiency, effectiveness, and survivability.

To meet the needs of the US Navy, enhancing survivability by reducing susceptibility to damage, a IPS prefers decentralized reconfiguration system is preferred for IPS instead of traditional reconfiguration techniques used for terrestrial power grids. A multi-agent system (MAS) is a loosely coupled network composed of several agents. These agents interact with their environments and communicate with each other to solve problems that are beyond the individual capabilities or knowledge of each single agent.

Because of its decentralized feature and lack of a global control feature, MAS appears to be the best candidate for IPS reconfiguration.

This research work proposes a new model of an IPS, based on the Naval Combat Survivability, DC Distribution Test-bed (NCS DCDT). The new model combines the electric power generation system, electric propulsion system, and AC-DC zonal distribution system. To decrease the probability of distribution zones losing power, the new model modifies original design of the zonal distribution system in NCS DCDT.

Another main endeavor of this research work is to design a MAS for reconfiguration of an IPS with AC-DC zonal distribution system. The MAS consists of three sub-MAS, named power generation MAS, propulsion MAS, and distribution MAS, and includes forty-one different agents which are instances of nineteen different abstract agent classes. The MAS is implemented with JAVA/JADE software and simulated on a platform of JADE 3.4.1 and JAVA jdk 1.5.0_08. Simulation results show that the MAS can execute reconfiguration functions such as fault area isolation, automatic switching, and load shedding.

DEDICATION

I would like to dedicate this dissertation to my parents and beloved family.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my academic advisor, Dr. Noel Schulz, for her constant guidance and support both technically and emotionally during my studies in Mississippi State University. She has given me not only the support in my academic program, but also the help in our family life.

I would like to acknowledge support and advice of my committee members, Dr. Herbert L. Ginn III, Dr. Anurag K. Srivastava and Dr. Burak Eksioglu during my research and course work.

I would like to thank my graduate colleagues in the Electric-ship research group of the Electrical and Computer Engineering Department at Mississippi State University for their help, support, and friendship during my studies and research.

I sincerely appreciate Mississippi State University, the Electric Ship Research and Development Consortium (ESRDC), and the Office of Naval Research (ONR) for their financial support during my dissertation work.

Special thanks are given to my parents, my wife, and my son and daughter, for their encouragement, support, and understanding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

LIST OF ABBREVIATIONS

AC	Alternating Current
AG	Auxiliary Generator
AMS	Agent Management System
BC	Buck Converter
CM	Converter Module
CORBA	Common Object Request Broker Architecture
CPL	Constant Power Load
DC	Direct Current
DCDT	DC Distribution Testbed
DF	Directory Facilitator
DG	Distributed Generation
EMI	Electromagnetic Interfere
GA	Genetic Algorithm
GIS	Geographic Information System
GPT	Generation and Propulsion Testbed
ICE	Internal Combustion Engine
IGBT	Insulated-Gate Bipolar Transistor
IPS	Integrated Power System
LB	Load Bank
MAS	Multi-Agent System
MD	Motor Drive
NCS	Naval Combat Survivability
PEBB	Power Electronics Building Block
PS	Power Supply
PSO	Particle Swarm Optimization

PWM	Pulse-Width Modulation
RMS	Root Mean Square
RTDS	Real-Time Digital Simulator
SA	Simulated Annealing
SBA	Substation Breaker Agent
SCADA	Supervised Control and Data Acquisition
SPS	Shipboard Power System
SSCM	Ship Service Converter Module
SSIM	Ship Service Inverter Module
TBA	Tie Breaker Agent
TS	Tabu Search
VTB	Virtual Test Bed

CHAPTER I

INTRODUCTION

This chapter introduces ship propulsion history, ship mechanical and electrical propulsion system, integrated power system (IPS), agent and multi-agent system (MAS), and reconfiguration of an IPS. It presents the objectives and contributions of this dissertation.

1.1 Ship Propulsion History and Ships with Mechanical Propulsion Systems

Early small and medium ships relied on paddles and/or sails to produce propulsion thrust. Paddles and sails are actually two kinds of power converters by which the manpower and wind power are changed to mechanical power for ship propulsion. As ships grew dramatically both in size and displacement, the power produced through sails and paddles no longer provided enough energy for more powerful propulsion systems.

With the invention and development of external combustion engines, such as steam engines and later, internal combustion engines, traditional commercial and military ships employed a mechanical propulsion system that uses gas turbines as the prime mover. A steam engine is an external combustion heat engine that makes use of the thermal energy that exists in steam, converting it to mechanical work. An advantage of the steam engine is its ability to convert heat from almost any source into mechanical work without changes in steam engine configuration [1]. The internal combustion engine

(ICE) is a heat engine that includes a combustion chamber. A fuel mixes and burns with an oxidizer in the combustion chamber to create gases of high temperature and pressure. The expansion of the gases can perform useful work by moving pistons. Contrasting with external combustion engines such as steam engines, the ICE performs useful work when expanding hot gases directly cause movement. Compared with external combustion engines, the advantages of the ICE are its high power-to-weight ratio and excellent fuel energy density. Jet aircraft, helicopters, and large ships, which require very high power, usually utilize ICEs as gas turbines [2].

A typical ship mechanical propulsion system includes a gas turbine, a shaft with a gear box, and a propeller. From the control engineer's point of view, the mechanical system also includes the ship dynamics, a shaft speed controller-governor, a propellor pitch controller, and a main controller. The main controller will coordinate behaviors of the governor and the propellor pitch controller, as well as control ship speed and optimize efficiency. Figure 1.1 [3] illustrates this mechanical propulsion ship system.

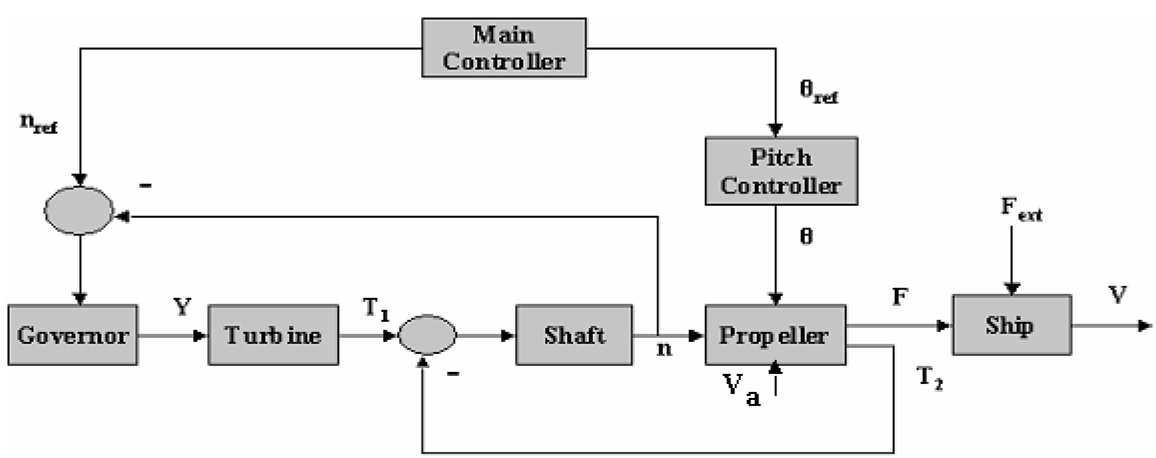


Figure 1.1 Diagram of Ship Mechanical Propulsion System [3]

Here n_{ref} and θ_{ref} are the shaft speed reference and propeller pitch reference, respectively. n and θ are the shaft speed and propeller pitch, respectively. Y is the fuel index and V_a is the water speed. T_1 and T_2 are turbine torque and propeller torque, respectively. F, F_{ext}, and V are propeller thrust, external force, and ship speed, respectively.

1.2 Ships with Electric Propulsion Systems

During the cycle of ship design and construction, the first and the most important step for ship designers is to design a propulsion system according to customer's demand for the ship. The mechanical propulsion systems are so large and heavy that ship designers have to design and construct the rest of the ship around it rather than create a tailored mechanical propulsion system for the ship [4]. Sizes and locations of machines in mechanical propulsion systems reduce the space available for cargo and passengers and limit efficient loading and unloading. The considerable length of the shaft in mechanical propulsion systems makes efficient space use challenging. All of these factors limit ship design flexibility to incorporate designs that would meet other needs, such as optimally allocating cargo space, cabin space, and the shape of propulsion units around the needs of the marketplace [4]. On the other hand, in electric propulsion systems, the turbine speed becomes fully decoupled from the propeller speed, making it possible to optimize the turbine speed with regard to fuel efficiency. No direct link, such as shaft and gearbox, occurs between the prime mover and the propeller in an electric propulsion system, which provides ship designers with more design flexibility [5]. Higher power-to-weight ratio of electric machines in an electric propulsion system reduces ship internal space and can accommodate more cargo and passengers for commercial ships and more weapons and personnel for warships. Figure 1.2 shows a structural comparison between electric and mechanical propulsion [6]. Compared with mechanical propulsion systems, electric propulsion systems improve fuel efficiency, provide flexible design, and reduce internal volume. Electric propulsion systems are widely used in large commercial ships and especially warships that demand a high level of maneuverability or travel often at a slow speed. In these circumstances the variable-speed characteristics of the electric motor and the optimum-efficiency constant speed of diesel engines driving alternators can be used to their best advantage.

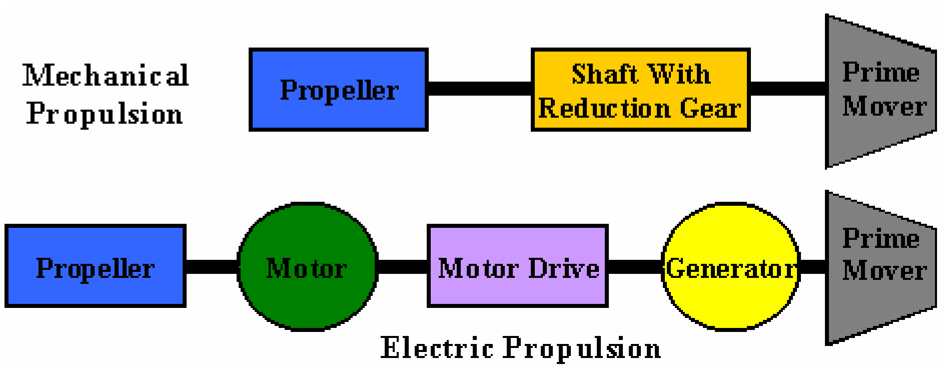


Figure1.2 Electrical vs. Mechanical Propulsion [6]

1.3 All-Electric Ships and IPS—Characteristics and Architecture

Current large warships such as aircraft carriers demand more and more propulsion power in a way that modern techniques for power generation and ship configuration almost reach a bottleneck. At the same time, growth continues in the need for electric power for hotel loads, pulsed weapons, and high power sensors. In current warships and combat vehicles the electric propulsion system separates from ship electric power systems for service loads and weapon loads. The large amounts of power locked in the mechanical propulsion system are not available for other uses and are wasted when ships are in a battle situation and do not need to escape with high speed. Future all-electric warships with an integrated power system (IPS) are capable of unlocking the large amounts of power dedicated to propulsion and redirecting this power for service loads, weapon loads, and other loads. The IPS for all-electric ships combines power generation system, electric propulsion system, power distribution system, auxiliary power system for pulsed weapons, and power control and management system all together. The move to IPS design will "significantly improve efficiency, effectiveness, and survivability while simultaneously increasing design flexibility, reducing costs, and enhancing Sailors' and Marines' quality of service" [7]

According to John Prousalidis and Dharshana Muthumuni [8], the characteristics of an IPS can be summarized as following:

- Hybrid power system—the IPS comprises both AC and DC subsystems of various operation voltages and frequencies.
- Large power system—the IPS for a modern container ship can be in the range of 80-100MW.
- Non-linear loads with non-conventional behavior—many power electronics elements and circuits, induction motors, and pulsed weapons account for this.
- Sensitivity to power quality and EMI problems—extensive use of power electronics, distribution systems with DG, and DC distribution system provokes power quality problems such as power instability, harmonic distortion, voltage dips, and switching electromagnetic transients.

Different architectures of IPS for future all-electric ships are in consideration. These architectures can be classified as two categories: IPS with AC distribution system and IPS with AC-DC distribution system. IPS with AC distribution system uses either radial distribution system or ring distribution system. However, IPS with AC-DC distribution system uses only DC zonal distribution system.

For the IPS designers of future all-electric ships, the first and the most important step is to determine a proper IPS architecture after comparing advantages and disadvantages of different IPS candidates and determining the appropriate trade-offs. The next step is to develop an entire IPS system mathematical model so that the designers can simulate the IPS system in both off-line and real-time modes, analyze power stability of the IPS, and perform IPS protection and reconfiguration simulations.

1.4 Reconfiguration of IPS

Current warships participate in a variety of situations, such as full speed fleeting, constant speed cruising, maneuvering, and loitering. The IPS must switch working modes according to different situations in which the vital loads indicated by their load priorities will change. For instance, during battle circumstances, the vital load in a warship may be the pulsed weapon load, so it must have the highest priority. When the warship begins to escape with full speed, the vital load will change to propulsion load. Thus, during the warship mission transition, the IPS needs to reconfigure the whole power system by changing the power system topology through the opening or closing of switches, sectionalizers and/or circuit breakers. Apart from the mission transition, faults are another reason for IPS reconfiguration. For the IPS of all-electric warships, the faults may happen due to material casualty of individual equipment (generator, load, cable, and power electronics building block (PEBB)) or widespread damage due to battle with hostile warships [8]. To meet the needs of the US Navy, enhancing survivability by reducing susceptibility to damage, the IPS should use a new reconfiguration strategy rather than traditional reconfiguration techniques used for terrestrial power grids. The power protection and reconfiguration system includes four basic strategies, fault detection, fault isolation, reconfiguration/restoration, and fault removal. The new reconfiguration strategy should be decentralized so that the failure of one or two components in the protection and reconfiguration system will not cause the failure of the whole system. The new reconfiguration strategy should be capable of reconfiguring the IPS automatically, rather than manually, to increase electric ship survivability. Also, the new reconfiguration strategy should be able to handle faults occurring in both AC and DC generation/distribution systems. The new reconfiguration technology covers all parts of the shipboard power system (SPS), running from fueling to power generation, to power distribution, to power storage, to electrical motors delivering mechanical energy, and to propelling devices, and is a new challenge for warship designers.

1.5 Multi-agent system (MAS) and agent-based reconfiguration

The term *agent* has been used by many people to mean many different things. In the field of artificial intelligence, an intelligent agent can be defined as an entity that is capable of perceiving changes of its environment and taking different actions according to different environmental changes. Here the term “intelligent” means ability of the entity to interact with its environment in human being’s manner, and take measures automatically [9]. A multi-agent system is a loosely coupled network composed of several agents, or problem solvers. These agents interact with their environments and communicate with each other to solve whole problems that are beyond the individual capabilities or knowledge of each problem solver [10]. Each agent can pursue its own goal (local optimum), as well as share a common goal (global optimum) with others. According to Katia Cycara [10], the characteristics of a MAS include (1) each agent has a limited viewpoint into the whole system; (2) control of the whole system is distributed and no system global control exists; (3) data are decentralized; and (4) each agent is running asynchronously.

Because of its decentralized feature and automatic control, MAS appears to be the best candidate for IPS reconfiguration to increase warship survivability. In the present time, some researchers are working on agent-based reconfiguration for shipboard power systems [11-18]. These SPS are all pure AC power system; the topology of these SPSs includes radial topology, ring topology, and mesh topology. The US Navy prefers IPS with AC-DC zonal distribution system, as discussed in Chapter IV. Agent-based reconfiguration for IPS with AC-DC distribution system is a new challenge for all-electric ship researchers and is also a valuable topic.

1.6 Problem Statement and Outline

Based on the discussion above on ship mechanical propulsion system, ship electric propulsion system, IPS architecture and characteristics, and MAS and agent-based reconfiguration for IPS, this dissertation will focus on:

- Analysis and design of an IPS, which includes electric power generation subsystem, electric propulsion subsystem, AC-DC zonal distribution subsystem, and power control and management.
- MAS-based reconfiguration of an IPS with AC-DC zonal distribution system. The MAS consists of three subsystems. The first MAS subsystem is for the electric propulsion subsystem, and includes the motor/load agent, inverter agent, rectifier agent, transformer agent, and bus agent. The second MAS subsystem is for the electric power generation subsystem, and includes the main generator agent, auxiliary generator agent, and common bus agent. The third MAS subsystem is for the AC-DC zonal distribution system, and includes the distribution transformer agent, power supply agent, Port Bus agent, Starboard Bus agent, SSCM (Ship Service Converter Module) agents, and other agents for three zonal branches. The MAS system is implemented and simulated using the JAVA/JADE platform.
- Test case design and demonstration of MAS-based reconfiguration for IPS with AC-DC zonal distribution system. Different test cases are designed to test load/component cut-off, zonal branches cut-off, propulsion subsystem auto-switch, SSCM auto-switch, and load shedding. Software will be developed with JAVA to illustrate the original architecture of the IPS before reconfiguration and final architecture of the IPS after reconfiguration.

The outline of this dissertation is as follows:

Chapter I introduced background on a ship propulsion system, shipboard power system and integrated power system, and multi-agent system, and MAS-based reconfiguration for shipboard power system. Chapter II reviews the literature of research on ship propulsion systems including both mechanical and electric propulsion systems; SPS and IPS; MAS applications, and agent-based reconfiguration for SPS. Chapter III summarizes preliminary works on SPS/IPS modeling and MAS-based reconfiguration of SPS/IPS, especially research work that has been done by electric ship research effort of the Electrical and Computer Engineering Department at Mississippi State University. Chapter IV presents analysis and design of an IPS with AC-DC zonal distribution system, encompassing part contribution of this dissertation. Chapters V, VI and VII discuss design and implementation of three sub-MAS, propulsion MAS, distribution MAS, and generation MAS, respectively. These three chapters encompass the main part and contribution of the dissertation. Chapter VIII designs a test case to simulate the whole MAS for reconfiguration of an IPS with AC-DC zonal distribution system. Chapter IX gives conclusions of the dissertation and future research directions related to the dissertation topic.

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

This chapter provides a literature survey in the research area of reconfiguration of IPS for all-electric ships. The survey includes three parts, architecture, modeling, and simulation of IPS; MAS applications in power systems; and agent-based reconfiguration for IPS. Following the literature survey is a summary of the dissertation work/contributions, which extends research activities currently carried out in these previous works.

2.1 Architecture, Modeling and Simulation of IPS

The IPS for an all-electric ship combines the power generation system, electric propulsion system, power distribution system with distributed generators (DG), auxiliary power system, and power control and management system all together. Future all-electric warships with IPS not only are able to redirect the large amounts of power dedicated to propulsion and release this power for service and pulsed weapon loads, but also significantly improve efficiency, effectiveness, and survivability. According to different distribution system topologies, the IPS can be classified into three categories, i.e., IPS with AC radial distribution system, IPS with AC ring distribution system, and IPS with AC-DC zonal distribution system. R. Jayabalan and B. Fahimi [19] define the IPS with radial AC distribution system as an architecture in which multiple independent distribution lines run from the source to various service loads. K.L.Butler and N.D.R. Sarma [20] illustrated this radial topology in their paper. This topology can optimize cost, but reduces the survivability of the shipboard power system because a relatively large target exposed to battle damage and multiple service loads are susceptible to at least two single point failures—loss of power source and failure along distribution path. An alternative to IPS with radial distribution system is IPS with AC ring distribution system in which a single source matches multiple services with a single common distribution bus. This topology increases system vulnerability due to single point failure. An IPS with both radial and ring distribution systems needs redundant power sources and buses to improve system survivability. To increase survivability with less than 100% redundancy, an IPS with AC-DC zonal distribution system has been recommended due to its increased accessibility to sources and loads and its more customized feature to support DC and 60/400 Hz AC, and has undergone additional research by many researchers including R.Jayabalan and B. Fahimi[19], S.D. Sudhoff et al. [21], A. Quroua, L. Domaschk, and J.H. Beno[22], and S.D. Pekarek et al.[23].

Modeling and simulation are important tools to examine system performance prior to experimentation and prototyping. Currently many researchers are working on modeling and simulation of IPS. A.Ouroua, L.Domaschk, and J.H.Beno[22] developed a model of the notional DDX power system architecture with MATLAB/SIMULINK and ACSL. This model includes an AC-DC mixed zonal distribution system, which is different from the Naval Combat Survivability (NCS) DC Distribution Testbed (DCDT). Also this model does not consider distributed generation. S.D.Pekarek et al. [23]

described the NCS Generation and Propulsion Testbed (GPT) and NCS DCDT and provided simulation results based on ACSL. However, they modeled each component with a nonlinear average value model. Terry Ericsen, Narain Hingorani, and Yuri Khersonsky [24] presented a VTB schematic model of advanced marine electrical power system. But they do not provide any information on the VTB simulation in their paper. J.Langston, S.Suryanarayanan, M.Steurer, M.Andrus, S.Woodruff, and P.F.Ribeiro [25] modeled and simulated a notional IPS using a Real-Time Digital Simulator (RTDS). Their paper focuses on experiences with the implementation of the IPS model other than simulation results. And their model of IPS is not the same as a combination of NCS GPT and NCS DCDT.

The Virtual Test Bed (VTB) package is a type of simulation software developed by the research team at University of South Carolina. The objective of VTB is to provide a modeling and simulation facility to all engineers who design complex, large-scale, multi-technical dynamic systems, addressing issues of system stability, power quality, reconfigurability, thermal management, control systems, and power system topology [26]. VTB provides users with the ability to import structural models that describe the physical properties of the system, and dynamic models from a variety of environments, to connect finite element models with lumped-element dynamic models, and to create and drive advanced visualizations from within the simulation environment [26]. The complexity of the design problem for an all-electric ship makes the use of the simulation driven design crucial to success. With its unique characteristics of multidiscipline, distributed simulation, and simulations with hardware in the loop, VTB is a competitive candidate for modeling and simulation of IPS. It is possible to translate the IPS model implemented using MATLAB/SIMULINK into a VTB model in the future.

2.2 MAS Application in Power Systems

An agent is "a software (or hardware) entity that is situated in an environment and is able to automatically react to changes in the environment" [27]. An intelligent agent is a kind of high-level agent with three key attributes—reactivity, pro-activeness, and social ability. These three characteristics set intelligent agents apart from many existing software and hardware systems [27]. A multi-agent system (MAS) is a co-operative system in which more than one intelligent agent work together to achieve a total goal. With agent technology, we are able to deal with systems that are dynamic and flexible in real-time. MAS can be applied for modeling and implementation of dynamic and scalable hardware/software systems. MAS applications in power systems include monitoring and diagnostics [28] [29], distributed control and restoration [30], market modeling and simulation [31] [32], and protection [33]-[37]. Applications currently being investigated in the field of distributed control and restoration are power system restoration, active distribution networks operation, micro-grid control, and control of shipboard electrical systems [27].

Currently some researchers in the field of power system control and management are working with architecture and design of MAS [38] [39]. Their main idea is to replace the centralized power system control and management with an alternative—decentralized MAS based control and management system. The same idea can also be applied for IPS, as stated in the following section.

2.3 Agent-Based Reconfiguration for IPS

To increase fight through and survivability of all-electric ships, the reconfiguration for IPS favors a decentralized system over a centralized system. The MAS is considered as an important candidate for reconfiguration of IPS by all-electric ship researchers and designers.

K.L.Butler-Purry [11] presented a MAS based self-healing system for SPS. The SPS consists of ring-structured switchboards with an AC radial distribution system. A self-healing reconfiguration system has been developed for the AC radial distribution system and consists of one preventive, one predictive, and two restorative methods. A multi-agent system that implements the self-healing reconfiguration system has hierarchical control architecture and consists of a supervised control and data acquisition (SCADA) agent, a geographic information system (GIS) agent, an on-line assessment agent, and multiple event detection and failure identification agents. The self-healing reconfiguration system is a centralized system because the system includes a GIS database and historical data, and operating limits databases, although the MAS has a decentralized structure.

Jignesh M. Solanki and Noel N. Schulz [12] [13] discussed MAS based reconfiguration for IPS with an AC radial distribution system. The MAS was first implemented with MATLAB as a platform. The IPS was modeled with VTB as a simple radial distribution system. MATLAB-VTB co-simulation had been executed to demonstrate fault detection, fault isolation, and power system restoration. Later JAVA/JADE was used to replace MATLAB as the platform for the MAS implementation, overcoming drawbacks of MATLAB, such as single thread programming, one-time data storage, and so on [13].

K.Huang, D.A. Cartes, and S.K. Srivastava [14][15][16] implemented MAS based algorithms for reconfiguration of SPS that is either mesh-structured AC distribution or ring-structured AC distribution. A recent research work by G. Morejon, S.K. Srivastava, and D.A. Cartes [17] shows MAS based reconfiguration for SPS. The SPS is modeled as radial AC distribution system with VTB and the MAS is implemented with JADE. The SPS communicates with MAS through Common Object Request Broker Architecture (CORBA).

From literature reviews of the three sections above, we can find that almost all MAS based reconfigurations of IPS are based on the IPS model, which is modeled as simple radial or ring distribution system and usually considers the generation system as an ideal power source (voltage source or current source). In fact, the IPS favors AC-DC zonal distribution system [19]. This kind of IPS model is incomplete or not accurate enough to investigate dynamic behavior of real generation system (prime mover and synchronous machine). Also the IPS model usually considers load as a constant and ignores the ship-speed dynamics. In the real world the propeller thrust is balanced by resistance from the hull, and external forces such as wind and waves [3]. Thus the load will change with time and the generation system should be a closed loop system with load feedback. On the other hand, all MAS based reconfigurations deal with faults occurring at AC generation and distribution system. In fact AC-DC zonal distribution system is more preferable than AC radial or ring distribution system for shipboard power systems, because it can increase reliability and survivability of warships with less than 100% redundancy. The MAS based reconfiguration for IPS should be able to deal with faults that happen on the AC generation system, AC distribution system, or AC-DC zonal distribution system.

2.4 Contributions of Proposed Work

Based on the literature review in above sections of this chapter, this dissertation will contribute following issues:

- Analysis and design of system level representation of an IPS — The IPS model includes electric power generation system, electric propulsion system, power distribution system, and power control and management system. The power generation system and electric propulsion system are based on Naval Combat Survivability Generation and Propulsion Testbed (NCS GPT) model. The power distribution system is based on Naval Combat Survivability DC Distribution Testbed (NCS DCDT). The power generation system will model turbine as prime mover, synchronous machine as generator, and some PEBBs as needed.

- MAS design and implementation—When a fault occurs, the protection system of the IPS will detect the fault and isolate the area including the fault without considering power balance. Each intelligent agent can get fault information from the protection system and exchange information with its neighbors. Based on the local circuit parameters and information from neighbor agents, the intelligent agents will reconfigure the topology of the IPS by opening/closing circuit breakers associated with them to meet power balance. The MAS is implemented with JADE and JAVA.

- Test case design and simulation of MAS—Different test cases are designed and the MAS is simulated to test the MAS capability of load/component cut-off, zonal branch cut-off, propulsion subsystem auto-switch, SSCM auto-switch, and load shedding. Software will be developed with JAVA/JADE to illustrate the original architecture of the IPS before reconfiguration and final architecture of the IPS after reconfiguration.

2.5 Summary

This chapter introduces architecture, modeling, and simulation of IPS, discusses MAS applications in power systems and agent-based reconfiguration for IPS, and describes contributions of this research work. The next chapter will summarize previous work on agent-based reconfiguration techniques at Mississippi State.

CHAPTER III

PREVIOUS WORK AT MISSISSIPPI STATE UNIVERSITY

This research work is based on and extends the previous work Jignesh Solanki accomplished at Mississippi State University [18]. It is necessary to introduce the preliminary work here, which will help provide a clear understanding of the differences between the preliminary work and proposed research work and the contributions of this dissertation.

3.1 Modeling of Power Systems and Ship Power Systems

In the first phase of the preliminary work, a test system is modeled with the combination of SIMULINK and VTB. This is a power system with two sources and three loads including a PQ load, as illustrated in Figure 3.1. The MAS is implemented using SIMULINK and the power system is modeled using the VTB platform.

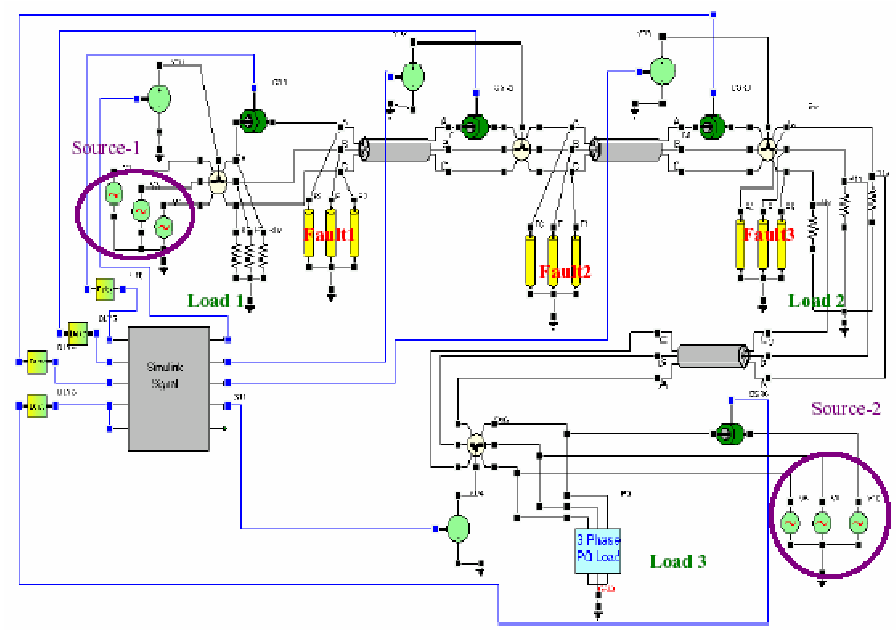


Figure 3.1 Test System I in Phase I [18]

To overcome the drawbacks of MATLAB/SIMULINK software, such as single-threaded programming and one-time data storage, the MAS is built with JAVA/JADE in the second phase of the preliminary work. Also several power systems with DG (or distribution networks) are modeled using VTB, with more focus on MAS-based reconfiguration of distribution systems with DG. One-line diagrams of these distribution networks are illustrated in Figure 3.2-3.6 [18].

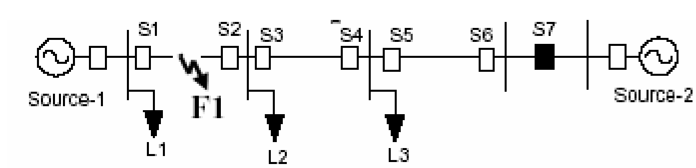


Figure 3.2 Distribution Network I in Phase II [18]

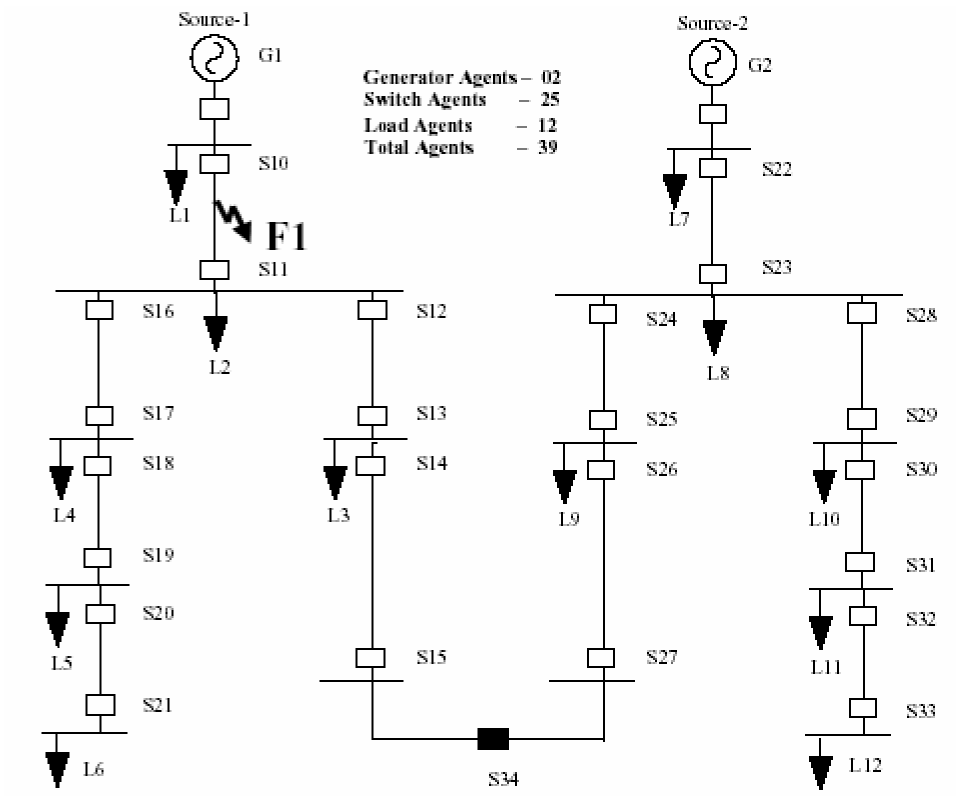


Figure 3.3 Distribution Network II in Phase II [18]

The distribution network I and II are two simple radial power systems without DG.

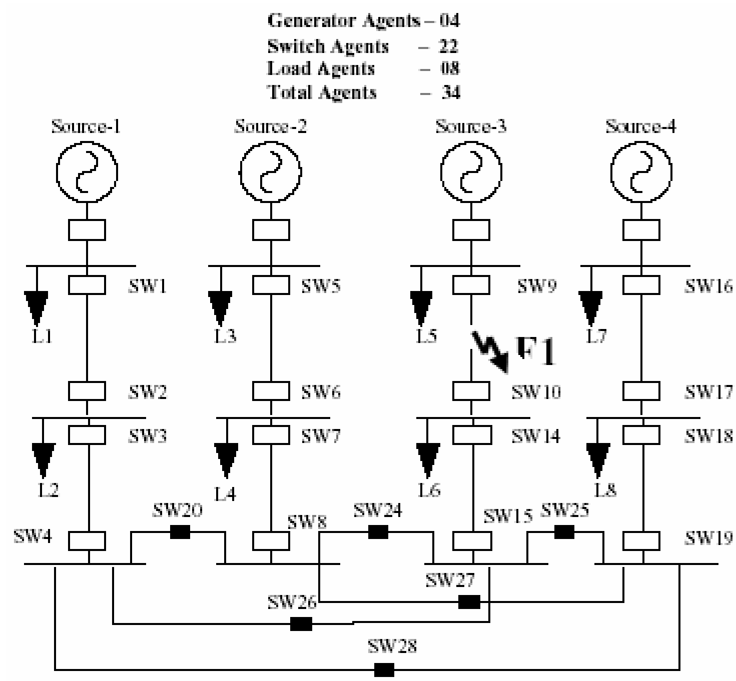


Figure 3.4 Distribution Network III in Phase II [18]

The distribution network III is also a radial power system with four generators on one side and one common bus in the other side.

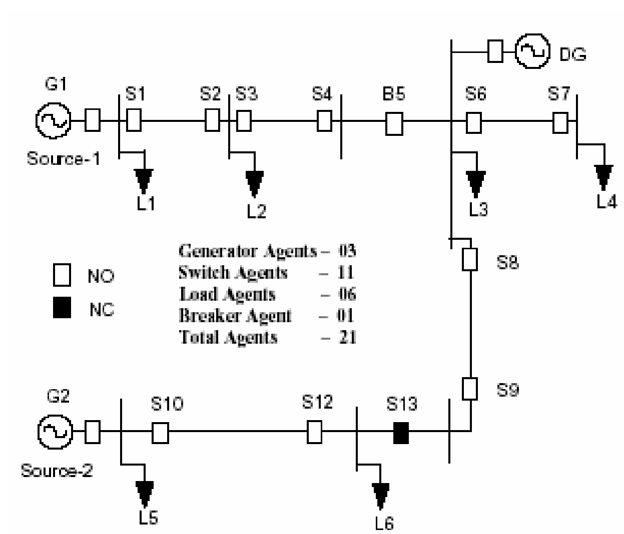


Figure 3.5 Distribution Network IV in Phase II [18]

The distribution network in Figure 3.5 is a radial power system with DG.

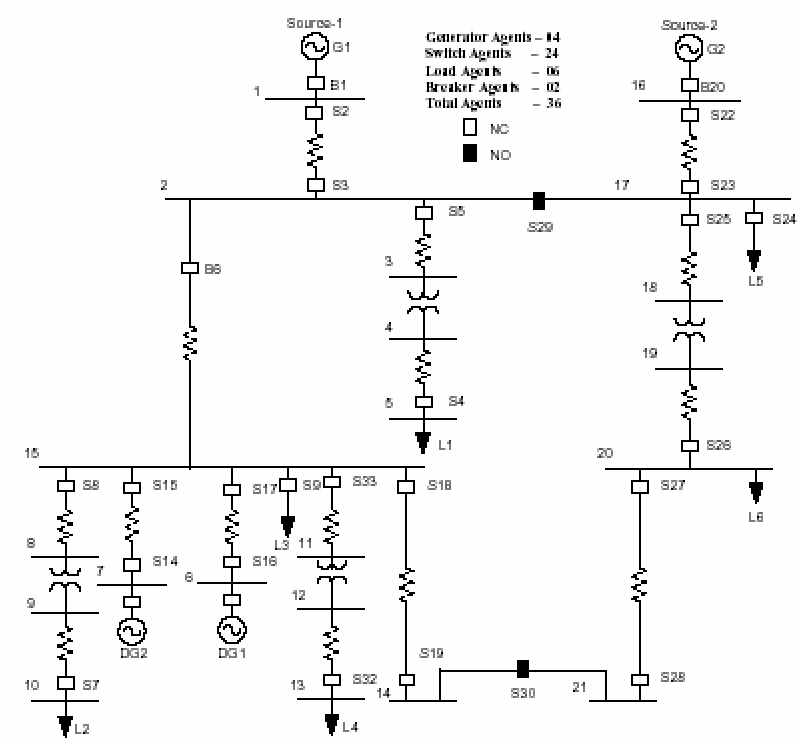


Figure 3.6 Distribution Network V in Phase II [18]

The distribution network in Figure 3.6 is a modified ship power system for an icebreaker. It is a radial power system with DGs.

As a summary, the distribution networks developed in phase II of the preliminary work are all pure AC power systems with radial architecture, focusing on the impacts of DG, load shedding, and islanding on the reconfiguration and restoration.

3.2 Multi-Agent Systems

In the first phase of the preliminary work, design of a MAS focuses on how to detect faults in a radial power system and reconfigure the power system. A MAS consists of three types of agents: switch agents (SAs), substation breaker agents (SBAs), and tie breaker agents (TBA). Each of SBAs and TBAs can sense the magnitudes of both voltage and current downstream, while each of SAs can only measure the magnitude of downstream current. When a fault happens, a switch agent will communicate with all its downstream switch agents if it senses the fault current. If the next downstream switch agents also detect the fault current, it can be derived that the fault is not located between these switches. Otherwise the fault occurs between these switches. Based on the result of fault detection, the SAs decide whether or not to lock the switches. The MAS is implemented with MATLAB and embedded into a radial power system, which is built with VTB.

In the second phase, additional different types of agents, such as generator agents (GA) and load agents (LA), are designed and embedded into the MAS to reconfigure and restore distribution networks with DG, load shedding, and islanding. The MAS is developed in JADE and can perform peer-to-peer communication between agents in an efficient way, besides overcoming the drawbacks of MAS in MATLAB.

As a summary, the MAS in the preliminary work is really a simple integrated system that combines a power protection system and a power reconfiguration system together. The MAS is designed specifically for simple radial AC power systems and works in circumstances where DG exists, load shedding, and islanding.

3.3 Summary

As outlined above and detailed in [18], Jignesh Solanki's work was the first MAS research within the MSU power research laboratory. This work developed the fundamentals of MAS and demonstrated them on fairly simplistic AC test systems as a proof of concept.

The work described in this dissertation extends that work with the following enhancements:

- The IPS is not a simple radial power system, but a combination of electric generation system, electric propulsion system, and AC-DC zonal distribution system.
- The IPS is not a pure AC power system, but an AC-DC mixed power system.
- The MAS implements faulted zone area isolation, propulsion subsystem auto-switching, SSCM auto-switching, and load shedding.

CHAPTER IV

ANALYSIS AND DESIGN OF AN IPS

As discussed in Chapter II, different architectures of IPS for future all-electric ships are in consideration. These are AC radial distribution system, AC ring distribution system, and AC-DC zonal distribution system. The first section of this chapter will briefly introduce these three kinds of IPS and discuss their advantages and disadvantages. The second section will focus on analysis and design at the system level of an IPS with an AC-DC zonal distribution system.

4.1 IPS with Different Distribution Architectures

4.1.1 An IPS with an AC Radial Distribution System

In an IPS with an AC radial distribution system, the power generation system utilizes a ring architecture in which generators are connected through generator switchboards in a ring configuration. In the AC radial distribution system, service loads are connected to the sources through multiple independent distribution lines [19]. Figure 4.1 shows a typical IPS with an AC radial distribution system [20]. The radial distribution architecture can optimize costs. However, because of multiple service loads being susceptible to at least two single point failures – loss of power source and failure along distribution path [19] , the radial AC distribution architecture is difficult to manage after battle damage and reduces the survivability of the shipboard power system.

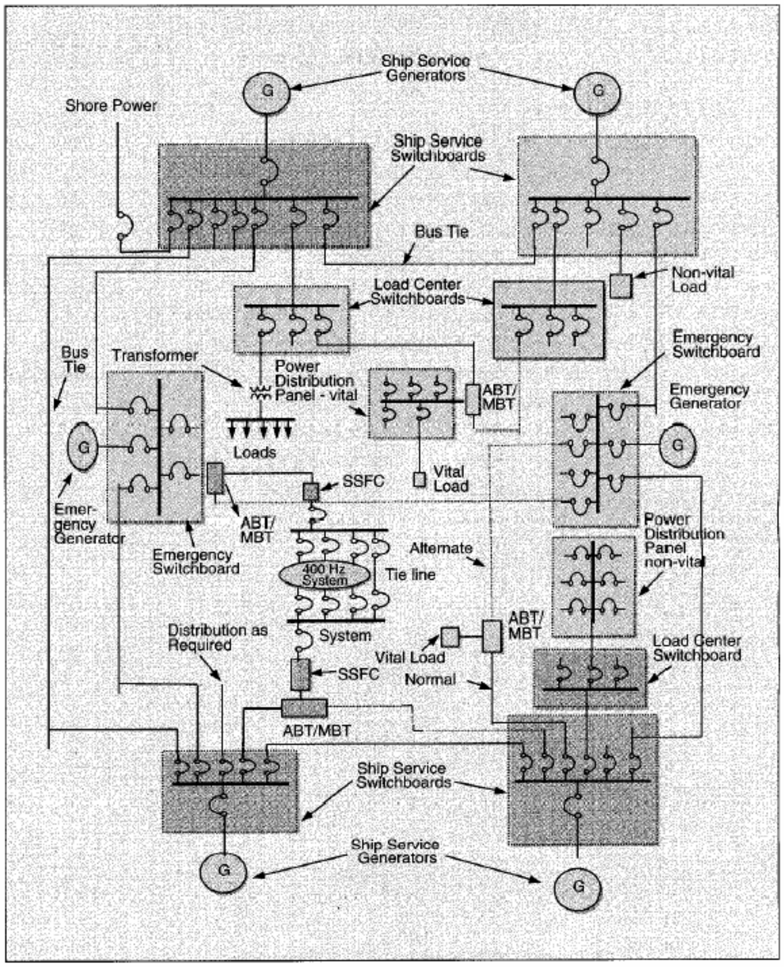


Figure 4.1 An IPS with an AC Radial Distribution System [20]

4.1.2 An IPS with an AC Ring Distribution System

Another IPS architecture with AC distribution system is a ring distribution system in which a single source connects multiple services with a single common distribution. Although the ring distribution configuration can relieve two single point failures in radial distribution systems, the common distribution bus increases system vulnerability due to single point failure [19]. Improving system survivability for IPS with AC distribution systems requires the installation of another backup system with high redundancy of generators, switchboards, power panels, and load centers. The need for a backup system creates ship design difficulty and decreases cargo volume.

4.1.3 An IPS with an AC-DC Zonal Distribution System

To increase survivability with less than 100% redundancy and meet the Navy's demand, the IPS with AC-DC zonal distribution system is preferred and designed in the NCS DCDT (Naval Combat Survivability, DC Distribution Test-bed) program. In addition, the IPS with AC-DC zonal distribution system is more customized to support DC and 60/400 Hz AC. The Navy has sponsored the development of the NCS DC DCDT [19, 23]. Together with the Naval Combat Survivability (NCS) Generation and Propulsion Testbed (GPT), the test beds form a reduced-scale integrated power system, as illustrated in Figure 4.2 [23].

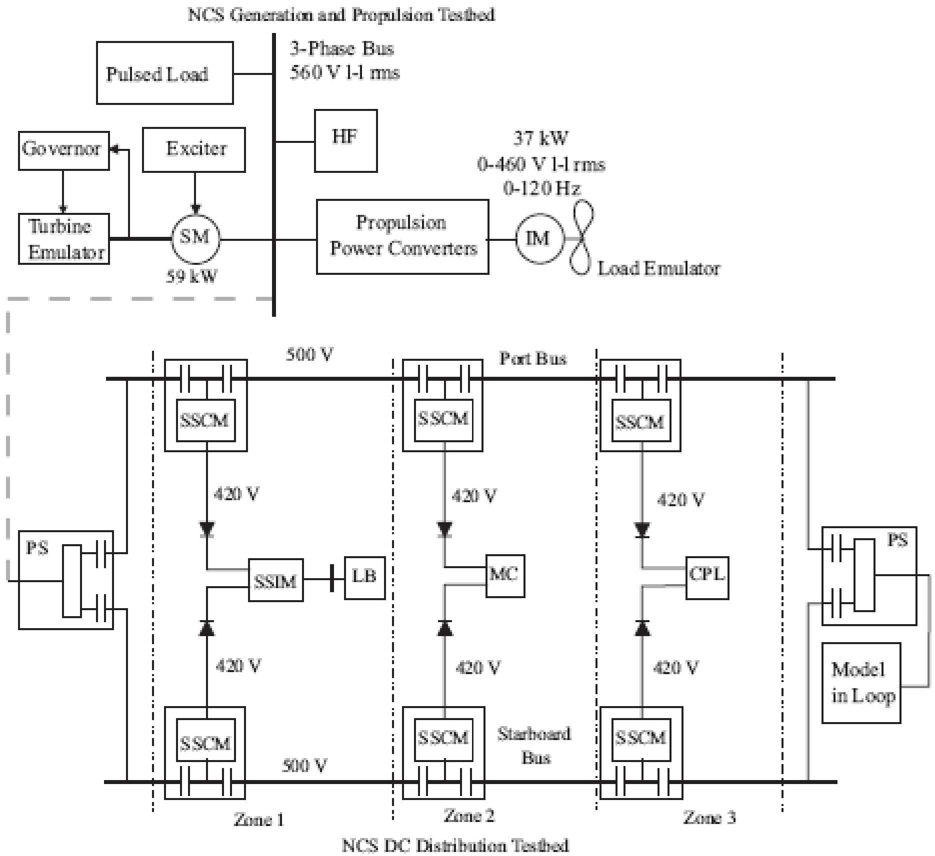


Figure 4.2 An IPS Testbed with AC-DC Zonal Distribution System [23]

The NCS DCDT testbed had been originally housed at the University of Missouri-Rolla, and "operated at a power level that that facilitates performing high risk experiments prior to implementation on full-scale ship hardware." [23]. The NCS DCDT testbed also had been designed to be easily transported and moved to Purdue University now.

An IPS with an AC-DC distribution system uses zonal distribution architecture rather than a ring or radial type to increase accessibility to sources and loads. A typical IPS with a zonal AC-DC distribution system under designers' consideration has two parts, an AC part and a DC part. For the AC part, two main gas turbine generators and two auxiliary generators connect to a common bus (but only one main generator is active at a time). The common bus can supply power to propulsion motors/pulsed weapons, as well as service loads. The redundancy of main generators enables the SPS AC part to handle reconfiguration due to faults or missions transients. Thus, the robustness and survivability of naval electrical ships are increased. For the DC part, there are two power supplies (PS1 and PS2), one feeding the port bus, and the other feeding the starboard bus (only one connection is active at a time). There are three zones of DC distribution. Each zone is fed by a converter module (CM) on the Port bus (CM1, CM2, or CM3) and a converter module on the Starboard bus (CM4, CM5, or CM6) operating from one of the two distribution busses. Diodes prevent a fault from one bus being fed by the opposite bus. The converter modules feature a droop characteristic so that they share power. The three loads consist of an inverter module (IM) that feeds an AC load bank (LB), a motor controller (MC), and a generic constant power load (CPL) [19].

4.2 Design of an IPS with an AC-DC Zonal Distribution System

The IPS in this dissertation is designed to consist of an electric power generation system, electrical propulsion system, and AC-DC zonal distribution system. Also the whole IPS uses almost the same model as a combination of NCS GPT (Naval Combat Survivability, Generation and Propulsion Test-bed) and NCS DCDT (Naval Combat Survivability, DC Distribution Test-bed), except of two differences. The first difference is that the NCS GPT and NCS DCDT are reduced-scale integrated power system [23] and the IPS designed in this research tries to emulate the real power system as the same scale as possible. The second difference is that the AC-DC zonal distribution system in this research utilizes a slightly different topology compared with the original distribution system design in the NCS DCDT testbed, to increase shipboard power system survivability.

The more realistic AC-DC zonal distribution system with some buses in three zone areas designed in NCS DCDT test bed is illustrated in Figure 4.3.

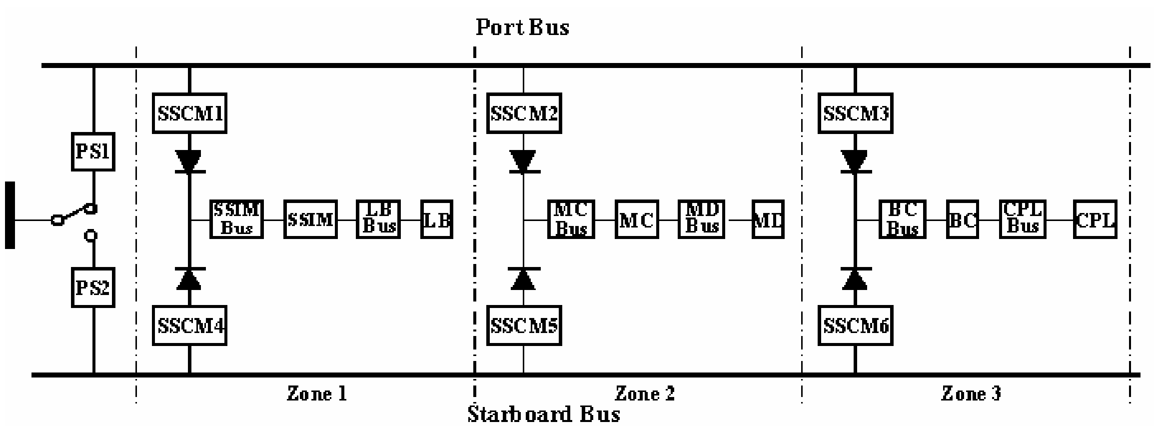


Figure 4.3 An AC-DC Zonal Distribution System

The AC-DC zonal distribution system includes two power supplies, two DC distribution buses, and three zones. At any one time, only one power supply and its distribution bus (for example, PS1 and Port Bus) are connected to the common bus. When a fault happens in the power supply or its distribution bus (such as power supply failure or distribution bus loss), another power supply and its distribution bus will pick up the full system load without interruption in service. If a fault happens in one zone branch (such as SSIM Bus-SSIM-LB Bus-LB branch in Zone 1), this zone branch will be cut off from the distribution bus and lose power. If a fault happens in one of three ship service converter modules (SSCM1, SSCM2, and SSCM3), say SSCM1, the PS1 will be cut off from the common bus and PS2 will connect to common bus, resulting in Zone 1 without interruption in service. For a scenario such as another fault at SSCM5 or SSCM6 following a fault at SSCM1, one zone will lose power no matter which power supply (PS1 or PS2) connects to the common bus, as illustrated in Figure 4.4.

Let six logic variables, A, B, C, D, E, and F, represent the working status of SSCM1, SSCM2, SSCM3, SSCM4, SSCM5, and SSCM6, respectively. Let logic variable H represents the working status of the power supplies PS1 and PS2. We can define logic variable A as [40]

$$A \quad \begin{cases} 1 & \text{if } SSCM1 \text{ is in normal condition} \\ 0 & \text{if } SSCM1 \text{ is in fault condition} \end{cases} \tag{4.1}$$

The definitions of B, C, D, E, and F are similar to A. Logic variable H can be defined as

$$H \quad \begin{cases} 1 & \text{if } PS1 \text{ is in "switch} - \text{on" condition } (PS2 \text{ is in "switch} - \text{off" condition}) \\ 0 & \text{if } PS2 \text{ is in "switch} - \text{on" condition } (PS1 \text{ is in "switch} - \text{off" condition}) \end{cases} \tag{4.2}$$

Assume that in normal condition the PS1 is in "switch on" status, the PS1 will change to "switch off" status whenever a fault happens in either SSCM1, or SSCM2, or SSCM3. Thus the relationship between variable H and variables A, B, and C is [40]

$$H \quad ABC \quad \text{and} \quad \overline{H} \quad \overline{ABC} \quad \overline{A} + \overline{B} + \overline{C} \tag{4.3}$$

For the scenario shown in Figure 3, "one zone lost power" can be described as

$$\left(\overline{D}+\overline{E}+\overline{F}\right)\overline{H} \quad \overline{D}\,\overline{A}+\overline{D}\,\overline{B}+\overline{D}\,\overline{C}+\overline{E}\,\overline{A}+\overline{E}\,\overline{B}+\overline{E}\,\overline{C}+\overline{F}\,\overline{A}+\overline{F}\,\overline{B}+\overline{F}\,\overline{C} \tag{4.4}$$

Thus total nine different scenarios can cause one zone power loss.

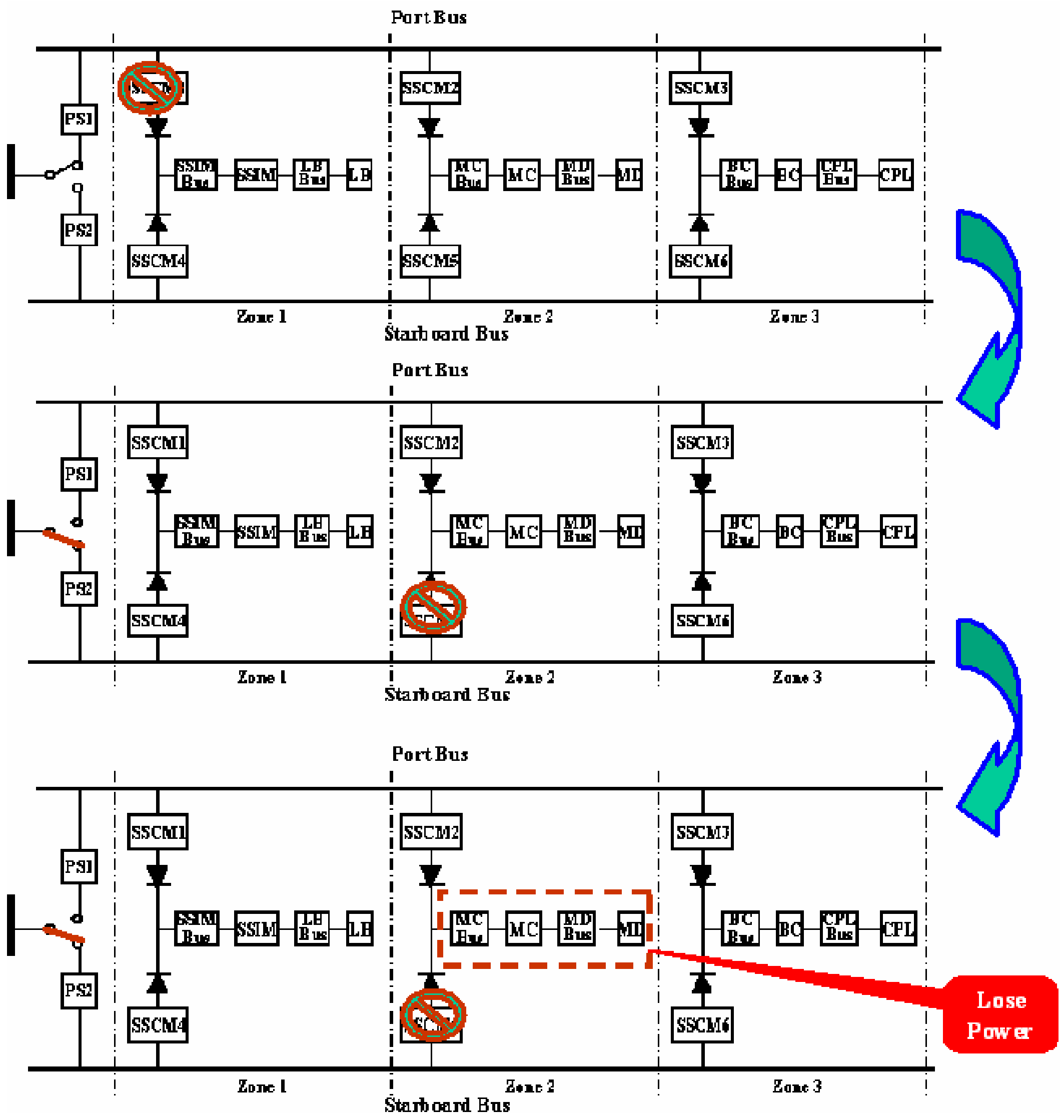


Figure 4.4 One Zone Loses Power for SSCM5 Fault Following SSCM1 Fault [40]

To increase the survivability of the distribution system, the two-direction switch between the common bus and power supplies can be removed and three two-direction switches can be installed between SSCM(1-3) and SSCM(4-6) Bus. The modified AC-DC zonal distribution system is illustrated in Figure 4. 5

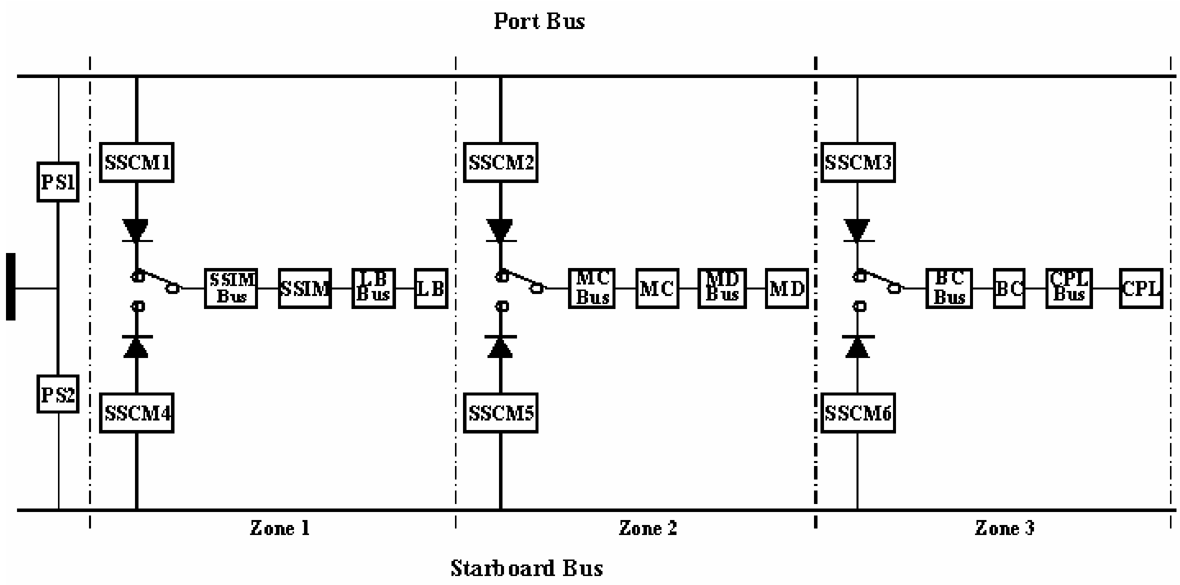


Figure 4.5 Modified AC-DC Zonal Distribution System

With the new topology of the AC-DC zonal distribution system, "one zone lost power" can be described as

$$\left(\overline{D}+\overline{E}+\overline{F}\right)\overline{H} \qquad \overline{D}\;\overline{A}+\overline{E}\;\overline{B}+\overline{F}\;\overline{C} \tag{4.5}$$

Thus the total scenarios, which will cause one zone power loss, decreases from nine for the original topology in Figure 4.3 to three for the new topology in Figure 4.5. To decrease power loss, further design finalizes a new AC-DC zonal distribution topology, in which PS1 and PS2 can be energized either concurrently or separately, as illustrated in Figure 4.6.

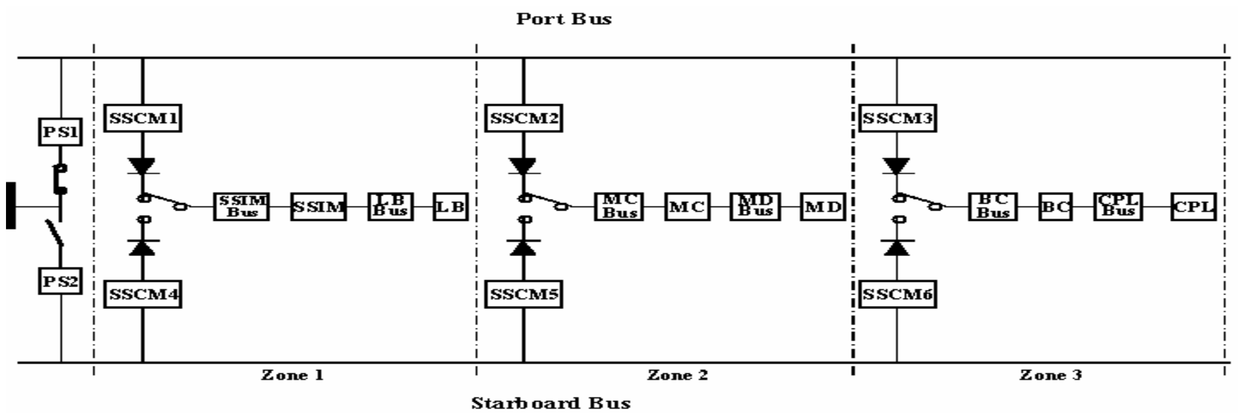


Figure 4.6 Final Redesigned AC-DC Zonal Distribution System

4.3 Summary

This chapter discusses different architectures of IPS, introduces an IPS model developed in the NCS DCDT project, and designs a new IPS with modified AC-DC zonal distribution system to increase survivability of the IPS. This chapter also briefly discusses selection of power converters. Next chapter will present implementation and simulation of the IPS.

CHAPTER V

MAS DESIGN AND SIMULATION FOR THE ELECTRIC PROPULSION SYSTEM

This chapter introduces different power system reconfiguration techniques, discusses MAS applications in power systems, and describes the design of the MAS for the electric propulsion system reconfiguration.

5.1 Reconfiguration Techniques and Multi-Agent System

Reconfiguration techniques can be classified into two categories, centralized and decentralized methods. Centralized reconfiguration techniques include a central control system that knows the entire system information. When faults happen, the central control system will adjust the topology of the entire distribution system to restore power for the unfaulted area, maximize the number of loads supplied, and minimize the switch operations involved in the reconfiguration process. Centralized reconfiguration methods include mathematical programming, meta-heuristics, and knowledge-based programming. Mathematical programming uses optimization techniques and software to calculate the best solution and implement reconfiguration. Examples include linear programming, sequential quadratic programming, Lagrangian and other methods. Meta-heuristics include Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS), and Particle Swarm Optimization (PSO). An Expert System is an example of knowledge-based programming [18]. However, these different methods, as illustrated in

Figure 5.1, have their own drawbacks. Mathematical programming is time-consuming. Meta-heuristics can save computing time but cannot guarantee an optimal solution. All of these techniques are based on centralized structures. Due to the consideration of survivability and robustness, these reconfiguration methods introduced above cannot be widely utilized in shipboard power systems.

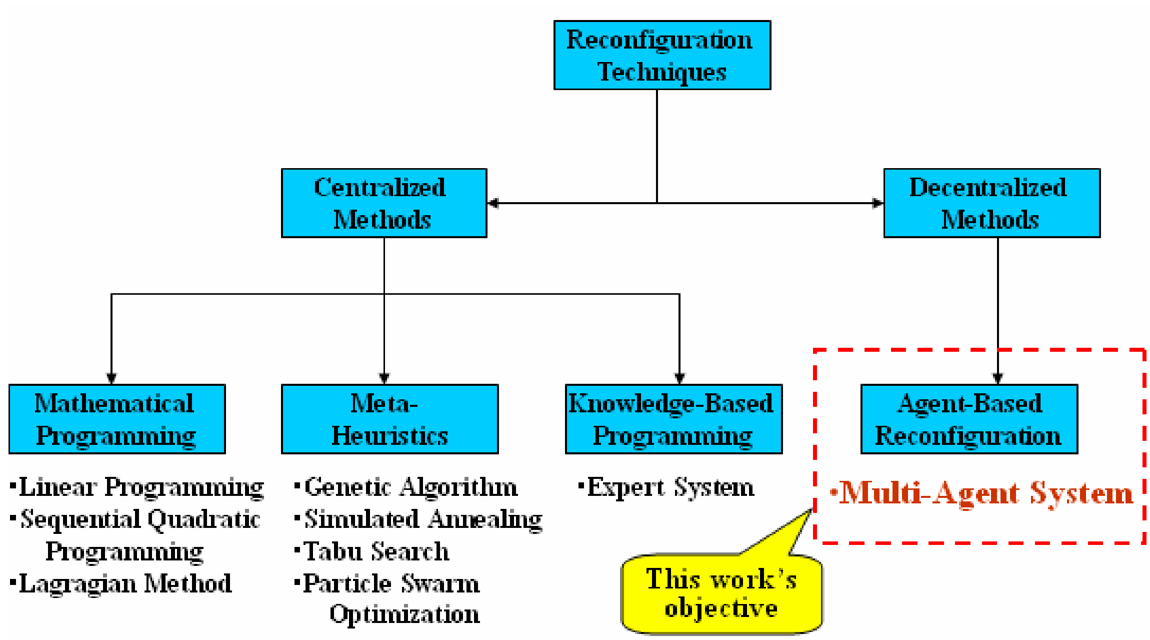


Figure 5.1 Different Reconfiguration Techniques

The integrated power system (IPS) for all-electric ships combines power generation system, electric propulsion system, power distribution system with distributed generators (DG), auxiliary power system for pulsed weapons, and power control and management system all together. Reconfiguration of IPS is needed for two reasons, i.e., faults and task transitions. For the IPS of all-electric warships, the faults may happen due to material casualty of individual equipment (generator, load, cable, and PEBB) or widespread damage due to battle with hostile warships. A warship with full speed escape changing to loiter is an example of task transition.

"Intelligent software agents are defined as being a software program that can perform specific tasks for a user and possesses a degree of intelligence that permits it to perform parts of its tasks automatically and to interact with its environment in a useful manner" [43]. A multi-agent system (MAS) can be defined as a loosely coupled network of problem solvers that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver. The characteristics of MAS are that (1) each agent has incomplete information and local viewpoint for solving the whole system problem; (2) there is no centralized global control system; (3) data are decentralized; and (4) computation is asynchronous [10].

MAS offers two main approaches to developing innovative applications in power systems, i.e., approach to building flexible and extensible hardware/software systems and approach of modeling and simulation [27].

The first application area for MAS is monitoring and diagnostics. Intelligent agents can be used to monitor the output from sensors and inform the engineers or diagnostic algorithms when faults have been detected. MAS can be used to integrate SCADA data, digital fault recorder data, and traveling-wave fault locator data in order to enhance post-fault diagnosis of power system faults [27]. The second application of MAS in power systems is distributed control and restoration. MAS can provide intelligent, fast, and adaptable local control and decision making, which can not be implemented with current central SCADA system and several distributed SCADA systems. The third application of MAS is modeling and simulation. Modern power system operation and planning are more complicated. With its distributed and intelligent features, MAS can be used to model and simulate complex power networks and energy markets. The fourth application field of MAS is power system protection. Intelligent agents can sense faults and issue control signals, taking the same role of modern protective relays.

Because of its decentralized feature and lack of a global control feature, MAS appears to be a preferable candidate for shipboard power system (SPS) reconfiguration.

5.2 MAS Design for the Electric Propulsion System

A typical electric propulsion system consists of a common bus, a transformer, a rectifier, an inverter, a motor drive, and a propeller. To increase naval demand for fight through survivability, an auxiliary electric propulsion system (except of the propeller) has been also installed in larger warships. It is assumed that at any time only one electric propulsion system is working [44]. The whole propulsion system is illustrated in Figure 5.2. If a fault occurs in the primary propulsion system, the power protection system will detect the fault and send a fault signal to the MAS-based reconfiguration system. The reconfiguration system will cut off the primary propulsion system and connect the auxiliary propulsion system into the common bus (power sources). This chapter will discuss the design, implementation, and simulation of the MAS-based reconfiguration system.

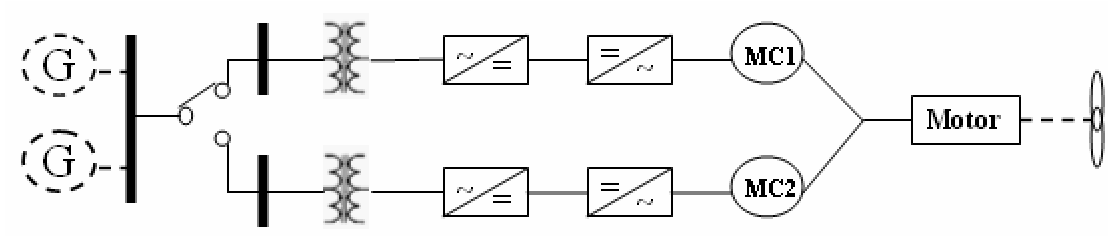


Figure 5.2 Electric Propulsion System

When a fault happens in the power generation system, the total electric power may not be enough to balance propulsion and service loads. The MAS will automatically adjust propulsion load in this case, as discussed in next chapter.

5.2.1 Topology and Reconfiguration Algorithm

For every component in the electric propulsion system, such as transformer, rectifier, and so on, an agent is assigned to associate with the component. For some key components that will execute control function, such as Bus and Bus1, a circuit breaker is also assigned to it. Thus the topology of the MAS is the same as that of electric propulsion system and illustrated in Figure 5.3 [44].

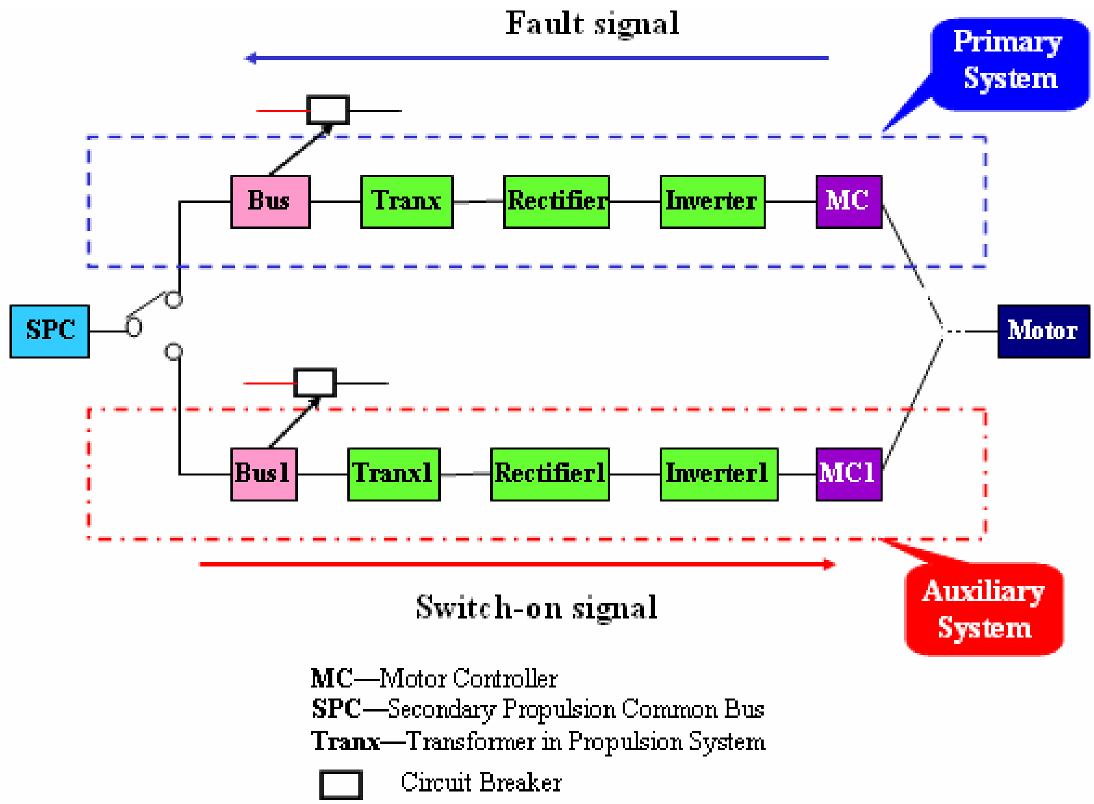


Figure 5.3 Topology of the MAS for Reconfiguration of the Electric Propulsion System

Any agent can only communicate with its neighboring agents, such as upstream neighboring agents and downstream neighboring agents. Thus no global control exists in the MAS. Of course these agents can communicate with each other through broadcasting. But broadcasting increases the communication burden of the MAS and decreases MAS execution speed.

The reconfiguration algorithm is described as following [44]:

If a fault happens in the primary propulsion system, the protection system will send a fault signal to the corresponding agent in the MAS-based reconfiguration system. The corresponding agent will send the fault signal to its upstream neighboring agent, as well as open the circuit breaker associated with it. With the same procedure, the fault signal will be finally relayed to the primary bus agent along the upstream direction. The primary bus agent will open its circuit breaker to cut off the primary propulsion system from the common bus (power sources), as well as send the fault signal to its neighboring auxiliary bus agent. After receiving the fault signal, the auxiliary bus agent sends a “switch-on” signal to its downstream neighboring agent, as well as closes its own circuit breaker. With the same procedure, the “switch-on” signal is finally sent to the auxiliary motor controller agent. After the circuit breaker associated with the auxiliary motor controller agent closes, the auxiliary propulsion system connects into the common bus (power sources) and begins to work. After reconfiguration, the previous auxiliary propulsion system becomes new primary propulsion system, and vice versa. It should be mentioned that the fault signal in the primary propulsion system flows according to the upstream direction, while the “switch-on” signal in the auxiliary propulsion system flows according to the downstream direction. When one propulsion system changes its role, for example, from primary to auxiliary, the content of the signal and flowing direction also change.

For traditional protection and reconfiguration system with relay coordination, all components in the propulsion system can be viewed as different loads. When a fault happens, the corresponding relay can sense it and isolate the corresponding component. For example, if a fault occurs in the inverter component, the protection and reconfiguration system will isolate the inverter by opening corresponding circuit breakers. The objective of the traditional protection and reconfiguration system is to find faults as soon as possible and isolate the faulted area as minimally as possible. However, the function of the propulsion system is to provide power to the propeller and removal of any components in the propulsion system will cut off the propulsion power train. The protection and reconfiguration system has to switch on the auxiliary propulsion system and cut off the primary propulsion system. Of course the decision of swapping roles of these two propulsion systems can be made by a logic controller. But the logic controller has to know the working status of all components in one propulsion system and is a centralized controller, which will increase vulnerability for shipboard power systems.

5.2.2 Agents Design

From Figure 5.3, it can be seen that these agents in the propulsion MAS can be divided into four different types: terminal agent, intermediate agent, bus agent, and secondary propulsion common bus agent. A terminal agent can be defined as an agent that has only one-direction neighboring agents—upstream agents. For instance, the motor agent has only upstream agents—motor controller (MC) agent and motor controller1 (MC1) agent, and is a terminal agent. An intermediate agent can be defined as an agent that has its neighboring agents in both upstream and downstream sides. For example, the motor controller agent, transformer agent, rectifier agent, and inverter agent are intermediate agents. The bus agents have their own downstream neighboring agent, but share the same upstream agent. The secondary propulsion common bus agent is more specific in that it has two downstream neighboring agents and one upstream neighboring agent. Also the secondary propulsion common bus agent needs more complex control—automatically selecting different downstream neighboring agent according to different working condition. Assuming the secondary propulsion common bus agent originally connects the bus agent, for instance, the secondary propulsion common bus agent will automatically switch to the bus1 agent when that bus has a problem. These different agents and their communication are summarized in Figure 5.4. Here it should be mentioned that the primary propulsion system and the auxiliary propulsion system are mutually exclusive. Mutually exclusive means at one time only one propulsion can normally work and another system has to be set up to open or fault, and vice versa.

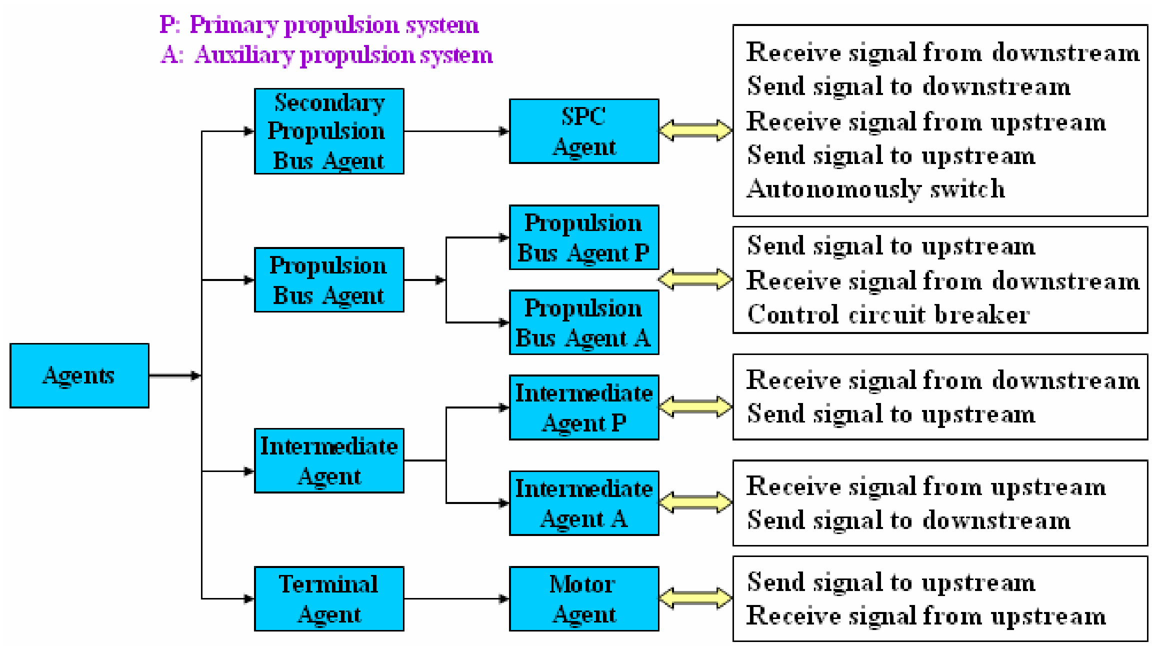


Figure 5.4 Different Agents and their Communication and Control

Because this chapter focuses on reconfiguration of electric propulsion system and the generator agent is not included in the MAS, design and implementation of intermediate agent, terminal agent, propulsion bus agent, and secondary propulsion common bus agent will be discussed in the following sections.

5.2.2.1 Intermediate Agent Design

The intermediate agent is designed as an abstract agent class, and has three inputs, i.e., the name of its upstream agent, the name of its downstream agent, and fault status (fault signal) of itself. The intermediate agent has three working conditions: normal (without fault and in primary propulsion system), fault, and open (without fault and in auxiliary propulsion system). If it receives a fault signal from either the power protection system or its downstream neighboring agent, the intermediate agent will send the fault signal to its upstream neighboring agent, as well as change its working condition from normal to fault. If the intermediate agent is in the auxiliary propulsion system, it will change its working condition to "normal" and send "switch-on" signal to its downstream neighboring agent when a fault happens in the primary propulsion system. Another task the intermediate agent needs to do is to pass information on the load, such as load magnitude and load priority level, to the upstream neighboring agent. Based on this information the propulsion common bus can make decision on propulsion motor output or distribution load shedding.

In the JADE software, the actual job an agent has to do is typically carried out within "behaviors". A behavior represents a task that an agent can carry out and all behaviors in an agent can run asynchronously. JADE software includes three different behaviors: one-shot behavior, cyclic behavior, and generic behavior. One-shot behaviors execute only once. Cyclic behaviors execute forever and repeat its action() method. The ticker behavior executes its onTick() method repetitively, waiting for a given period after each execution. The ticker behavior is a kind of cyclic behavior but the period can be controlled. Generic behaviors embed a status and execute different operations depending on that status.

For the intermediate agent, downstream communications (receiving signal) can be implemented through periodically visiting the downstream neighboring agent and can be achieved by using a cyclic behavior. Upstream communications (sending signal) can be implemented through periodically sending a signal to the upstream neighboring agent and can be achieved by using a ticker behavior. The ticker behavior also includes a one-shot behavior. By using cyclic behavior and ticker behavior for downstream communication and upstream communication, respectively, the pace of sending signals is slower than that of receiving signal, as illustrated in Figure 5.5. This can guarantee that the message queue used for signal storage will not be blocked by oversized message.

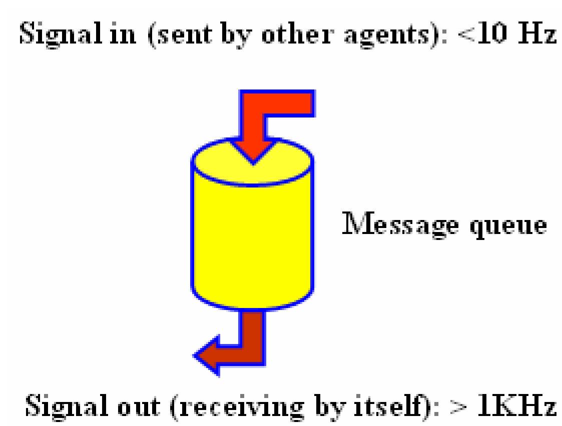


Figure 5.5 Comparison of Signal Sending Pace and Receiving pace

Figure 5.6 shows a flow chart of the intermediate agent class [44]. The transformer agent, rectifier agent, and inverter agent are instances of the intermediate agent class. It should be pointed out that in Figure 5.6 two communication behaviors, i.e., communication with downstream agent and communication with upstream agent, are executing concurrently and asynchronously due to the multi-thread feature of JAVA/JADE software.

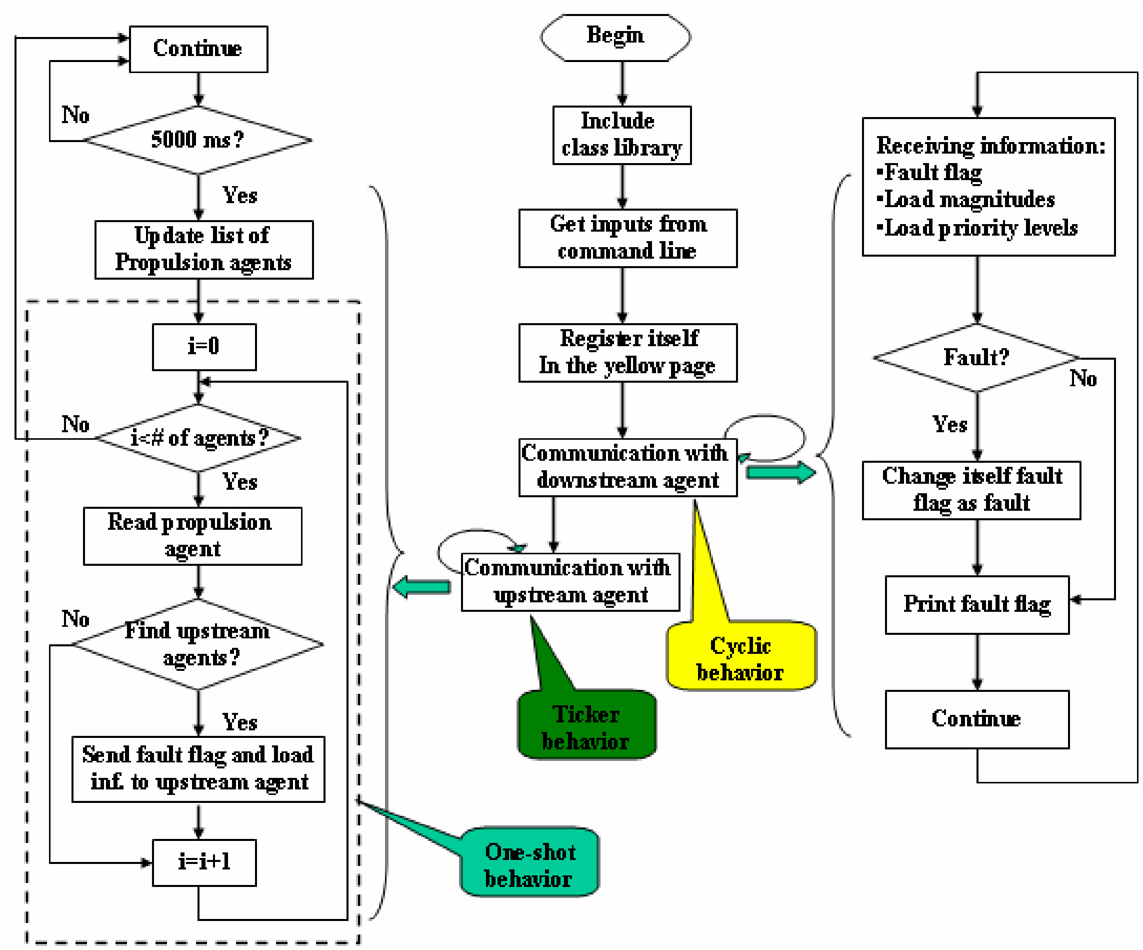


Figure 5.6 Flow Chart of Intermediate Agent Class [44]

According to the flow chart in Figure 5.6, the intermediate agent class is implemented with JAVA/JADE code listed in Appendix.

Given the names of its upstream and downstream neighboring agents, how can an intermediate agent find out its neighboring agent? This can be implemented by a “yellow pages" service provided by JADE package. In JADE software a main container holds two special agents: the Agent Management System (AMS) and Directory Facilitator (DF) [45]. The AMS agent provides a naming service, ensuring that each agent in the platform has a unique name. The DF agent provides a "Yellow Pages" service, which allows agents to publish one or more services they provided so that other agents can find and exploit them. By means of the "Yellow Pages" service, an agent can dynamically find other agents and increase scalability of the whole MAS.

5.2.2.2 Secondary Propulsion Common Bus Agent Design

The secondary propulsion common bus agent is designed as an abstract agent class, and has six inputs, i.e., the name of its upstream agent, the name of its downstream bus agent, the connection flag of its downstream bus agent, the name of its downstream bus1 agent, the connection flag of its downstream bus1 agent, and working condition (fault status) of itself.

The secondary propulsion common bus agent is designed to execute two communication functions. The first communication function is to receive downstream fault flag and load information such as load fault flag, load magnitude, and load priority level from its downstream neighboring agent and send these messages to its upstream neighboring agent. The second communication function is to receive the power injection magnitude from its upstream neighboring agent and send this magnitude to its downstream neighboring agent. Finally the power injection magnitude has been relayed to the motor agent, and the motor agent will adjust propeller thrust according to the power injection magnitude. In addition to the communication functions, the secondary propulsion common bus agent needs to execute a control function—automatically switching to downstream neighboring bus, which is in normal working condition and does not connect to the secondary propulsion common bus, from the current connecting bus that is faulted. The autonomous switching strategy is described in the following flow chart, as illustrated in Figure 5.7.

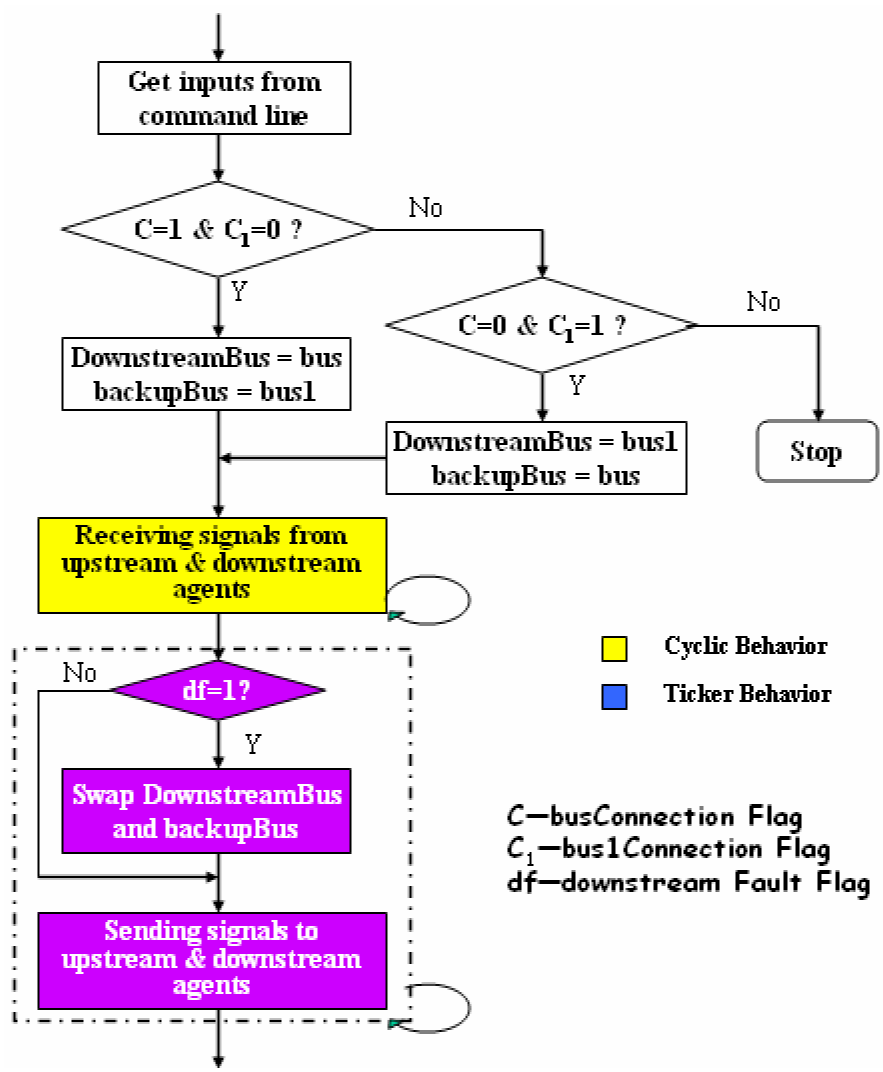


Figure 5.7 Automatic Switching Strategy of Secondary Propulsion Common Bus Agent Class

5.2.2.3 Terminal Agent and Bus Agent Design

The terminal agent is designed to have five inputs, i.e., the name of its upstream agent1, the name of its upstream agent2, load working condition (fault flag), load magnitude, and load priority level. The terminal agent needs to send load information such as fault flag, magnitude, and priority level to its upstream agents, and the load information will be finally relayed to the common bus agent in the power generation MAS. Based on propulsion load information and distribution load information, the common bus agent will make decision and issue control signal to the terminal agent. The control signal will decide propeller working status—full speed moving, half speed moving, or stop. Thus the terminal agent needs one side bi-direction communication—sending to and receiving from its upstream agents. The flow chart of the terminal agent class is illustrated in Figure 5.8.

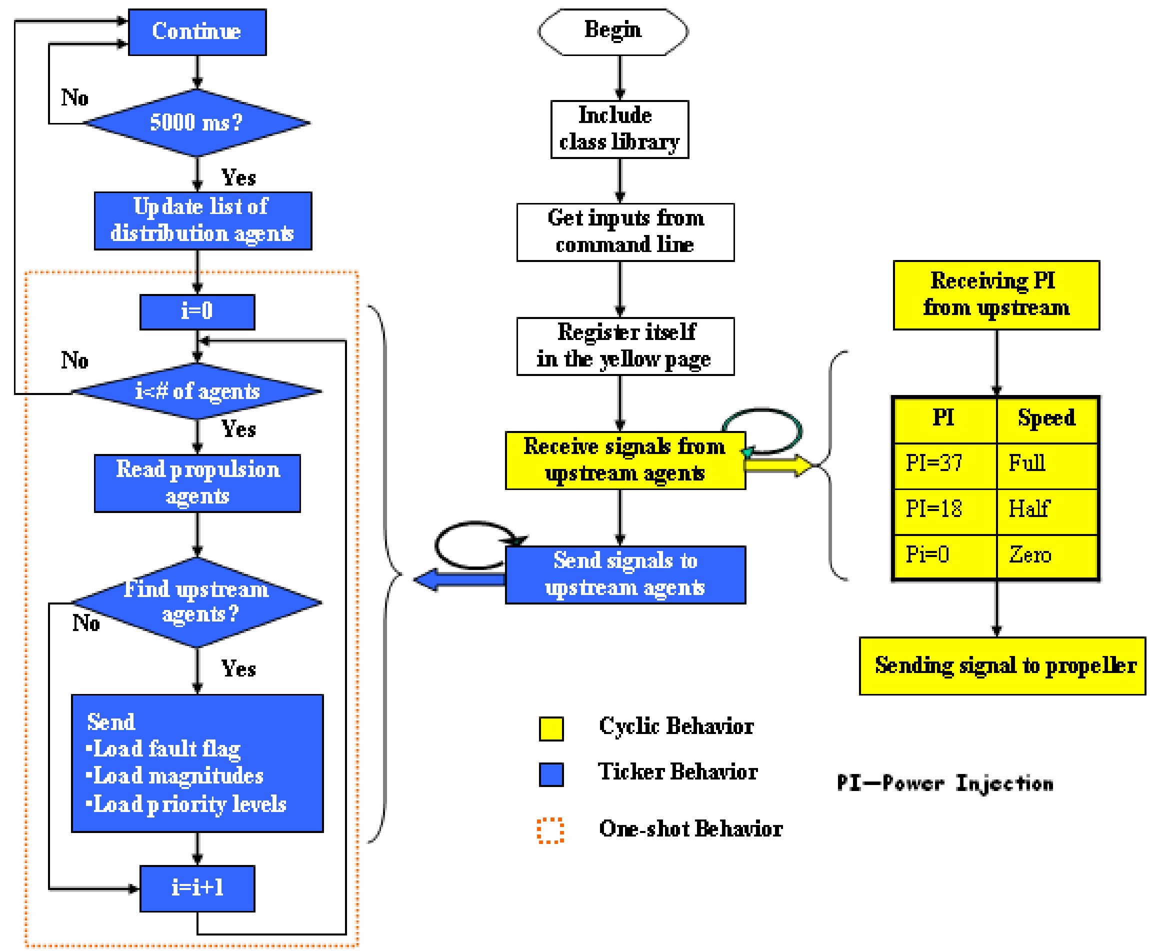


Figure 5.8 Flow Chart of Propulsion Terminal Agent Class

The propulsion bus agent is designed as an abstract agent class, and has four inputs, i.e., the name of its upstream agent, the name of its downstream agent, fault status (fault signal) of itself, and connection flag. Bus agent and Bus1 agent are instances of the propulsion bus agent class. If the bus agent connects to the secondary propulsion common bus agent, the connection flag is equal to 1 and the bus agent will communicate with its upstream neighboring agent—secondary propulsion common bus agent. Otherwise, the connection flag is zero and the bus agent will not communicate with the secondary propulsion common bus agent. The propulsion bus agent is almost the same as intermediate agent, except for having four inputs and conditional upstream communication. Figure 5.9 shows flow chart of the propulsion bus agent.

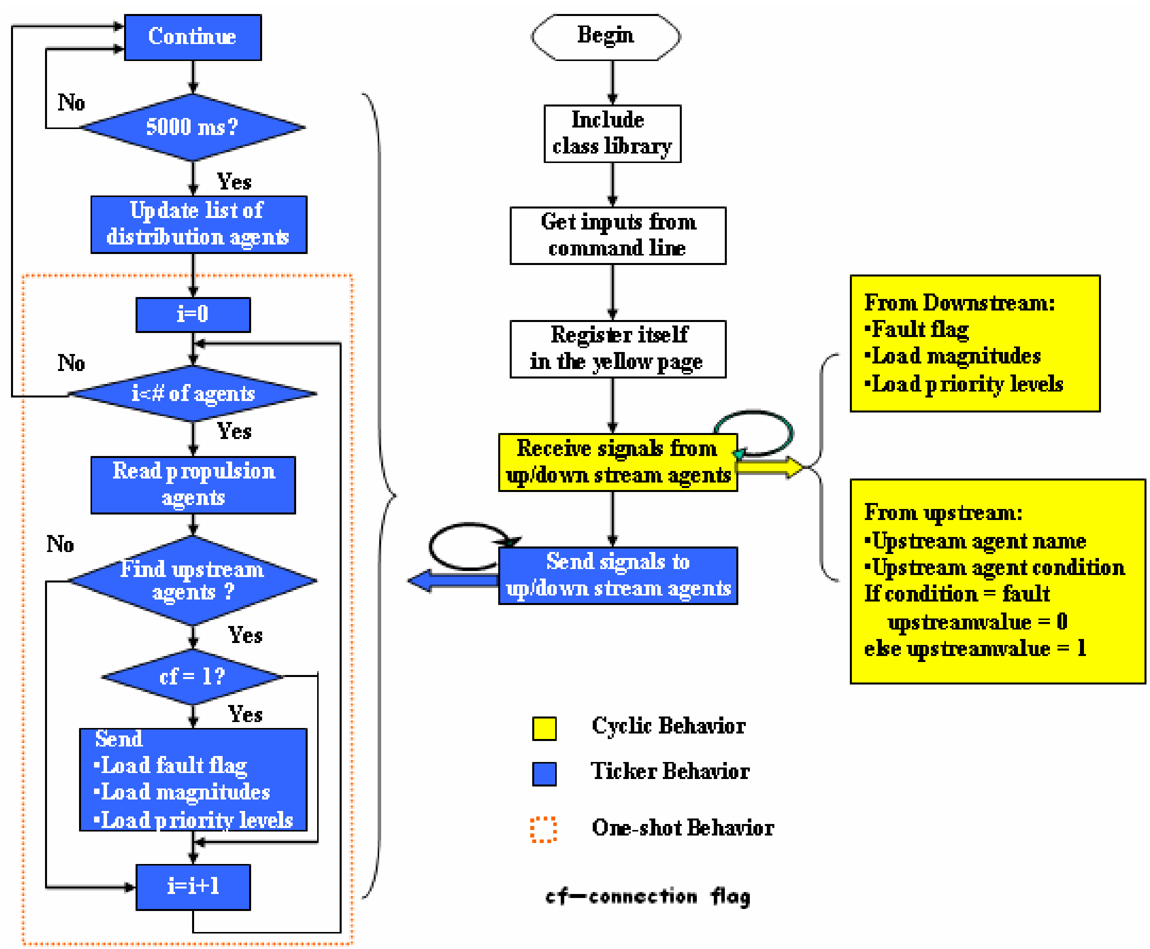


Figure 5.9 Flow Chart of Propulsion Bus Agent Class

5.3 Simulation

In the real world, some physical systems are not available or difficult to implement. Also experiments of physical systems may be expensive and even dangerous.

This is the reason why modeling and simulation techniques are used to simulate real physical systems. In addition, modeling and simulation can be used to demonstrate behavior and performance of the real physical systems, and evaluate functions and effectiveness of the real physical systems. Other advantages of modeling and simulation include suppressing disturbances in noisy environments, and monitoring variables that are not accessible in the real world [42]. For design and implementation of protection and reconfiguration system of ship power systems, it is not practical to build the protection and reconfiguration system prototypes first and then make further modification due to time and cost consideration. Thus modeling and simulation are an effective approach for the protection and reconfiguration system proactive design.

The JADE package is a middleware that facilitates the development of multi-agent systems. The MAS for reconfiguration of a ship electric propulsion system is implemented with JADE 3.4.1 and running in JAVA jdk1.5.0_08 platform. According to the MAS topology in Figure 6.3, the MAS includes twelve agents. They are Motor agent, MC agent, MC1 agent, Inverter agent, Inverter1 agent, Rectifier agent, Rectifier1 agent, Transformer agent, Transformer1 agent, Bus agent, Bus1 agent, and Temp agent. These agents are summarized in Table 5.1. All of these agents get start-up arguments specified on the command line as their inputs, as illustrated in Figure 5.10.

Table 5.1 Propulsion Agents Summary

Agent Name	Agent Class	Inputs (in order)
Motor	PropLoadAgent	Upstream agent name Upstream agent1 name PropLoadAgent working condition Load magnitude Load priority level
MC MC1 Inverter Inverter1 Rectifier Rectifier1 Transformer Transformer1	propintermediateAgent	Upstream agent name Downstream agent name propintermediateAgent working condition
Bus Bus1	propbusAgent	Upstream agent name Downstream agent name propbusAgent working condition propbusAgent connection flag
Temp	tempBusAgent	Upstream agent name Upstream agent connection flag Upstream agent1 name Upstream agent1 connection flag Downstream agent name tempBusAgent working condition

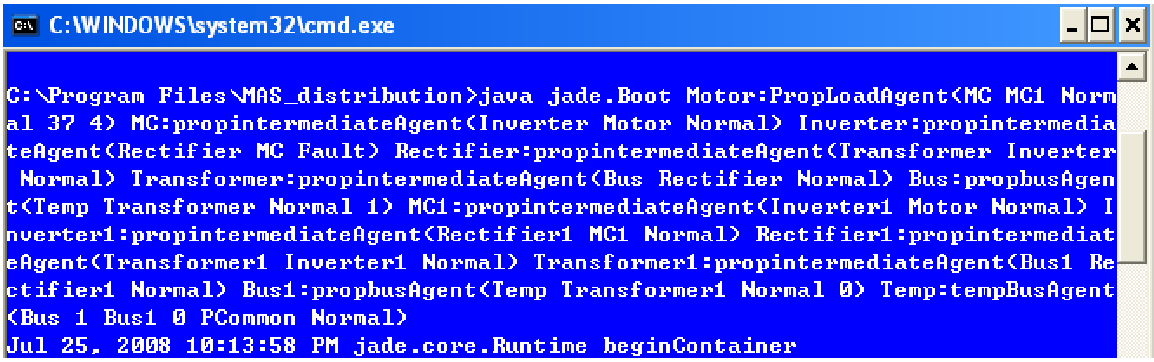


Figure 5.10 Agents Get Their Start-Up Inputs from the Command Line

In JADE software each running instance of the JADE runtime environment is called a container, because it contains several different agents. A platform is defined as a set of active containers in which the first container to start in the platform must be a main container while all other containers are normal containers [45]. The main container differs from other normal containers in that it holds two special agents—AMS and DF. The AMS (Agent Management System) ensures that each agent in the platform has a unique name and has the authority to create or kill other agents. The DF (Directory Facilitator) provides a Yellow Pages service by which an agent can find other agents that provide the services it requires. The following figure illustrates platforms, a main container, normal containers, AMS, and DF [45].

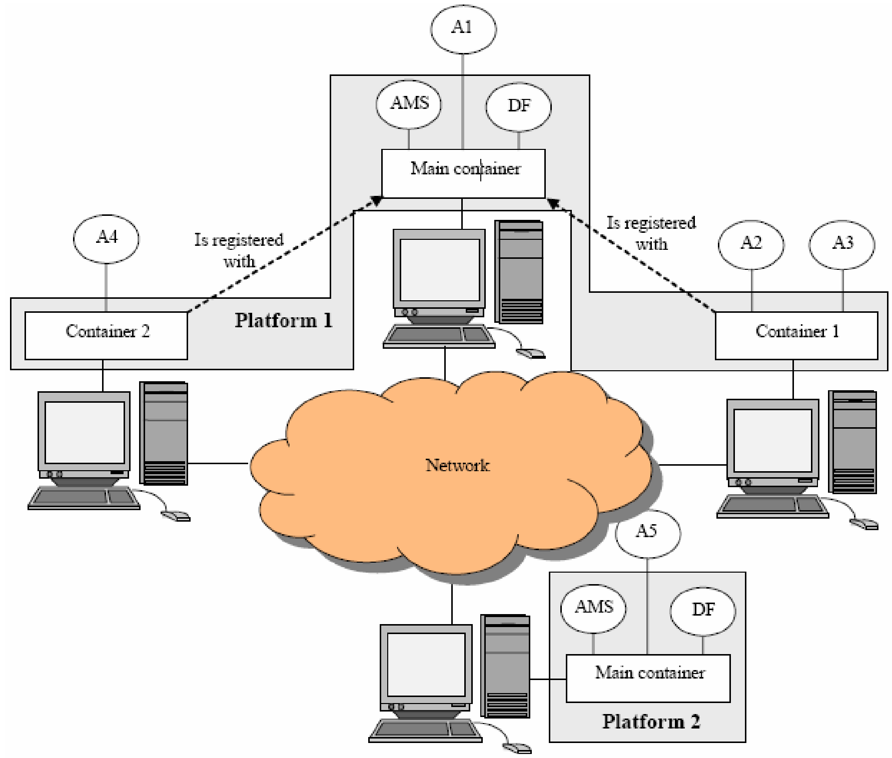


Figure 5.11 Platforms, Containers, AMS, and DF [45]

Figure5.12 shows a main container, named Main-Container@73v3q71, created and initialized by launching JADE software. The main container includes twelve propulsion agents. These propulsion agents and their initial working condition and neighboring agents are illustrated in Figure 5.13.

```
Jul 25, 2008 10:13:58 PM jade.core.Runtime beginContainer
INFO: ----------------------------------
    This is JADE 3.6 - revision 6032 of 2008/05/05 14:07:10
    downloaded in Open Source, under LGPL restrictions,
    at http://jade.tilab.com/
----------------------------------------
Jul 25, 2008 10:14:03 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Jul 25, 2008 10:14:03 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Jul 25, 2008 10:14:03 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Jul 25, 2008 10:14:03 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Jul 25, 2008 10:14:03 PM jade.core.messaging.MessagingService clearCachedSlice
INFO: Clearing cache
Jul 25, 2008 10:14:03 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParse
rImpl$JAXPSAXParser
Jul 25, 2008 10:14:03 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://76.114.1.134:7778/acc
Jul 25, 2008 10:14:04 PM jade.core.AgentContainerImpl joinPlatform
INFO: --------------------------------------
Agent container Main-Container@73v3q71 is ready.
--------------------------------------------
```

Figure 5.12 A Main Container Initialization

```
Hello! Propulsion bus agent Bus@73v3q71:1099/JADE is ready.
  The upstream neighbor of Bus is Temp
  The downstream neighbor of Bus is Transformer
  Initial operating condition of the Bus is Normal
  Upstream connection of the Bus is 1
Hello! Intermediate-agent Rectifier1@73v3q71:1099/JADE is ready.
  The upstream neighbor of Rectifier1 is Transformer1
  The downstream neighbor of Rectifier1 is Inverter1
  Initial operating condition of the Rectifier1 is Normal
  Initial switch status is On
Hello! Intermediate-agent Inverter@73v3q71:1099/JADE is ready.
  The upstream neighbor of Inverter is Rectifier
  The downstream neighbor of Inverter is MC
  Initial operating condition of the Inverter is Fault
Hello! Intermediate-agent Inverter1@73v3q71:1099/JADE is ready.
  The upstream neighbor of Inverter1 is Rectifier1
  The downstream neighbor of Inverter1 is MC1
  Initial operating condition of the Inverter1 is Normal
Hello! Intermediate-agent Rectifier@73v3q71:1099/JADE is ready.
  The upstream neighbor of Rectifier is Transformer
  The downstream neighbor of Rectifier is Inverter
  Initial operating condition of the Rectifier is Normal
Hello! Propulsion bus agent Bus1@73v3q71:1099/JADE is ready.
  The upstream neighbor of Bus1 is Temp
  The downstream neighbor of Bus1 is Transformer1
  Initial operating condition of the Bus1 is Normal
Hello! Propulsion load agent Motor@73v3q71:1099/JADE is ready.
  The upstream neighbors of Motor are MC and MC1
  Initial operating condition of the Motor is Normal
  Initial power the load needs is 37 MW
Hello! Intermediate-agent Transformer@73v3q71:1099/JADE is ready.
  The upstream neighbor of Transformer is Bus
  The downstream neighbor of Transformer is Rectifier
  Initial operating condition of the Transformer is Normal
Hello! Propulsion temp bus agent Temp@73v3q71:1099/JADE is ready.
  The downstream bus of Temp is Bus
  The upstream neighbor of Temp is PCommon
  Initial operating condition of the Temp is Normal
Hello! Intermediate-agent MC1@73v3q71:1099/JADE is ready.
  The upstream neighbor of MC1 is Inverter1
  The downstream neighbor of MC1 is Motor
```

Figure 5.13 Propulsion MAS Initialization

Every agent in the MAS needs to register in the “yellow pages” of the DF agent, in order to be found by other agents. Figure 5.14 shows the content of the “yellow pages”. All of twelve agents are registered. The registration order is random, showing that these agents are running concurrently and asynchronously.

```
Motor found the following distribution agents:
  j = 0  Transformer
  j = 1  Motor
  j = 2  Temp
  j = 3  MC1
  j = 4  Rectifier1
  j = 5  Transformer1
  j = 6  MC
  j = 7  Inverter
  j = 8  Inverter1
  j = 9  Rectifier
  j = 10  Bus1
  j = 11  Bus
```

Figure 5.14	Propulsion Agents Registration

Here a simple test case, Test Case 1, is designed in which only the initial working condition of the inverter agent is set “fault”. The initial condition of the propulsion system is illustrated in Figure 5.15. After reconfiguration and communication among these agents, the bus agent is set to “switch-off” and the bus1 agent “switch on”. Also the bus agent connection flag switches to zero from one and the bus1 agent connection flag switches to one from zero, indicating that the primary propulsion system is cut off from the common bus and the auxiliary propulsion system connects to the common bus. The post-reconfiguration situation of the propulsion system is illustrated in Figure 5.16. The simulation results are shown in Figure 5.17.

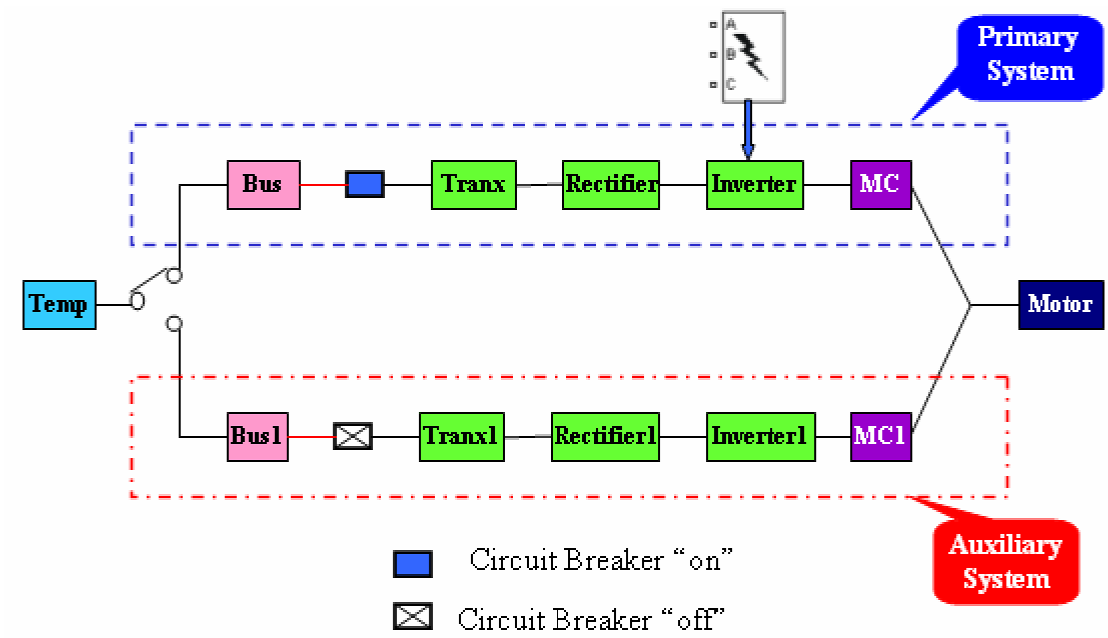


Figure 5.15 Initial Condition of the Propulsion System In Test Case 1

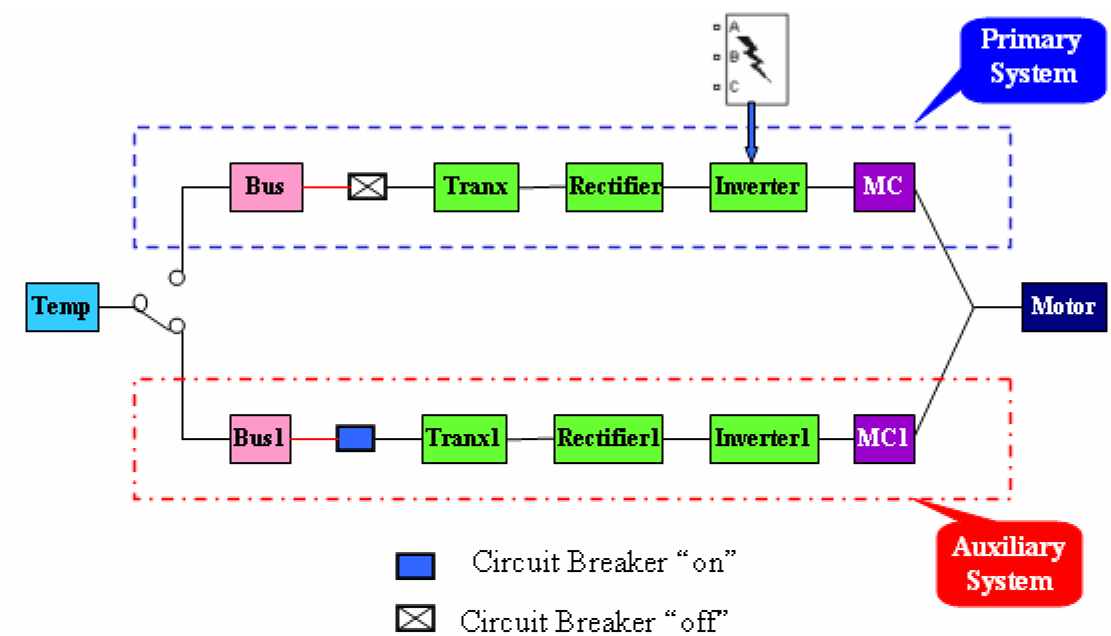


Figure 5.16 Post-Reconfiguration Situation of the Propulsion System In Test Case 1

```
Inverter switch status: On
Inverter1 switch status: On
Transformer1 switch status: On
Rectifier1 switch status: On
Rectifier switch status: Off
  Propbus connected to Temp: Bus1
  Bus1 working condition of Temp: Normal
  Load working condition of Temp: Normal
  Load of Temp = 37 MW
  Load level of Temp = 4
Transformer switch status: Off
Motor current working condition: Normal
MC1 switch status: On
MC switch status: On
MC1 switch status: On
Bus current working condition: Normal
Bus switch status: Off
  Downstream working condition of Bus: Fault
  Load working condition of Bus: Normal
  Downstream load of Bus = 37 MW
  Downstream priority level of Bus = 4
Bus1 current working condition: Normal
  Connection name of Bus1: Bus1  connection: 1
Bus current working condition: Normal
  Connection name of Bus: Bus1  connection: 0
Bus1 current working condition: Normal
  Connection name of Bus1: Bus1  connection: 1
Bus1 current working condition: Normal
Bus1 switch status: On
  Downstream working condition of Bus1: Normal
  Load working condition of Bus1: Normal
  Downstream load of Bus1 = 37 MW
  Downstream priority level of Bus1 = 4
```

Figure 5.17 Simulation Results of Test Case 1

5.4 Summary

This chapter designs and implements the propulsion MAS, which includes twelve different agents instanced from four different abstract agent classes. To decrease size and weight, the MAS may be designed that only the key control agents have circuit breakers Next chapter will discuss distribution MAS.

CHAPTER VI

MAS DESIGN AND SIMULATION FOR THE AC-DC DISTRIBUTION SYSTEM

Different architectures of IPS for future all-electric ships are in consideration. These architectures can be classified as two categories: IPS with AC distribution system and IPS with AC-DC distribution system. To increase survivability with less than 100% redundancy and meet the Navy's demand, the IPS with AC-DC zonal distribution system is preferred and designed in the NCS DCDT. To decrease the possibility of zones losing power, a modified AC-DC zonal distribution system was designed, as described in Chapter IV.

This chapter presents topology and reconfiguration algorithms of MAS for the AC-DC zonal distribution system, and discusses the design and simulation of the MAS.

6.1 Topology and Algorithms of the MAS

6.1.1 MAS Topology

For every component in the AC-DC zonal distribution system, an agent is assigned to associate with the component. The distribution system topology is illustrated in Figure 6.1, and the topology of the MAS should be the same as the distribution system topology, as illustrated in Figure 6.2.

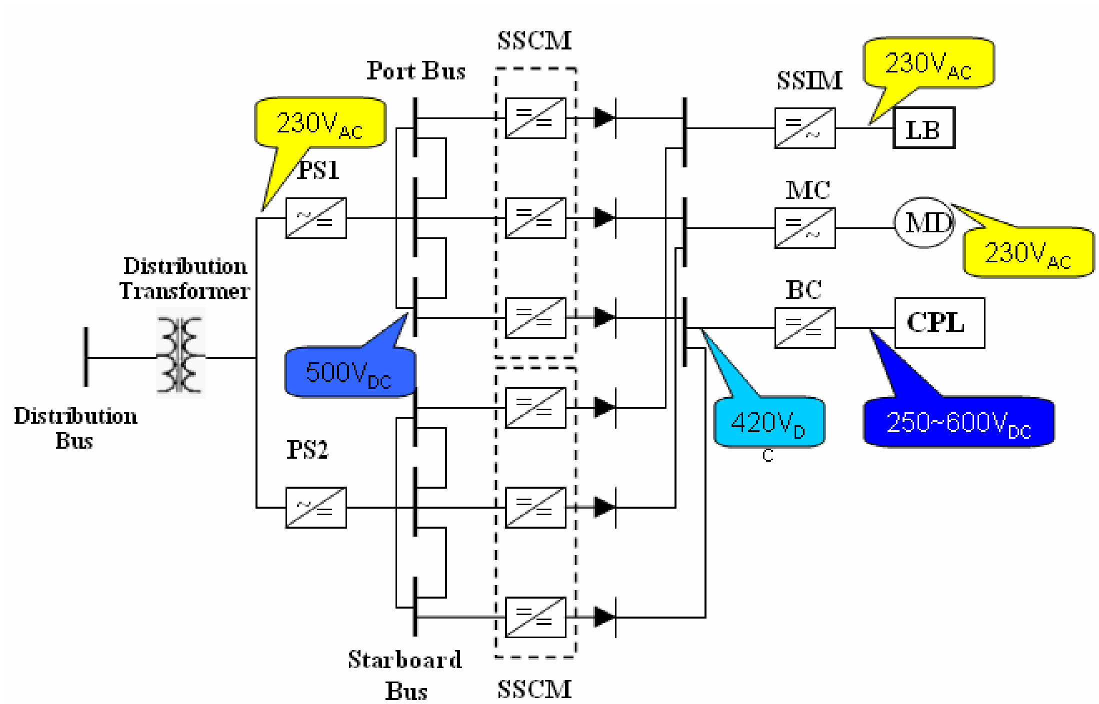


Figure 6.1 Topology of the AC-DC Zonal Distribution System

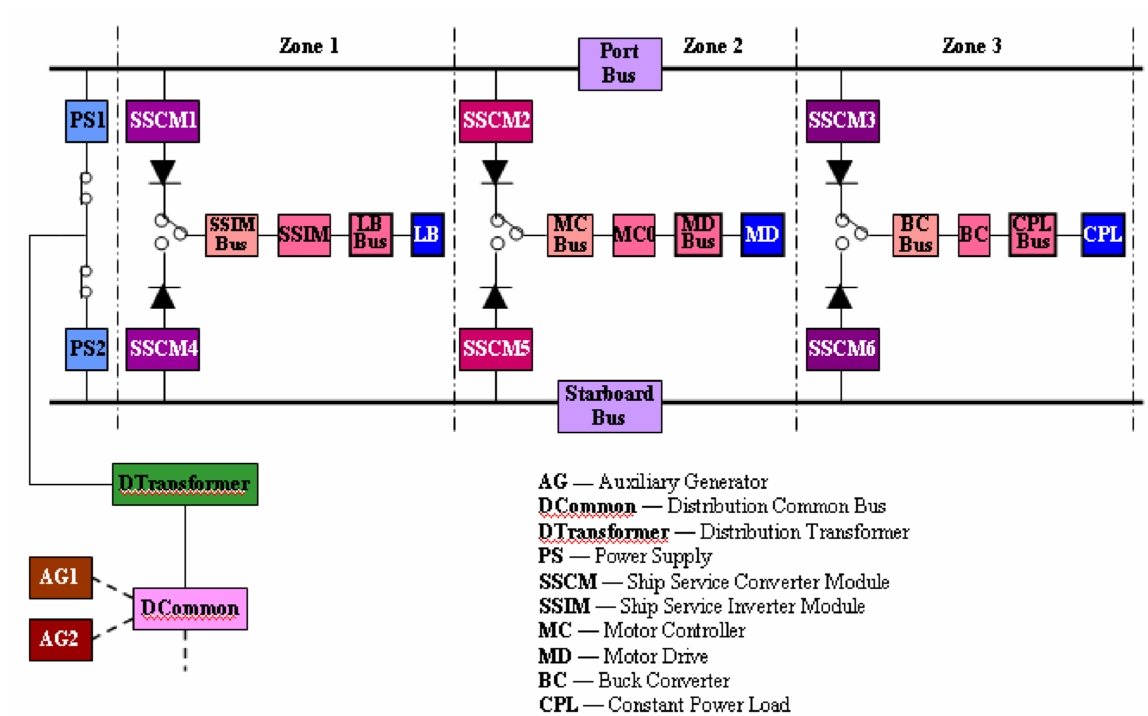


Figure 6.2 Topology of the MAS for the Distribution System

Also a circuit breaker is assigned to associate with every key control agent, such as converter bus agent and power supply agent, as discussed in the following sections.

6.1.2 Reconfiguration Algorithms

The reconfiguration algorithm for distribution MAS is more complicated than that of the propulsion MAS. Different reconfiguration algorithms should be designed for faults happening in different areas. Also to implement different reconfiguration functions, such as islanding and load shedding, different algorithms are needed.

Algorithm 1—Fault happens in a distribution zone

If one fault happens within one zone of three distribution zones, for instance, the SSIM in the zone 1 has a problem; the protection system will detect the fault and send a fault signal to the SSIM agent. The SSIM agent will forward the fault signal to its upstream neighboring agent—SSIM Bus agent, and the SSIM Bus agent can cut off the zonal branch through associated circuit breaker.

Algorithm 2—Fault happens in any SSCMs

If a fault happens in one SSCM of six SSCMs, for instance, SSCM1, the protection system will detect the fault and send a fault signal to the SSCM1 agent. The SSCM1 agent will send the fault signal to its downstream neighboring agent--SSIM bus agent. And the SSIM bus agent can automatically make a connection between the SSCM4 and SSIM bus through the two ends switch.

Algorithm 3—Fault happens in any AGs

If a fault happens in one auxiliary generator, for instance, the AG1, the protection system will detect the fault and send a fault signal to the AG1 agent. The AG1 agent will send the fault signal to its downstream neighboring agent--DCommon agent. The DCommon agent will communicate with the PCommon agent to collect information on total generation, load magnitudes and priority levels. If the generation is not enough to the total load demands, the DCommon and PCommon agents will decide which load(s) will be decreased or shed down according to load priority levels.

6.2 MAS Design for the AC-DC Zonal Distribution System

The objective of the MAS is to implement reconfiguration algorithms through distributed agents. Each agent only fulfils its own objective. However, the whole objective of the MAS is achieved through implementing every agent's objectives.

The MAS for the AC-DC Zonal Distribution System reconfiguration consists of twenty-three agents. These agents can be divided into three different groups, distribution zone (branch) group, SSCM group, and interface and control group. The following sections describe design of agents in different groups.

6.2.1 Design of Agents in Distribution Group

There are three distribution zone groups, corresponding to three distribution zones. The distribution zone group1 includes SSIM Bus agent, SSIM agent, LB Bus agent, and LB agent. The distribution zone group2 includes SSIM Bus agent, SSIM agent, LB Bus agent, and LB agent. The distribution zone group3 includes BC Bus agent, BC agent, CPL Bus agent, and CPL agent. Thus the distribution zone groups have a total of twelve agents. These agents can be classified into three categories, i.e., load agent, intermediate agent, and converter bus agent. The first two agents are almost the same as a terminal agent and intermediate agent in the propulsion MAS discussed in Chapter VI. However, the converter bus agent in the distribution MAS differs from the bus agent in the propulsion MAS. The following sections will discuss the design of the load agent, intermediate agent, and converter bus agent in the distribution MAS.

6.2.1.1 Design of Load Agent and Intermediate Agent

The load agent is designed to have four inputs, i.e., the name of its upstream neighboring agent, its working condition (fault flag), load magnitude, and load priority level. The objective of the load agent is to send its load magnitude and load priority level to its upstream neighboring agent. This information will be finally relayed to distribution common bus agent and propulsion common bus agent, which will make a decision whether or not to execute load shedding and islanding. Another objective of the load agent is to send its working condition (fault flag) information to its upstream neighboring agent, and this information will be finally received by the converter bus agent which decides whether or not to cut off the distribution branch. Thus the load agent needs one direction communication—sending signals to its upstream neighboring agent. The flow chart of the load agent is illustrated in Figure 6.3. It is almost the same as that of terminal agent in propulsion MAS, except for one direction communication.

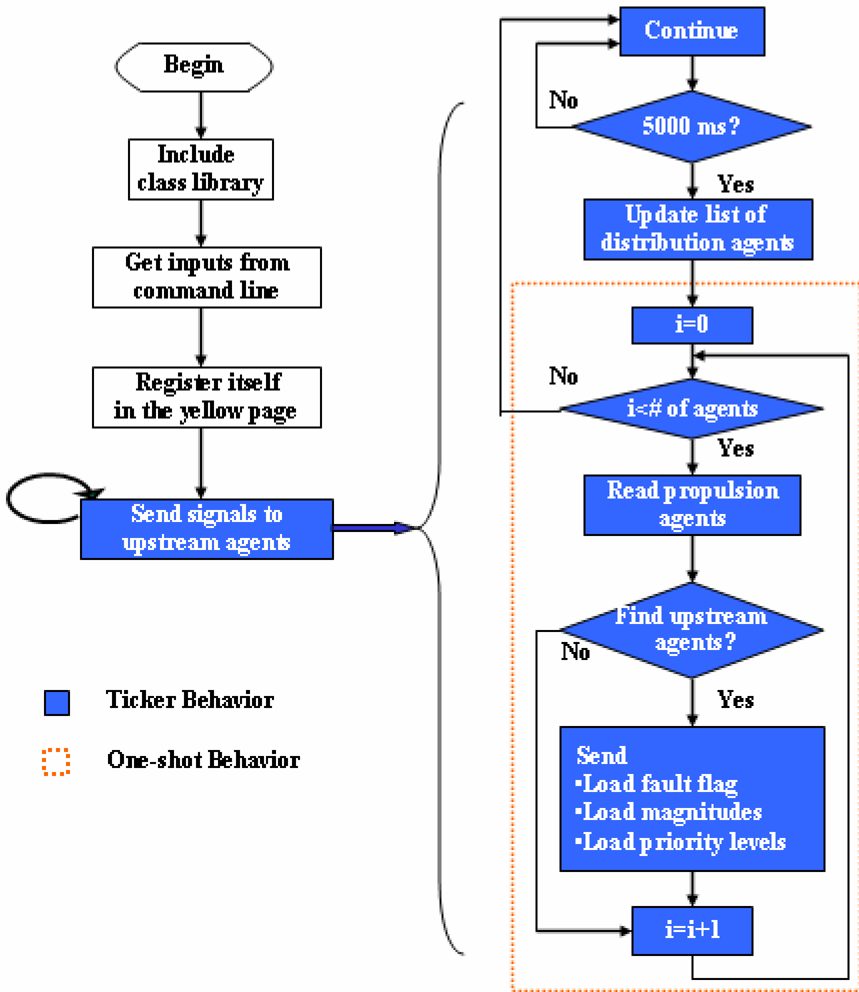


Figure 6.3 Flow Chart of the Load Agent in the Distribution MAS

The intermediate agent is designed to have three inputs, i.e., the name of its upstream neighboring agent, the name of its downstream neighboring agent, and its working condition (fault flag). One objective of the intermediate agent is to receive downstream working condition, load magnitude, and load priority level from its downstream neighboring agent. Second objective of the intermediate agent is to send fault flag, load magnitude, and load priority level to its upstream neighboring agent. The fault flag is defined as a logic variable:

$$
\textit{fault flag} \quad \begin{cases} \textit{current agent working condition} & \textit{if current working condition} \quad \textit{fault} \\ \textit{downstream working condition} & \textit{otherwise} \end{cases} \tag{6.1}
$$

Thus the intermediate agent needs two direction communications. The flow chart of the intermediate agent is shown in Figure 6.4.

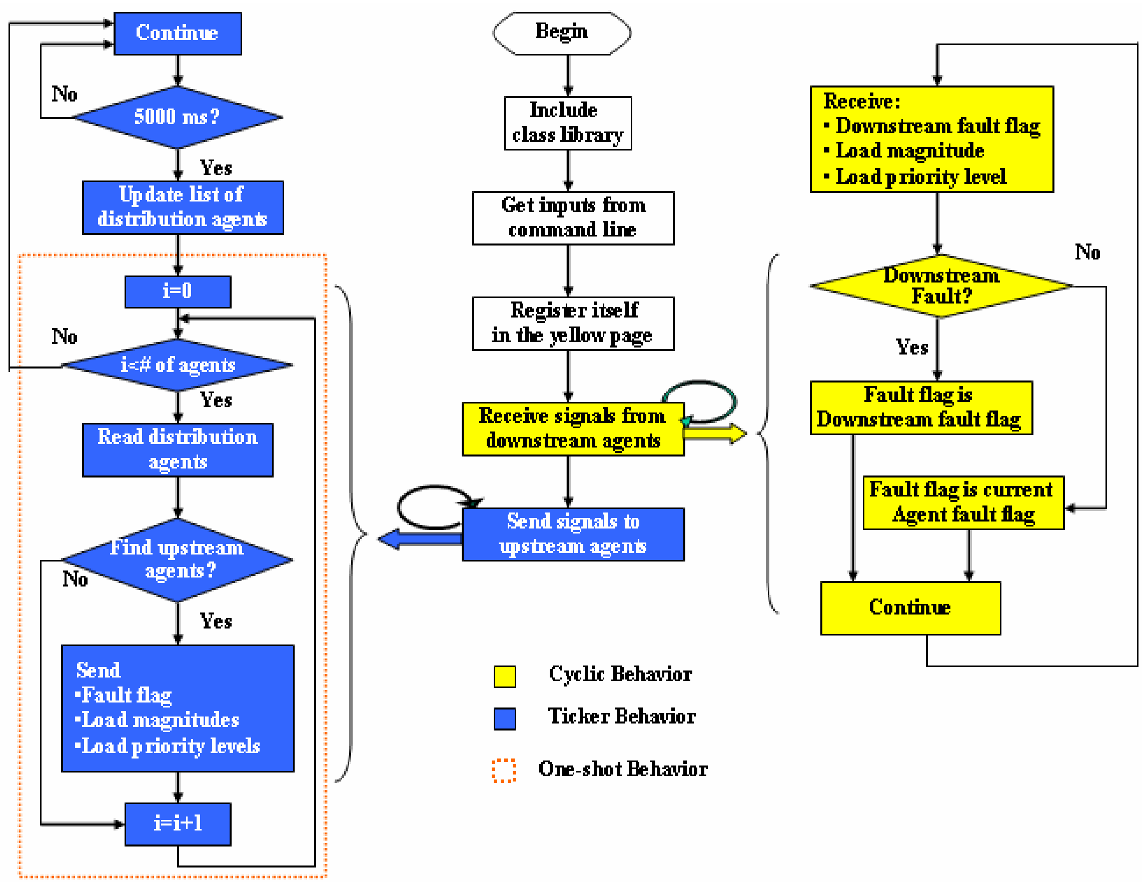


Figure 6.4 Flow Chart of the Intermediate Agent in the Distribution MAS

6.2.1.2 Design of Converter Bus Agent

The converter bus agent is designed to have six inputs. They are the name of upstream neighboring agent1, connection flag of the upstream neighboring agent1, name of upstream neighboring agent2, connection flag of the upstream neighboring agent2, name downstream neighboring agent, and its working condition. Like other distribution intermediate agents, the converter bus agent will receive signals such as downstream fault flag, load magnitude, and load priority level from its downstream neighboring agent, and send these signals to its upstream neighboring agent. If the downstream fault flag is "fault", the bus agent will cut off the downstream distribution branch. Thus the converter bus agent executes its first control function, based on the downstream fault flag. Also, the converter bus agent receives signals from its upstream neighboring agent, such as the name and fault flag of the upstream neighboring agent. If the fault flag is "fault", the converter bus agent will automatically switch to another upstream neighboring agent, executing its second control function. As a summary, the communication function and control function of the converter bus agent are illustrated in Figure 6.5.

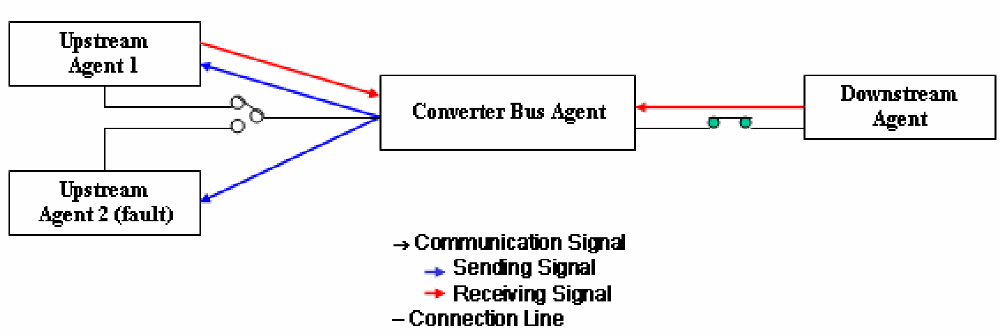


Figure 6.5 Communication and Control Function of the Converter Bus Agent in the Distribution MAS

To avoid message queue blockage by an oversized message, the function of receiving signals is implemented with cyclic behavior, and the function of sending signals with ticker behavior. Control functions can be implemented with one-shot behavior. The flow chart of the converter bus agent is illustrated in Figure 6.6.

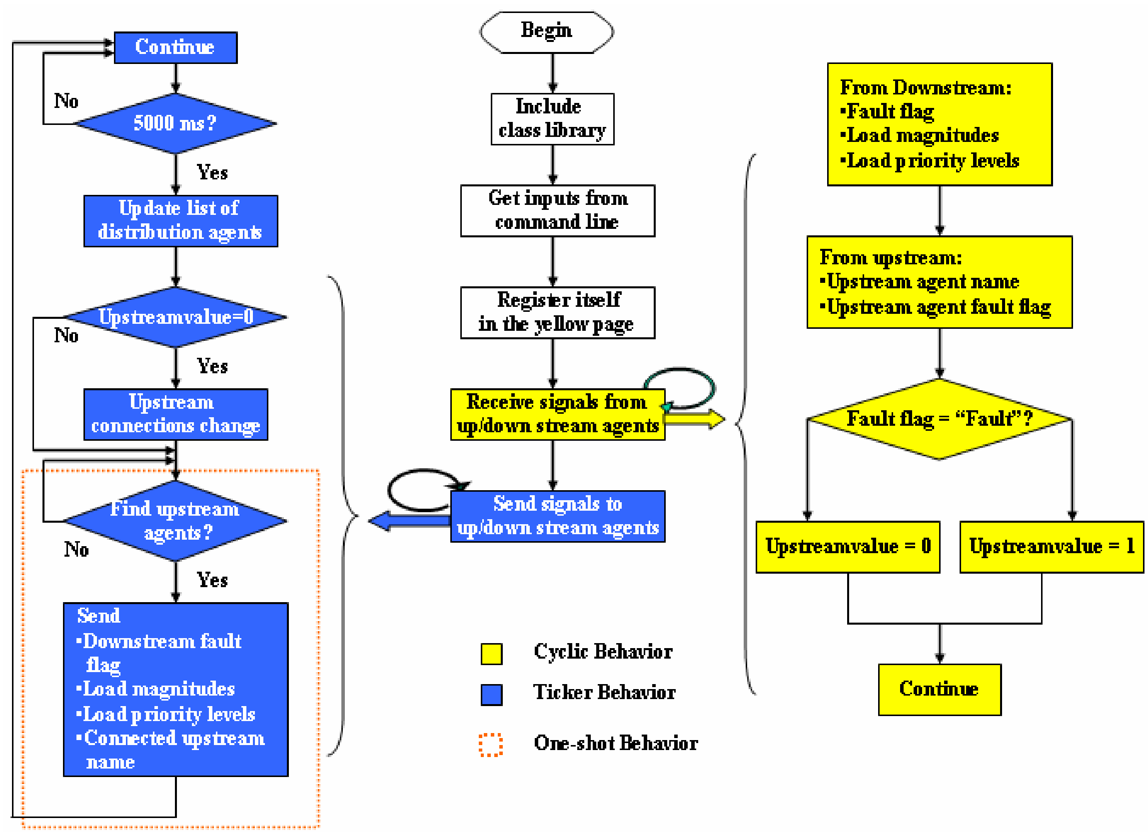


Figure 6.6 Flow Chart of the Converter Bus Agent in the Distribution MAS

6.2.2 Design of Agents in SSCM Group

The SSCM group includes six agents. They are SSCM1, SSCM2, SSCM3, SSCM4, SSCM5, and SSCM6 agent. These agents are instances of three abstract agent classes, i.e., SSCM14, SSCM25, and SSCM36. SSCM1 agent and SSCM4 agent are two instances of the SSCM14 agent class, and mutually exclusive. That means, at a time only one instance of the SSCM4 agent class can connect to the converter bus agent and another instance will be backup and disconnect from the converter bus agent. SSCM2 agent and SSCM5 agent are mutually exclusive too, and so are SSCM3 agent and SSCM6 agent. This section will presents design of SSCM14 agent class. The SSCM25 and SSCM36 agent classes will not be discussed here, because they are almost the same as SSCM14 agent class except of different signal performatives.

The SSCM14 agent class is designed to have four inputs, i.e., the name of its upstream neighboring agent, the name of its downstream agent, its working condition (fault flag), and downstream connection flag. The SSCM14 receives downstream fault flag, load magnitude, and load priority level from its downstream neighboring agent, and sends this information to its upstream neighboring agent. In addition, the SSCM14 also receives the name of its downstream connecting agent, to which it will send back its own name and fault flag. Based on the fault flag of SSCM14, the converter bus agent will make control decisions for switching. So the SSCM14 receives signals only from its downstream neighboring agent and sends signals to both its downstream and upstream neighboring agents.

The communication ability is one of the most important features of JADE software. JADE agents can exchange messages with each other in a format defined by the FIPA (Foundation for Intelligent Physical Agents) international standard for agent interoperability. The message format includes a number of fields, such as sender, receivers, performative, content, language, ontology, and so on. "Performative" is the communicative intention that the sender wants to achieve by sending the message. The performative may be REQUEST, INFORM, QUERY_IF, CFP, PROPOSE, and so on. To create consistency among all agents, whenever possible, communication that sends signals in the downstream direction will use CFP performative, and communication that sends signals in the upstream direction will use REQUEST performative. According to this rule, the SSCM14 agent sends signals to its downstream neighboring agent – SSIM bus agent in CFP performative, and to its upstream neighboring agent – Port Bus or Starboard Bus agent in REQUEST performative.

To implement these communication functions, the SSCM14 agent class is designed based on following flow chart:

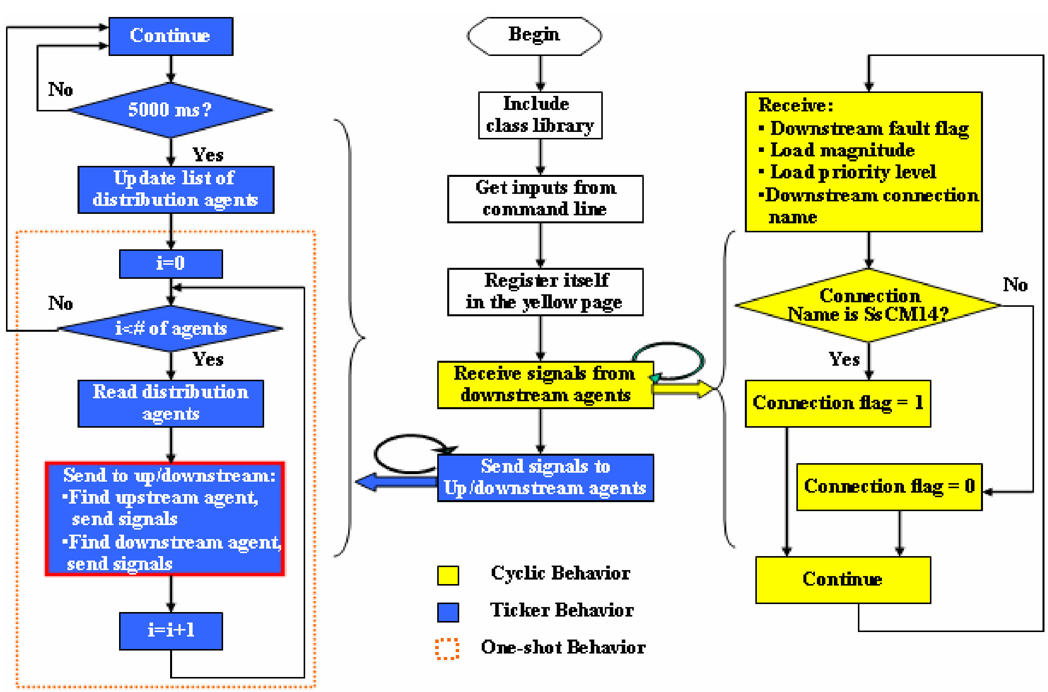


Figure 6.7 Flow Chart of the SSCM14 Agent Class in the Distribution MAS

In Figure 6.7, the red outlined block in the left side shows the sending functions. The more details, such as signals that are sent out, destinations of these signals, and message performatives, are shown in Figure 6.8.

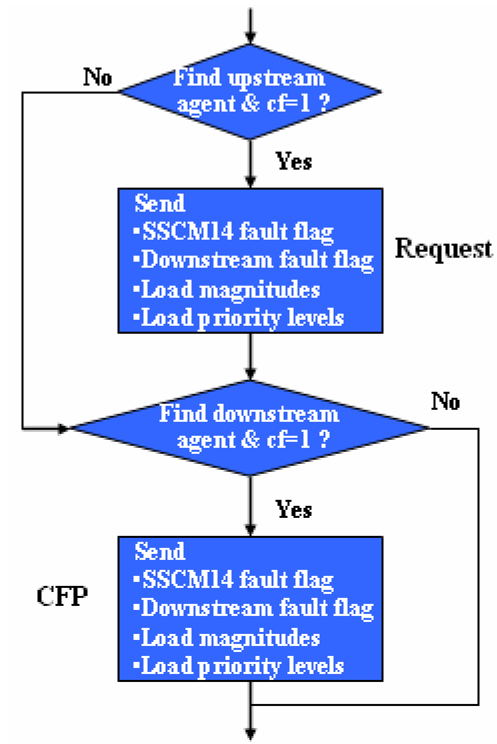


Figure 6.8 Sending Functions in the SSCM14 Agent Class in the Distribution MAS

6.2.3 Design of Agents in Interface and Control Group

The interface and control group includes PortBus agent, StarboardBus agent, PS1 agent, PS2 agent, and DTransformer agent. PortBus and StarboardBus agents are two instances of PSbus agent class, and DTransformer agent is the instance of distransformer agent class. This section will discuss design of the PSbus agent class, PS1 agent class, PS2 agent class, and distransformer agent class.

6.2.3.1 Design of PSbus Agent Class

The PSbus agent class is designed to have five inputs, including the name of its upstream neighboring agent, the name of its downstream neighboring agent1, the name of its downstream neighboring agent2, the name of its downstream neighboring agent3, and its fault flag. The PSbus agent class will receive signals from its three downstream neighboring agents. These signals includes the name and fault flag of the downstream neighboring agent, and fault flag, load magnitude, and load priority level of downstream distribution branch. These signals are shown in Figure 6.9.

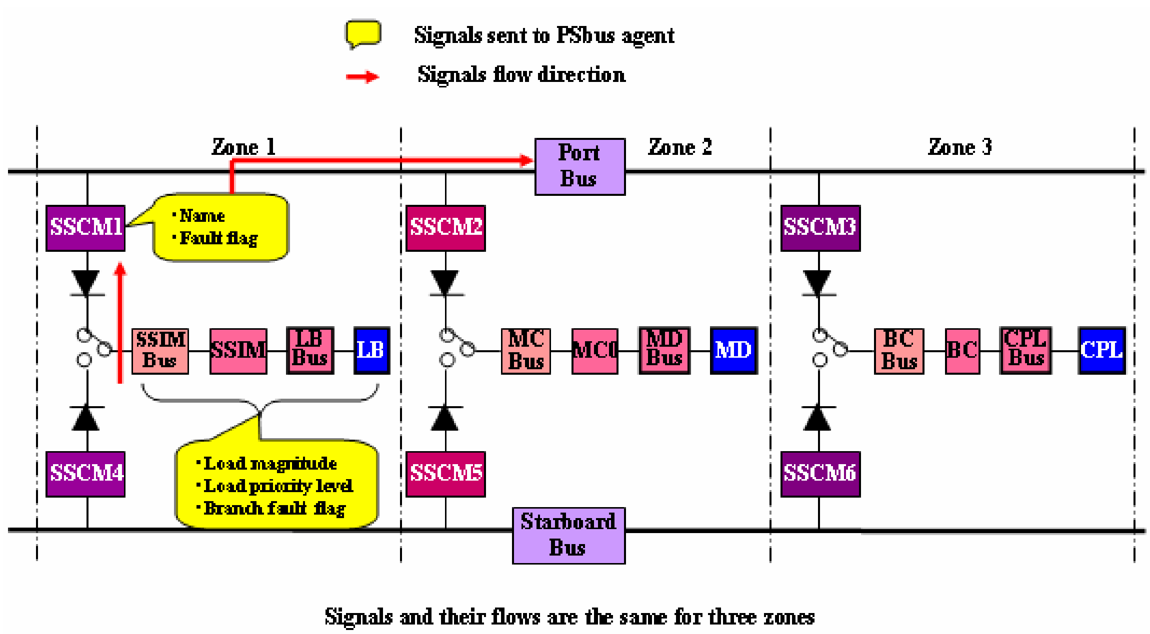


Figure 6.9 Signals Flow into PSbus Agent Class in the Distribution MAS

The PSbus agent class will send signals, including its fault flag, three downstream fault flags, and load magnitudes and priority levels for three distribution zones, to its upstream neighboring agent – PS1/PS2 agent. The flow chart of the PSbus agent class is almost the same as intermediate agent class in the propulsion MAS discussed in last chapter, and illustrated in Figure 6.10.

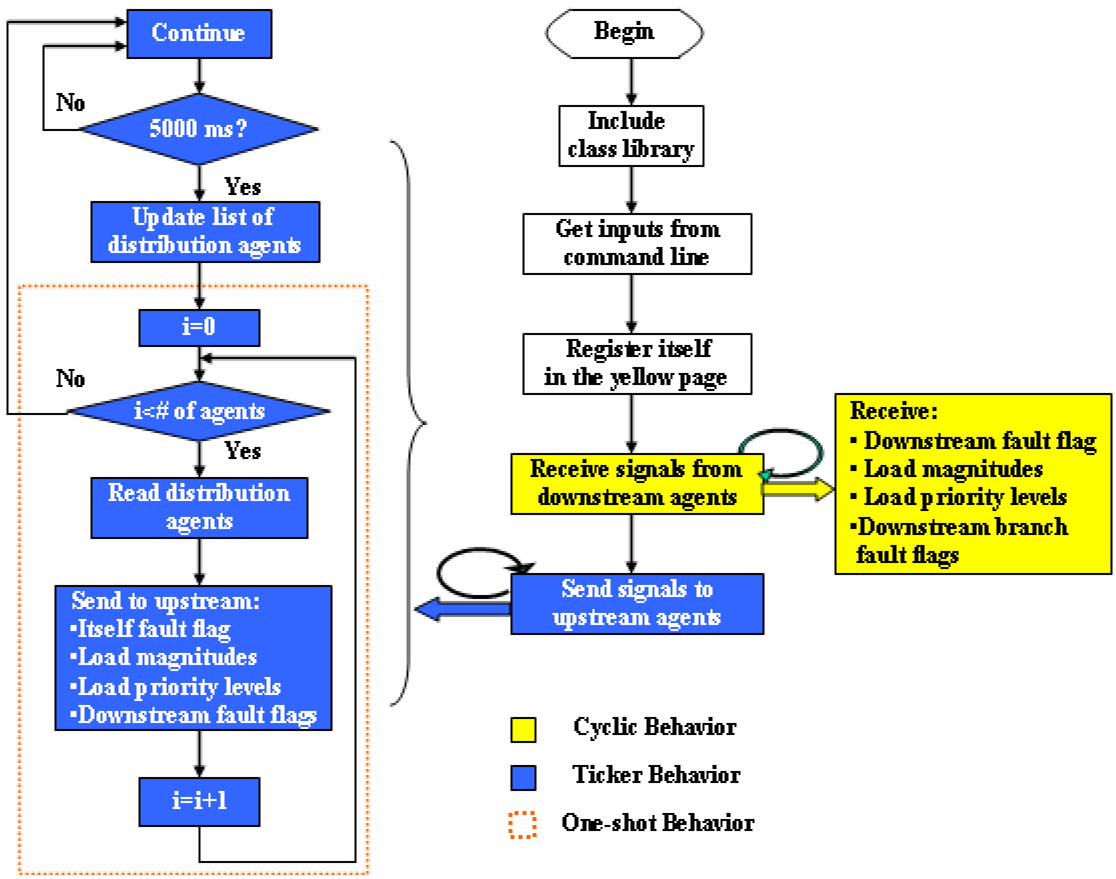


Figure 6.10 Flow Chart of the PSbus Agent Class in the Distribution MAS

6.2.3.2 Design of PS1 and PS2 Agent Class

The PS1 and PS2 agent classes are almost the same, except for different sending message performatives. Thus only design of PS1 is discussed here.

The PS1 agent class is designed to have three inputs, i.e., the name of upstream neighboring agent, the name of downstream neighboring agent, and its fault flag. The PS1 agent class will receive thirteen signals from its downstream neighboring agent – PortBus agent. These messages are listed in Table 6.1. Based on the message it received, the PS1 agent class will send signals to its upstream neighboring agent according to the algorithm illustrated in Figure 6.11. The flow chart of the PS1 is shown in Figure 6.12.

Table 6.1 Signals Sent to Upstream Neighboring Agent by PS1 Agent Class

Order	Signal
1	PortBus fault flag
2	SSCM1 fault flag
3	SSCM2 fault flag
4	SSCM3 fault flag
5	Distribution branch1 fault flag
6	Distribution branch2 fault flag
7	Distribution branch3 fault flag
8	Load1 magnitude
9	Load2 magnitude
10	Load3 magnitude
11	Load1 priority level
12	Load2 priority level
13	Load3 priority level

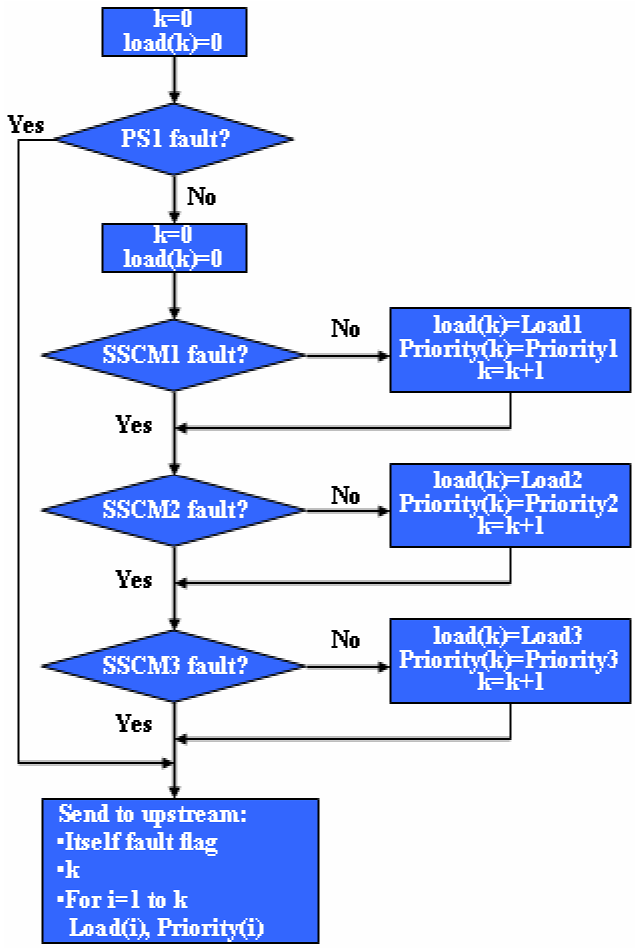


Figure 6.11 Algorithm of Sending Signals for the PS1 Agent Class in the Distribution MAS

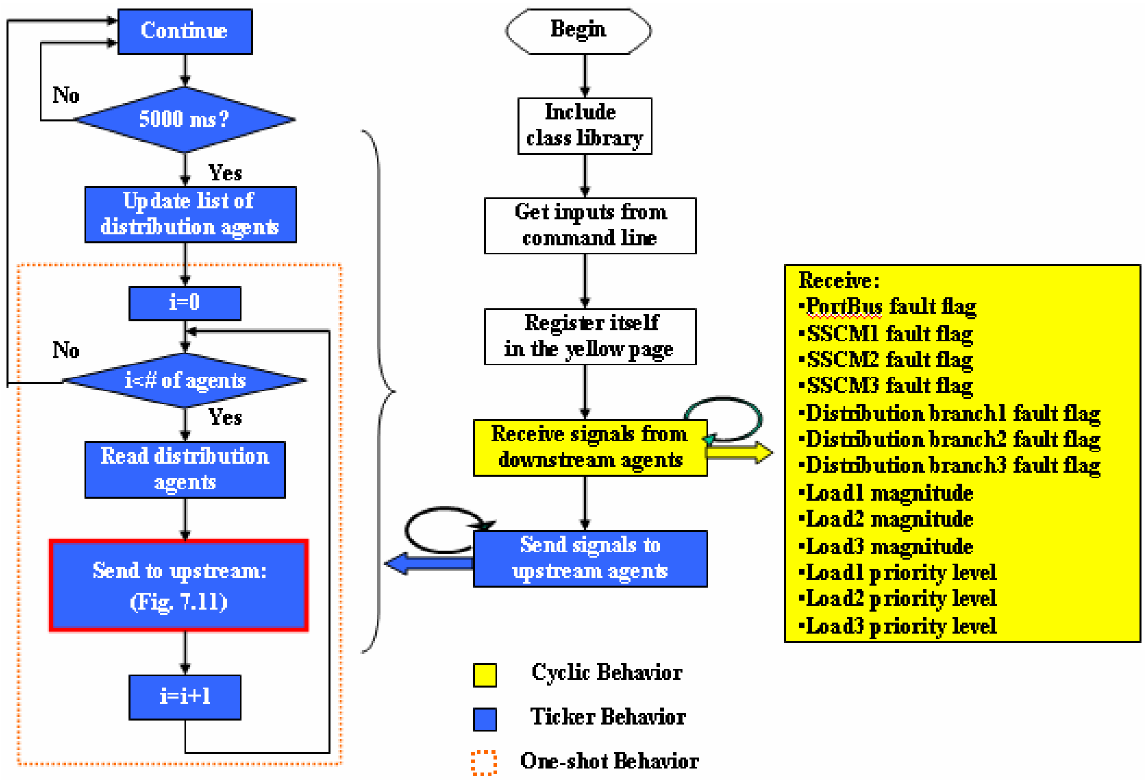


Figure 6.12 Flow Chart of the PS1 Agent Class in the Distribution MAS

6.2.3.3 Design of Distransformer Agent Class

The distransformer agent class is designed to have four inputs, i.e., the name of its upstream neighboring agent, the name of its downstream neighboring agent1, the name of its downstream neighboring agent2, and its fault flag. The distransformer agent class will receive the fault flag of its downstream neighboring agent, magnitudes and priority levels of the distribution branch loads that are sent to its downstream neighboring agent, from each of its downstream neighboring agents. To receive a specific number of signals from its downstream neighboring agent, the distransformer agent needs to know the number of signals to be sent. The distribution branch loads can be sent to distransformer agent through either PS1 or PS2. How can the distransformer agent recognize the number of loads from PS1 or PS2? This problem can be solved by sending a control signal, k, from PS1 and PS2. The algorithm for receiving function is shown in Figure 6.13.

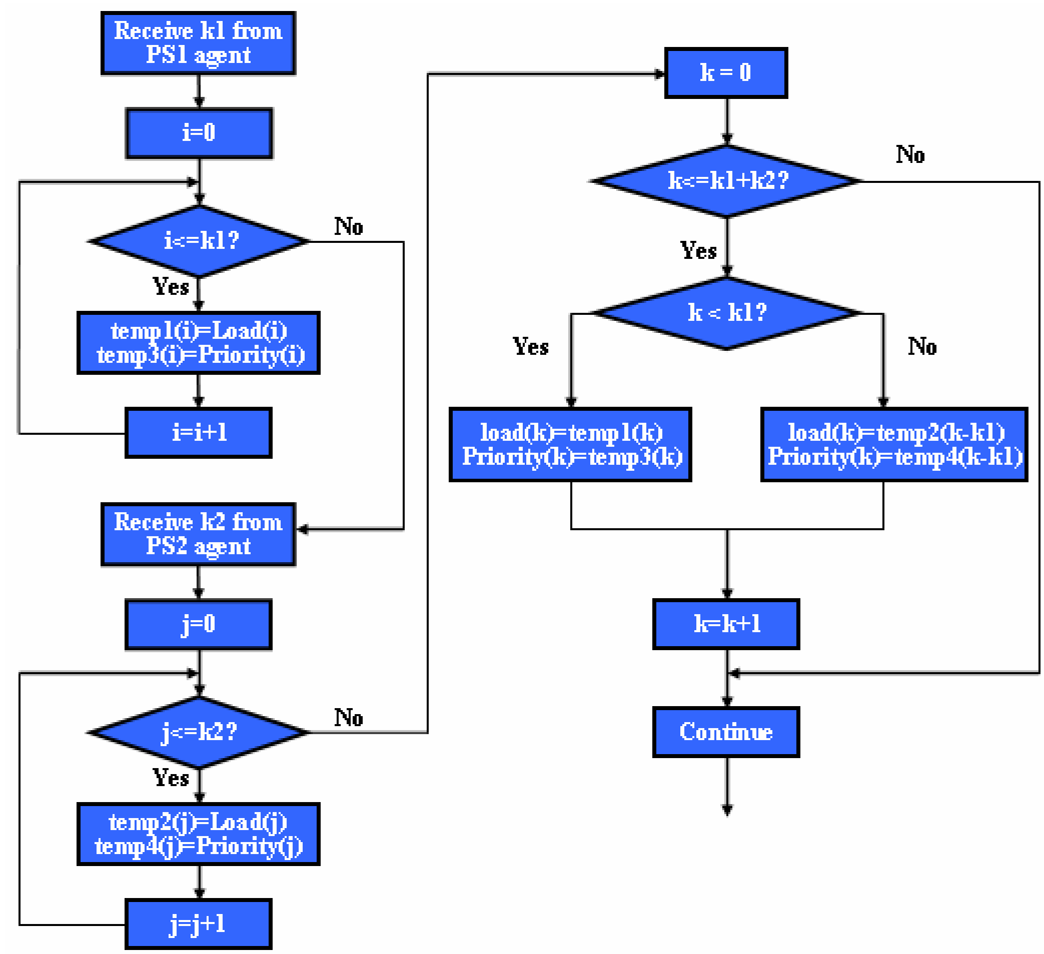


Figure 6.13 Algorithm of Receiving Signals for the Distransformer Agent Class in the Distribution MAS

The distransformer agent will send signals, including its fault flag, the number of loads, load magnitudes, and load priority levels to its upstream neighboring agent. The flow chart of the distransformer agent class is almost the same as that of PS1 agent class, except for algorithm for receiving signals.

6.3 Simulation

The MAS for reconfiguration of the AC-DC zonal distribution system consists of twenty-three agents, which are instances of ten different abstract agent classes. These agents and agent classes are summarized in Table 6.2.

Table 6.2 Distribution Agent Summary

Agent Name	**Agent Class**	**Inputs (in order)**
LB MD CPL	LoadAgent	Upstream agent name LoadAgent working condition Load magnitude Load priority level
LBBus SSIM MDBus MC0 CPLBus BC	intermediateAgent	Upstream agent name Downstream agent name intermediateAgent working condition
SSIMBus MCBus BCBus	converterBusAgent	Upstream agent1 name Upstream agent1 connection flag Upstream agent2 name Upstream agent2 connection flag Downstream agent name converterBusAgent working condition
SSCM1-6	SSCM14Agent SSCM25Agent SSCM36Agent	Upstream agent name Downstream agent name SSCM14/25/36Agent working condition SSCM14/25/36agent connection flag
PortBus StarboardBus	PSbusAgent	Upstream agent name Downstream agent1 name Downstream agent2 name Downstream agent3 name PSbusAgent working condition
PS1 PS2	PS1Agent PS2Agent	Upstream agent name Downstream agent name PS1/2Agent working condition
DTransformer	distransformerAgent	Upstream agent name Downstream agent1 name Downstream agent2 name distransformerAgent working condition

6.3.1 Test Case 2

This test case is designed to test distribution zones. When a fault happens in a distribution zone, a fault signal will be sent to a converterBus agent in that distribution zone along upstream direction. The comparison of initial condition and post-reconfiguration situation is illustrated in Figure 6.14. Initial conditions of three loads are summarized in Table 6.3, and illustrated in Figure 6.15 when simulation running.

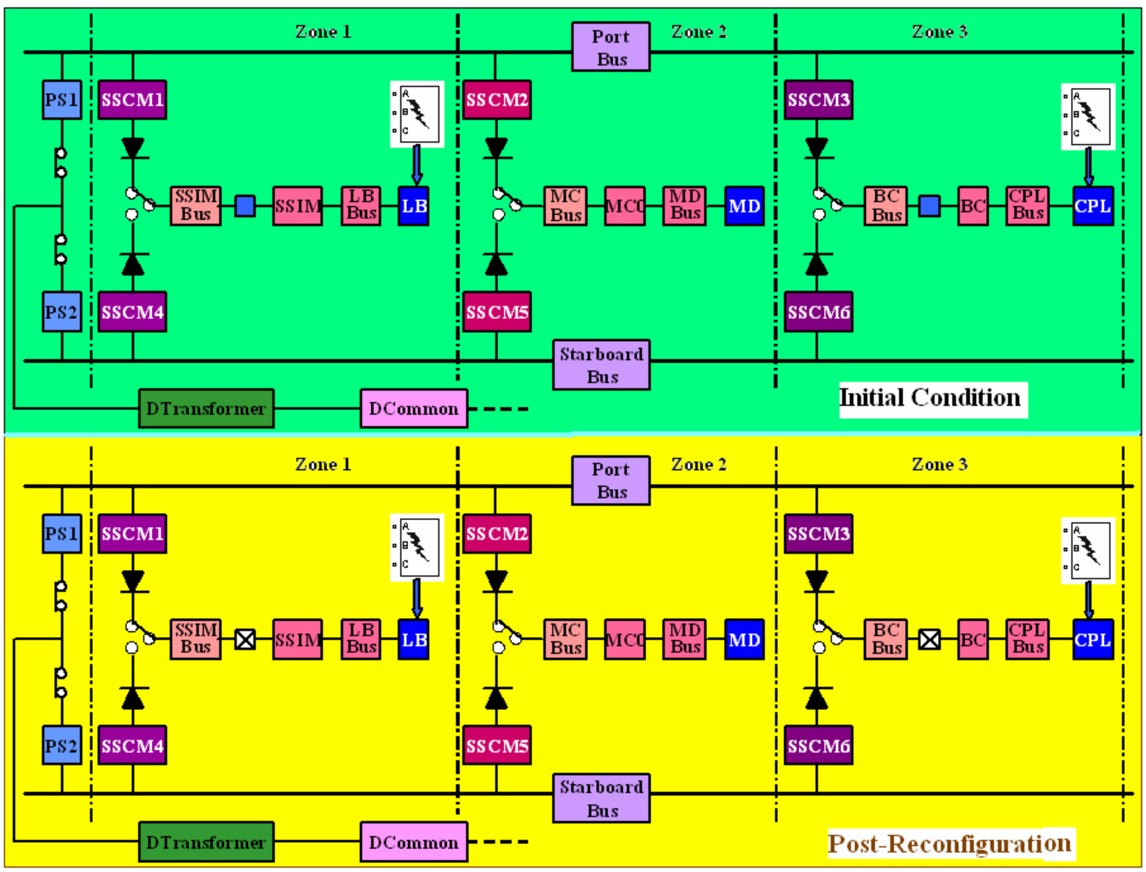


Figure 6.14 Comparison of Initial and Post-Reconfiguration Conditions

Table 6.3 Loads Initial Condition

Load	Initial Condition
Load Bank	Load = 2 MW Load priority = 1 working condition = Fault
Motor Drive	Load = 3 MW Load priority = 3 working condition = Normal
Constant Power Load	Load = 4 MW Load priority = 2 working condition = Fault

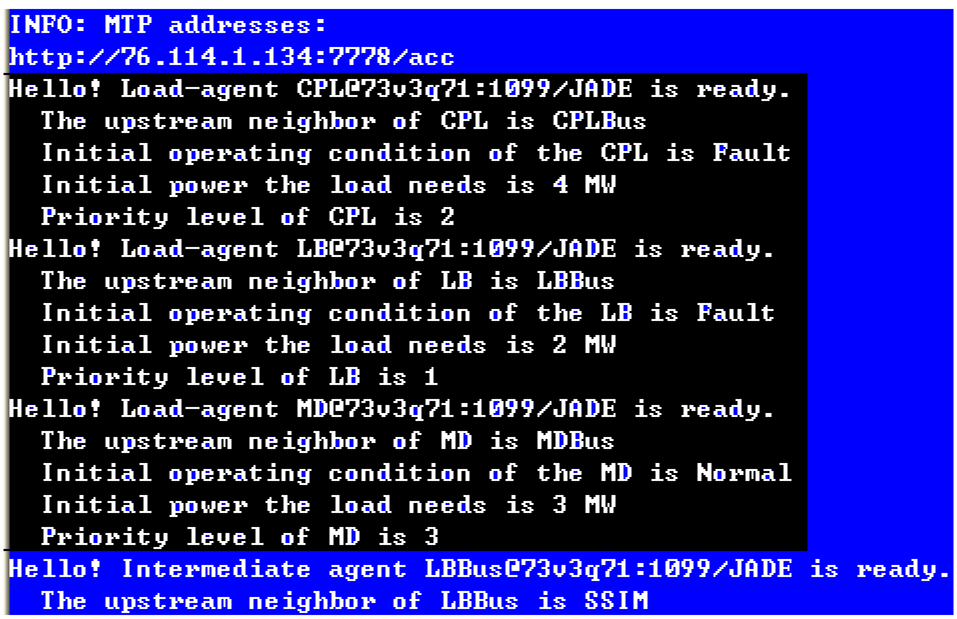


Figure 6.15 Loads Initial Condition in the Distribution MAS Simulation

From Figure 6.14 it can be shown that LB and CPL are in fault condition and MD is in normal condition. After simulation running and agent communication, the BCBus agent catches the fault flag that originally came from CPL, and so does the SSIMBus agent from LB, as illustrated in Figure 6.16.

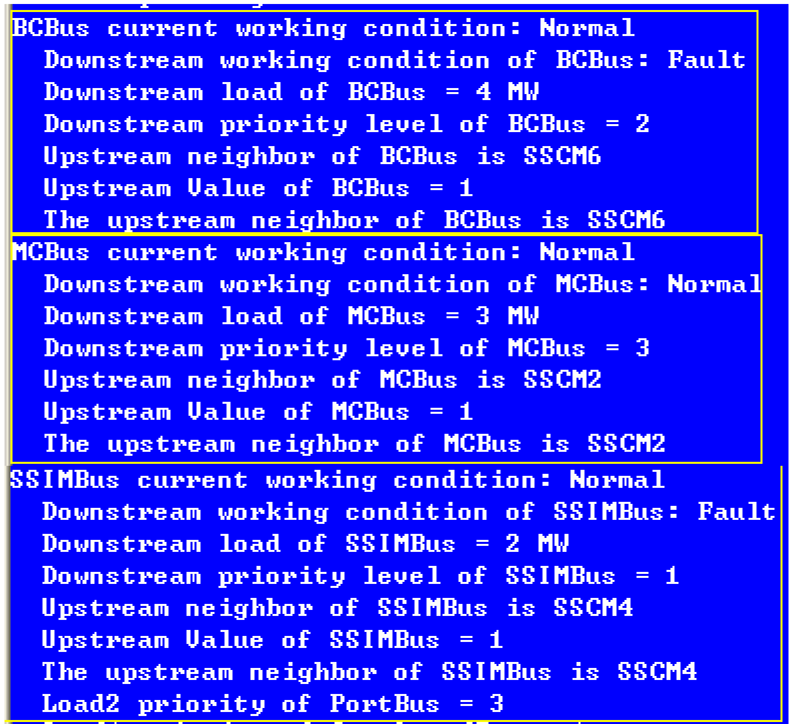


Figure 6.16 ConverterBus Agents Catch Fault Flags in Test Case 2

6.3.2 Test Case 3

The objective of this test case is to test switching function of the converterBus agent. The initial conditions of three converterBus agents and their upstream neighboring agents are listed in Table 6.4. The comparison of initial condition and post-reconfiguration for the distribution system is illustrated in Figure 6.17. The post-reconfiguration situation is also illustrated in Figure 6.18 as simulation running.

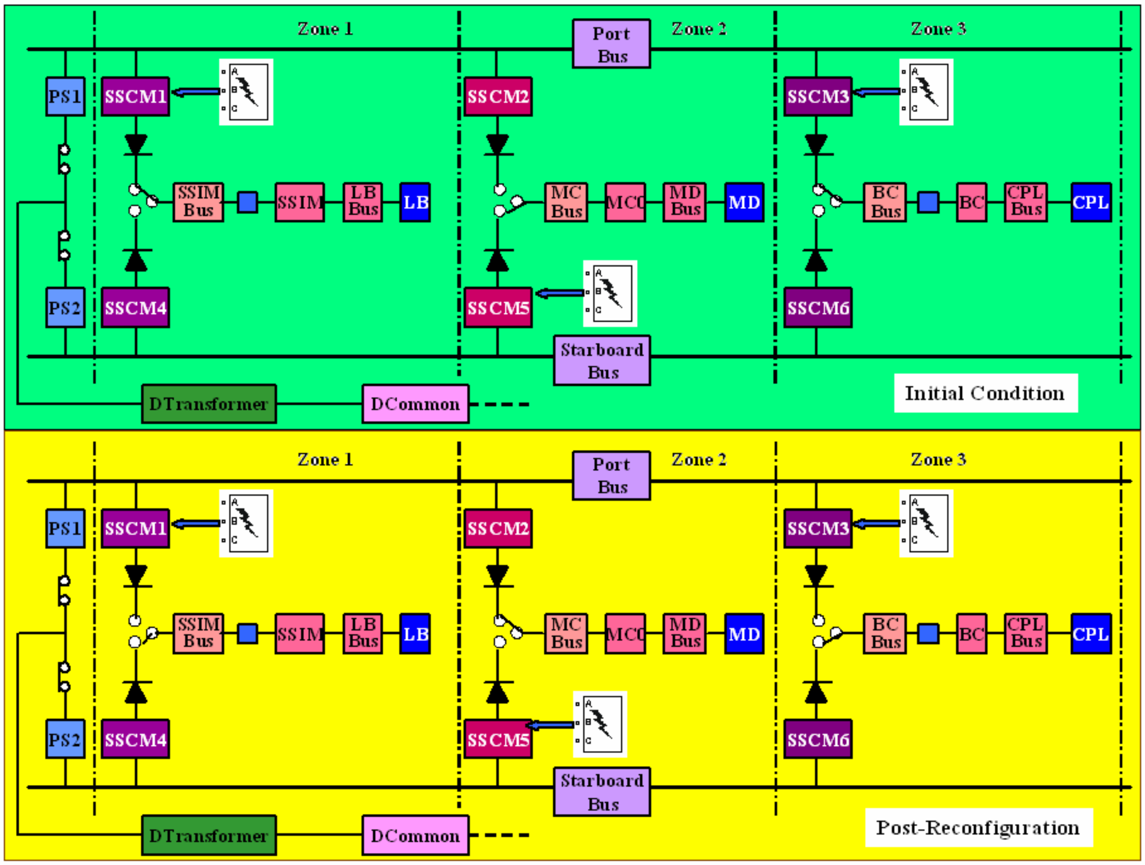


Figure 6.17 Comparison of Initial and Post-Reconfiguration Conditions in Test Case 3

Table 6.4 ConverterBus Agent Initial Condition

ConverterBus Agent	Upstream Neighboring Agent	Initial Connection
SSIMBus	SSCM1 – Fault SSCM4 – Normal	Connected Disconnected
MCBus	SSCM2 – Normal SSCM5 – Fault	Disconnected Connected
BCBus	SSCM3 – Fault SSCM6 – Normal	Connected Disconnected

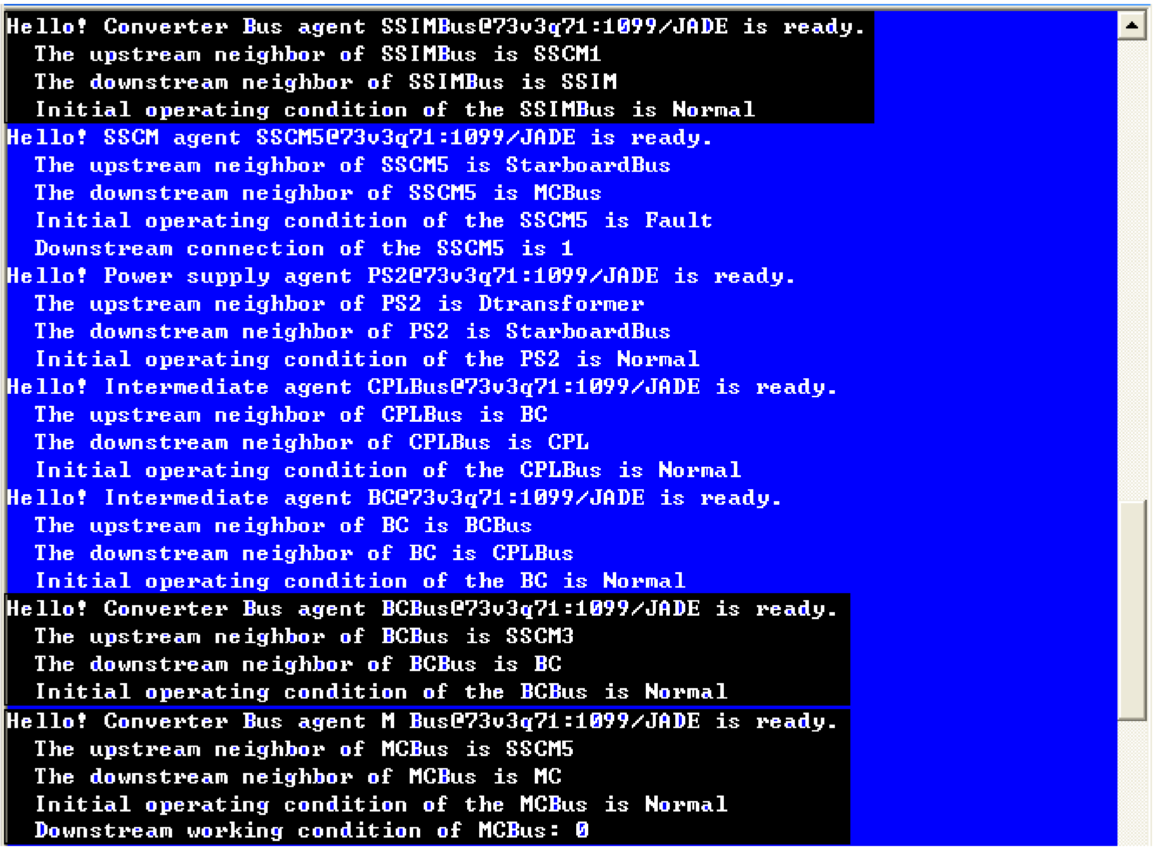


Figure 6.18 ConverterBus Agent Initial Condition in Test Case 3

Because the SSCM1 is in fault condition, the SSIMBus agent automatically switches to SSCM4 after communicating with its upstream neighboring agents. Similarly, the MCBus agent switches to SSCM2, and BCBus to SSCM6, as illustrated in Figure 6.19.

```
  Dtransformer k2= 2
PS1: load[0] = 3 MW  priority[0] = 3
BCBus current working condition: Normal
  Downstream working condition of BCBus: Fault
  Downstream load of BCBus = 4 MW
  Downstream priority level of BCBus = 2
  Upstream neighbor of BCBus is SSCM6
  Upstream Value of BCBus = 1
  The upstream neighbor of BCBus is SSCM6
AG1 current working condition: Normal
DCommon connected to: AG1
  AG1 working condition of DCommon: Normal
  AG1 power of DCommon =  2
MD current working condition: Normal
  Load3 priority of StarboardBus = 2
MCBus current working condition: Normal
  Downstream working condition of MCBus: Normal
  Downstream load of MCBus = 3 MW
  Downstream priority level of MCBus = 3
  Upstream neighbor of MCBus is SSCM2
  Upstream Value of MCBus = 1
  The upstream neighbor of MCBus is SSCM2
SSIMBus current working condition: Normal
  Downstream working condition of SSIMBus: Fault
  Downstream load of SSIMBus = 2 MW
  Downstream priority level of SSIMBus = 1
  Upstream neighbor of SSIMBus is SSCM4
  Upstream Value of SSIMBus = 1
  The upstream neighbor of SSIMBus is SSCM4
```

Figure 6.19 ConverterBus Agents Automatically Switch in Test Case 3

6.3.3 Test Case 4

This test case is designed to test reconfiguration algorithm and control function of whole distribution MAS. Initial conditions of three distribution loads are the same as in test case 2 and listed in Table 6.3. Initial conditions of SSCM1-6 are the same as in test case 3 and listed in Table 6.4. Thus test case 4 is built upon test case 2 and 3. As the test case runs, the distransformer agent should receive load information, such as load fault flag, load magnitude, and load priority level, from the distribution agents through the interface and control agents, as illustrated in Figure 6.20.

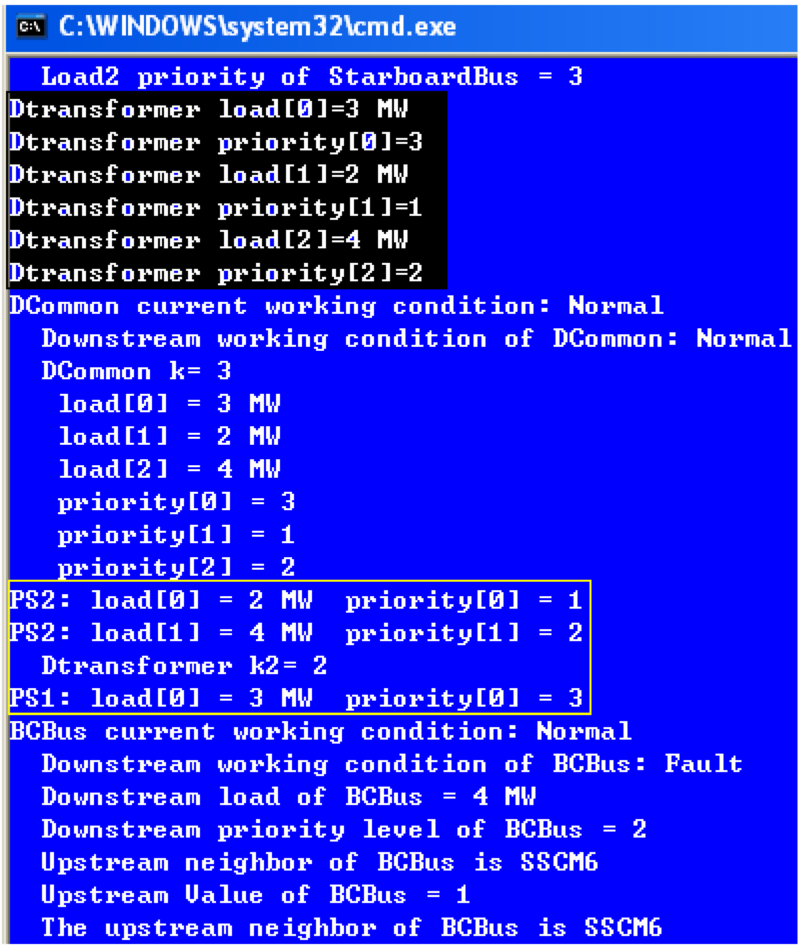


Figure 6.20 Distransformer Agent Gets Load Information

6.4 Summary

This chapter discusses design and implementation of distribution MAS, design different test cases for different reconfiguration functions, and simulates the distribution MAS. Next chapter will present design and implementation of generation MAS.

CHAPTER VII

MAS DESIGN AND SIMULATION FOR THE ELECTRIC POWER GENERATION SYSTEM

The electric power generation system will provide power for the propulsion and distribution loads. During battles, pulsed weapons can "borrow" the main part of power that is used for propulsion system when warships are steaming at full speed. The electric power generation system consists of two main generators, two auxiliary generators, and a common bus. Design of the common bus agent is difficult, although the topology of the electric power generation system is simple. Like chapters V and VI, this chapter will discuss topology, reconfiguration algorithm, and agent design of the electric power generation MAS.

7.1 Topology and Algorithms of the Electric Power Generation MAS

For every component in the electric generation system, such as main generator, auxiliary generator, and common bus, an agent and a circuit breaker are assigned to associate with the component. The electric power generation system and corresponding MAS topology are illustrated in Figure 7.1 and 7.2, respectively.

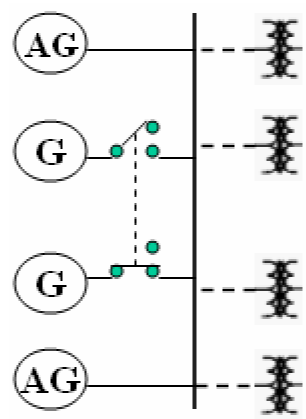


Figure 7.1 Electric Power Generation System

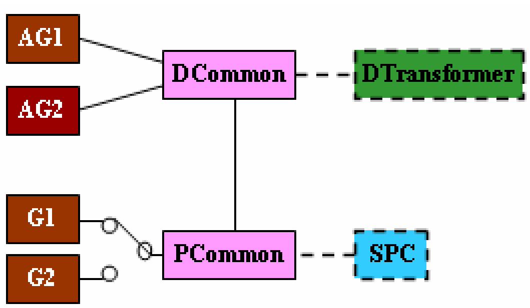


Figure 7.2 Topology of the Power Generation MAS

It should be mentioned that, the common bus in the power generation system is divided into three different agents, i.e., propulsion common bus agent, distribution common bus agent, and secondary propulsion bus. The reason is to increase survivability through distributing control functionality into different agents and decrease design complexity.

The reconfiguration algorithm for power generation MAS is more straightforward. The main generator G1 and G2 are mutually exclusive. At a time only one main generator is connected to the propulsion common bus. If a main generator has a problem, the corresponding agent will send a fault signal to the propulsion common bus agent, and the propulsion common bus agent automatically switches to another main generator. If an auxiliary generator has a problem, the corresponding agent will send a fault signal to the distribution common bus agent, and cut off this auxiliary generator from the distribution common bus. Another objective of the propulsion common bus agent is to receive information on power supplies, propulsion load, and distribution loads. Based on this information, the propulsion common bus will adjust motor output. Similarly, the distribution common bus will adjust distribution loads through information exchange with the propulsion common bus agent.

7.2 Design of Agents in the Power Generation MAS

The power generation MAS consists of six different agents. They are G1 agent, G2 agent, AG1 agent, AG2 agent, PCommon agent, and DCommon agent. Meanings of these agent names and their abstract agent classes are summarized in Table 7.1.

Table 7.1 Meanings of Power Generation Agents

Agent Name	Meaning	Abstract Agent Class
G1	Generator 1	MaingenAgent
G2	Generator 2	
AG1	Auxiliary Generator 1	agen1Agent
AG2	Auxiliary Generator 2	agen2Agent
PCommon	Propulsion Common Bus	PCommonBusAgent
DCommon	Distribution Common Bus	CommonBusAgent

AG1 and AG2 agents are very similar, except for different communication message performatives. Following sections will discuss design of the main generator agent, auxiliary generator agent, and common bus agent.

7.2.1 Design of Main Generator Agent

The main generator agent, named MaingenAgent, is designed to have four inputs, i.e., the name of its downstream neighboring agent, its working condition (fault flag), connection flag, and the power it can provide. To ensure that two main generators are mutually exclusive, one important communication function of the main generator agent is to receive a control signal from its downstream neighboring agent – propulsion common bus agent. This control signal is connName, the name of the main generator that connects to the propulsion common bus. A logic variable, connFlag, can be defined as

$$connFlag \quad \begin{cases} 1 & if\ connName\ is\ the\ name\ of\ this\ main\ generator \\ 0 & if\ connName\ is\ the\ name\ of\ another\ main\ generator \end{cases} \qquad (7.1)$$

If the connFlag is equal to one, the main generator connects to the propulsion common bus agent and needs to send its fault flag and power magnitude to its downstream neighboring agent – propulsion common bus agent. Thus the main generator agent needs communication in downstream side. It should receive a message from and may send messages to its downstream neighboring agent. The flow chart of the main generator agent is illustrated in Figure 7.3.

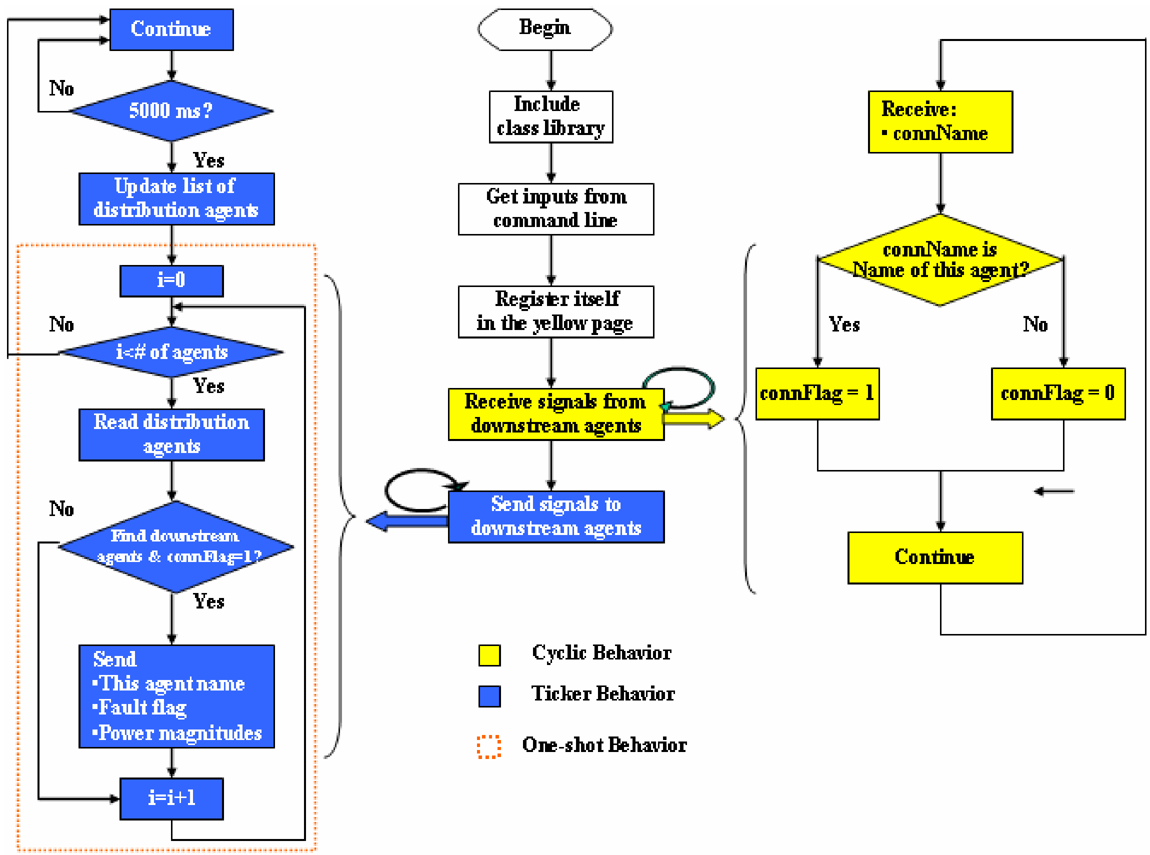


Figure 7.3 Flow Chart of the Main Generator Agent in the Generation MAS

7.2.2 Design of Auxiliary Generator Agent

Two auxiliary generator agents, agen1Agent and agen2Agent, are almost the same, except of different communication message performatives. In this section only agen1Agent is discussed. The auxiliary generator agent, named agen1Agent, is designed to have three inputs, i.e., the name of its downstream neighboring agent, its working condition (fault flag), and the power it can provide. The auxiliary generator bus only sends its information, such as the name, fault flag, and power magnitude, to its downstream neighboring agent – distribution common bus agent. Its flow chart is similar to that of a load agent in distribution MAS (section 6.2.1.1) and is not shown here. The only difference between agen1Agent and agen2Agent is that, the agen1Agent sends messages with INFORM performative and agen2Agent with PROPOSE performative.

7.2.3 Design of Common Bus Agent

The propulsion common bus agent, named PCommonBusAgent, is more complicated, compared with other agents in the three MAS. It is designed to have seven inputs, including the name of generator1, connection flag of generator1, the name of generator2, connection flag of generator2, the name of distribution common bus agent, the name of downstream neighboring agent, and its working condition. It should have three-directional communication functions, sending messages to and receiving messages from main generators, distribution common bus, and temp bus. Based on the main generator's information, the propulsion common bus agent can implement its automatic switching function. When generator1 has a fault, for instance, a fault message will be sent to the propulsion common bus agent. After receiving the fault flag of generator1, the propulsion common bus agent automatically switches to the generator2. Based on the information on power and load magnitudes and priority level, the propulsion common bus agent will decide whether to change propulsion load level, or shed distribution loads. Communication functions and automatic switch control are illustrated in Figure 7.4.

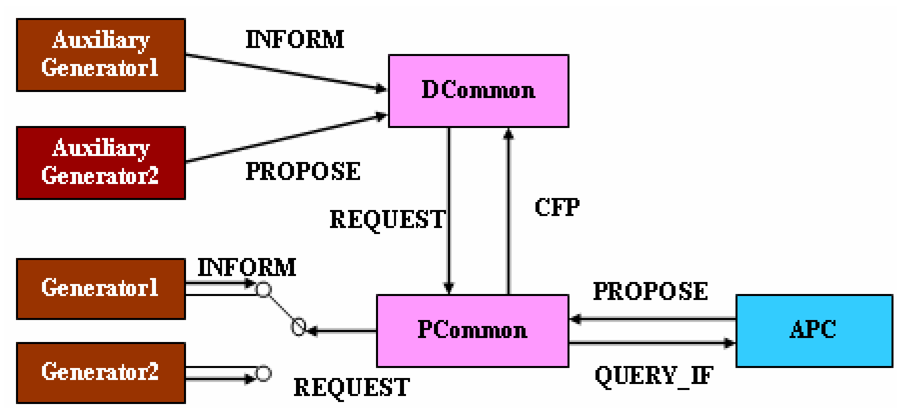


Figure 7.4 Communication Message Performatives in the Power Generation MAS

The flow chart of the propulsion common bus agent is shown in Figure 7.5.

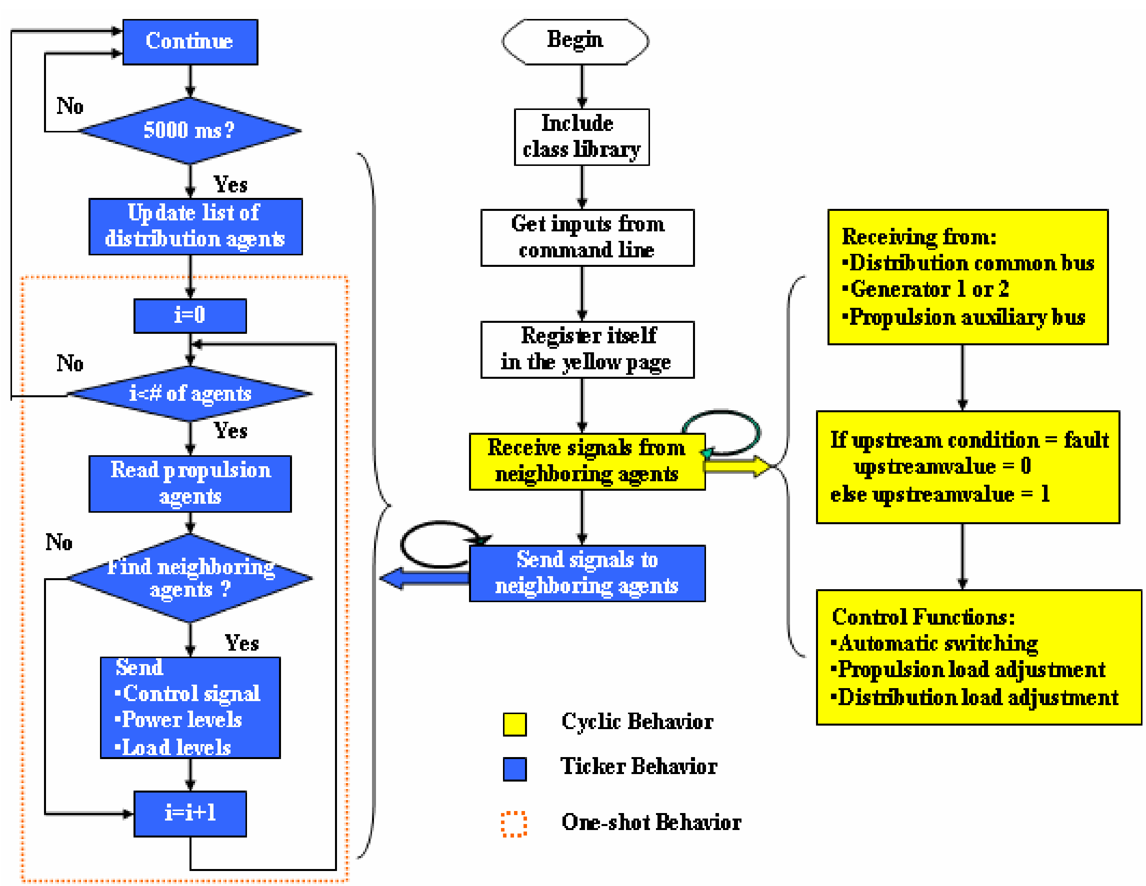


Figure 7.5 Flow Chart of the Propulsion Common Bus Agent

7.3 Simulation

The MAS for reconfiguration of the electric power generation system consists of six agents, which are instances of five different abstract agent classes. These agents and agent classes are summarized in Table 7.2.

Table 7.2 Summary of Power Generation Agents

Agent Name	Agent Class	Inputs (in order)
G1 G2	maingenAgent	Downstream agent name maingenAgent working condition Downstream connection flag Generator power level
AG1	agen1Agent	Downstream agent name agen1Agent working condition Auxiliary generator power level
AG2	agen2Agent	Downstream agent name agen2Agent working condition Auxiliary generator power level
PCommon	PCommonBusAgent	Generator1 name Generator1 connection flag Generator2 name Generator2 connection flag Distribution common bus name Downstream agent name PCommon working condition
DCommon	CommonBusAgent	Downstream name Propulsion common bus name Distribution common bus working condition

7.3.1 Test Case 5

This test case is designed to test switching functionality of the PCommonBusAgent. The initial conditions of two main generator agents, G1 and G2, and their connection flags are listed in Table 7.3, and illustrated in Figure 7.6 as the simulation runs. We can see that, G1 is initially connected to the propulsion common bus and in fault condition. G2 is in normal condition and does not connect to the common bus.

Table 7.3 Initial Conditions of Main Generators G1 and G2

Main generator agent	Upstream Neighboring Agent	Initial Connection
G1	PCommon	Connected Fault
G2	PCommon	Disconnected Normal

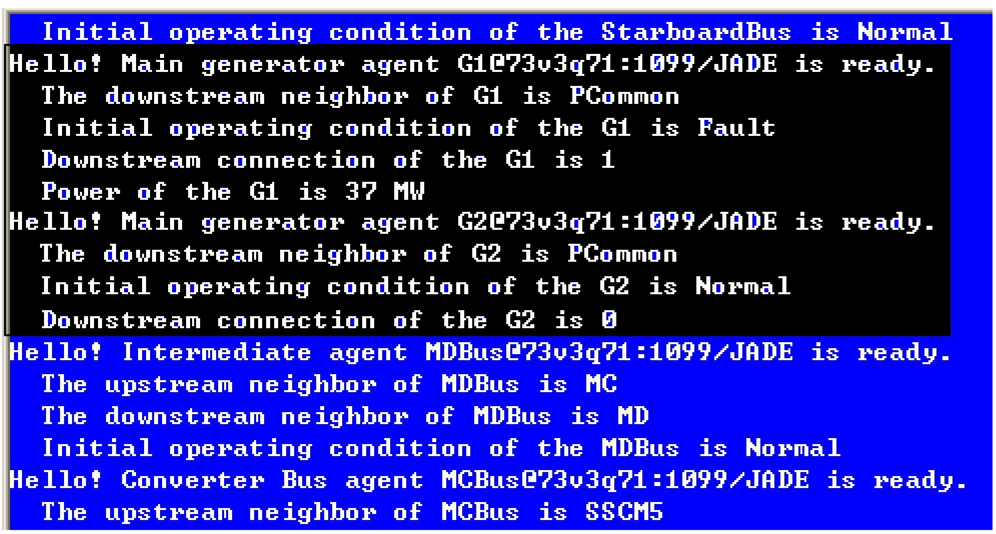


Figure 7.6 Main Generators Initial Condition in the Generation MAS Simulation

The initial condition and post-reconfiguration situation are illustrated in Figure 7.7. After simulation running and agent communication, the propulsion common bus agent automatically switches to the main generator G2, as illustrated in Figure 7.7 and

7.8.

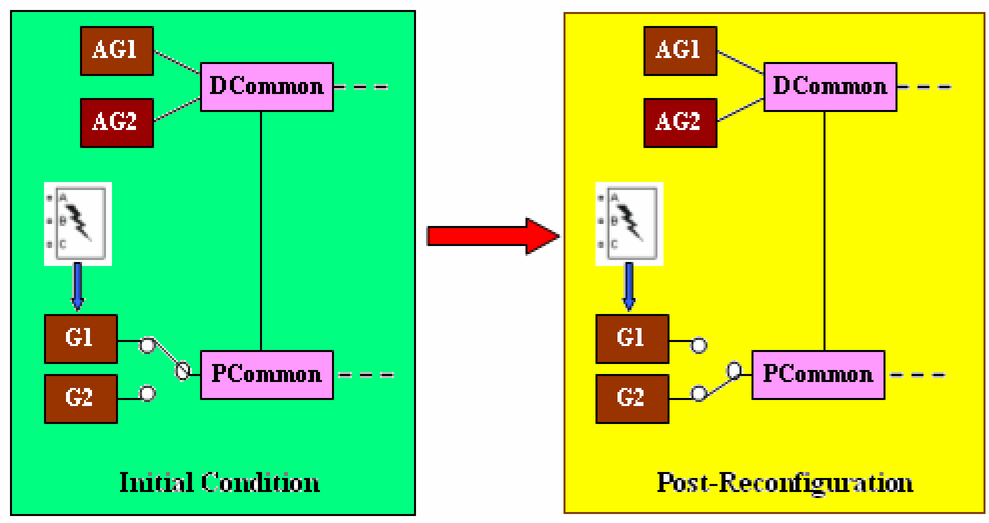


Figure 7.7 Initial Condition and Post-Reconfiguration Situation in Test Case 5

```
PortBus current working condition: Normal
  Load1 priority of PortBus = 1
Transformer1 switch status: On
G1 current working condition: Fault
  Connection name of G1: G2  connection: 0
G2 current working condition: Normal
  Connection name of G2: G2  connection: 1
  Propulsion common bus of DCommon: PCommon
  Prop common bus fault flag of DCommon: Normal
PCommon current working condition: Normal
  Distribution common bus of PCommon: DCommon
  Distribution common bus working condition of PCommon: Normal
  Generator connected to PCommon: G2
  G2 working condition of PCommon: Normal
  Generator power of PCommon = 37 MW
  AG2 power of DCommon =  8
  Load2 priority of PortBus = 3
SSIMBus current working condition: Normal
```

Figure 7.8 Propulsion Common Bus Agent Automatically Switch in Test Case 5

7.3.2 Test Case 6

This test case is designed to test distribution common bus – DCommon. First, the DCommon agent communicates with distribution transformer agent, to receive information on distribution loads. Second, the DCommon agent receives power magnitudes from the auxiliary generator agents. Finally, the DCommon agent exchanges information with the propulsion common bus agent, to gather information on power of the main generator and magnitude and priority level of the propulsion load. The information on distribution load, propulsion load, main generator power, and auxiliary generator powers are listed in Table 7.4. Here the smaller is the priority level value, the higher the load priority. Simulation results of the test case are illustrated in Figure 7.9, which shows the functionality of the distribution common bus agent.

Table 7.4 Load and Generator Information

Category	Magnitude of Load/Power	Load Priority Level
Distribution Load	LB=2MW	1
	MD=3WM	3
	CPL=4WM	2
Auxiliary Generator	AG1=2MW	
	AG2=8MW	
Main Generator	G1=36MW	
	G2=36MW	
Propulsion Load	Motor=36MW	4

```
DCommon current working condition: Normal
  Downstream working condition of DCommon: Normal
  DCommon k= 3
   load[0] = 3 MW
   load[1] = 2 MW
   load[2] = 4 MW
   priority[0] = 3
   priority[1] = 1
   priority[2] = 2
DCommon connected to: AG2
  AG2 working condition of DCommon: Normal
  AG2 power of DCommon =  8
DCommon connected to: AG1
  AG1 working condition of DCommon: Normal
  AG1 power of DCommon =  2
G1 current working condition: Fault
  Connection name of G1: G2  connection: 0
G2 current working condition: Normal
  Connection name of G2: G2  connection: 1
  Propulsion commom bus of DCommon: PCommon
  Prop common bus fault flag of DCommon: Normal
  Prop power of DCommon =  ****36 MW
  Prop load of DCommon =  ****36 MW
  Propload priority of DCommon =  ****4
  Downstream working condition of MCBus: Normal
```

Figure 7.9 DCommon agent receiving information in Test Case 6

7.4 Summary

This chapter presents the topology and reconfiguration algorithms of the generation MAS, discusses design and implementation of different agent classes, including the main generator and auxiliary generator agents and common bus agent, designs a test case to verify reconfiguration functionality, and simulates the generation MAS in the JAVA/JADE platform. Next chapter will discuss the entire MAS for reconfiguration of the IPS.

CHAPTER VIII

SIMULATION OF MAS-BASED RECONFIGURATION FOR AN IPS

In chapters V, VI, and VII, three MASs have been designed and simulated separately for reconfiguration of an electric propulsion system, an AC-DC zonal distribution system, and an electric power generation system. As an IPS integrates electric propulsion system, AC-DC zonal distribution system, and electric power generation system together, this chapter combines these three MASs, and simulates MAS-based reconfiguration of the whole IPS. The first section introduces the topology of the whole IPS and presents a summary of all agents and abstract agent classes. The second section describes a test case in which MAS-based reconfiguration for an IPS is simulated, and illustrates simulation results.

8.1 Topology of the Entire MAS

As it has been mentioned in previous chapters, to increase survivability with less than 100% redundancy and meet the Navy's demand, the IPS with AC-DC zonal distribution system is preferred and was designed in the NCS DCDT. To decrease the possibility of power-loss zones, a new IPS architecture with modified zonal distribution topology has been designed in Chapter IV, as illustrated in Figure 8.1. For every component in the IPS, such as generator, transformer, common bus, and load, an agent is assigned to associate with the component. A circuit breaker is assigned to every key control agent, such as converter bus agent, distribution transformer agent, and common bus agent. Thus the whole MAS and the IPS have the same topology. Figure 8.2 shows the topology of the whole MAS.

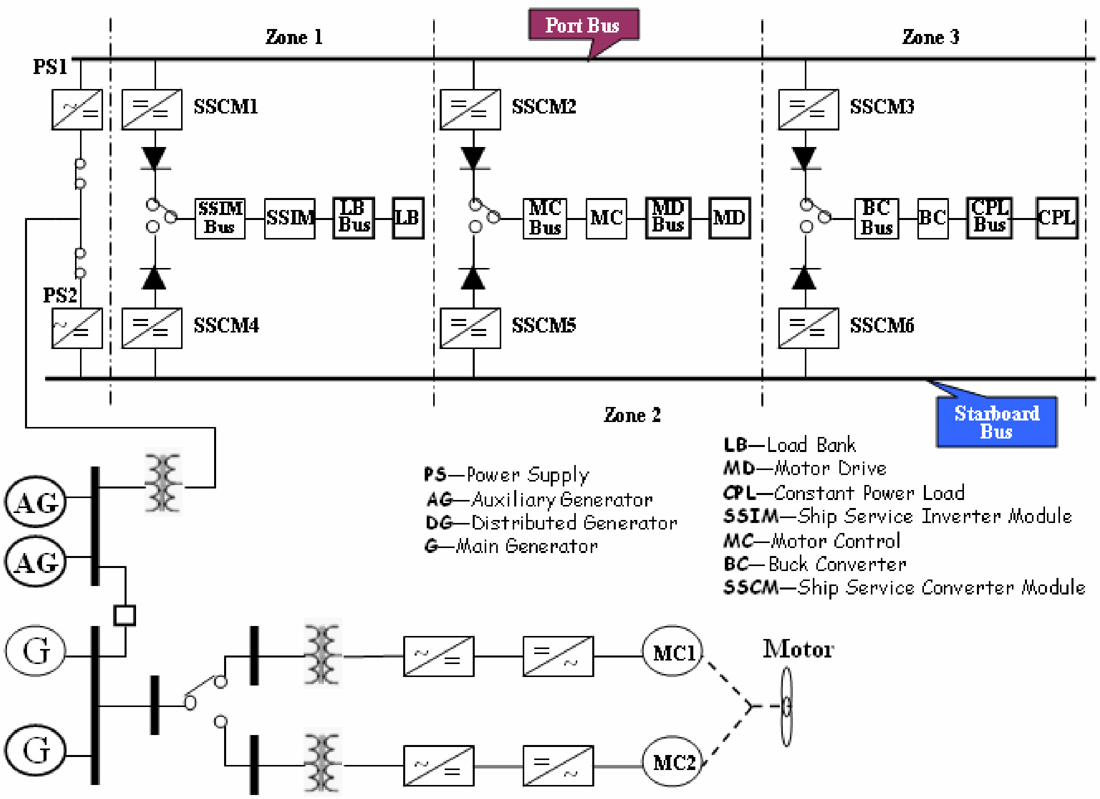


Figure 8.1 Topology of an IPS

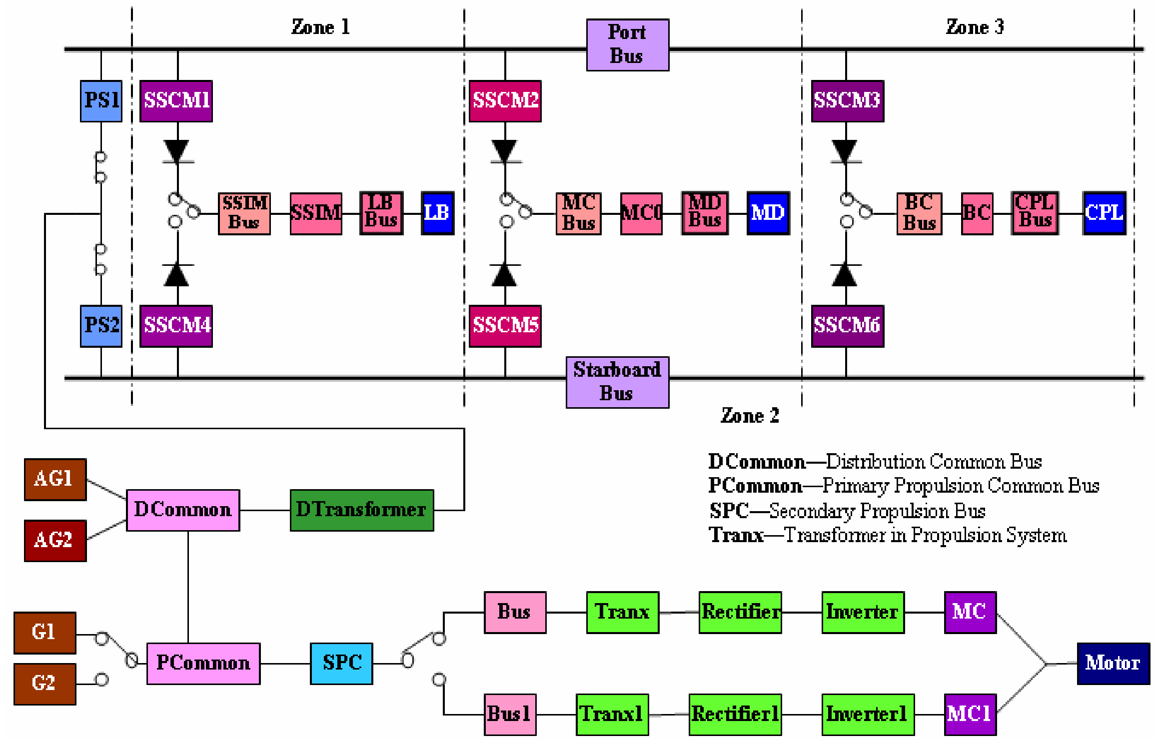


Figure 8.2 Topology of the whole MAS

It can be seen that the combined MAS consists of three sub-MASs, named propulsion MAS, generation MAS, and distribution MAS, including forty-one different agents. These agents are instances of nineteen different abstract agent classes. Table 8.1 summarizes these sub-MASs, agents, and agent classes.

Table 8.1 Summary of Sub-MAS, Agent, and Agent Class

Sub-MAS	Agent	Abstract Agent Class
Distribution MAS	LB MD CPL	LoadAgent
	LBBus SSIM MDBus MC0 CPLBus BC	intermediateAgent
	SSIMBus MCBus BCBus	converterBusAgent
	SSCM1 SSCM4	SSCM14Agent
	SSCM2 SSCM5	SSCM25Agent
	SSCM3 SSCM6	SSCM36Agent
	PortBus StarboardBus	PSbusAgent
	PS1	PS1Agent
	PS2	PS2Agent
	DTransformer	distransformerAgent
Generation MAS	G1	MaingenAgent
	G2	
	AG1	agen1Agent
	AG2	agen2Agent
	PCommon	PCommonBusAgent
	DCommon	CommonBusAgent
Propulsion MAS	MC MC1 Inverter Inverter1 Rectifier Rectifier1 Transformer Transformer1	propintermediateAgent
	Bus	propbusAgent
	Bus1	
	SPC	tempBusAgent
	Motor	PropLoadAgent

8.2 Simulation of MAS-Based Reconfiguration of an IPS

8.2.1 Introduction to the Simulation Platform

All abstract agent classes have been programmed with JAVA/JADE software. The simulation environment had been established on a Dell PC in Electrical and Computer Engineering Department at Mississippi State University in 2006. The simulation environment was based on JAVA jdk1.5.0_08 and JADE 3.4.1 platform. Selecting JADE as the agent programming language was based on consideration that the JADE package is designed to facilitate the development of multi-agent systems.

Another issue related to agent coding, debugging, and running is setting up environment variables. To ensure that whenever JAVA and JADE programs are compiled or executed, the correct libraries are used, the environment variables, PATH and CLASSPATH, have to be set up. These set up commands depend on operating systems, and the directories in which jdk and JADE are installed.

8.2.2 Design and Simulation of Test Case 7

This test case was designed to test functions of the whole MAS, which includes three sub-MASs with forty-one different agents. Initial conditions of all agents are the same as those in Test Case 1-6, and are listed in Table 8.2 and illustrated in Figure 8.3. Figure 8.4 shows the post-reconfiguration situation of the entire MAS.

Table 8.2 Summary of Agent Initial Condition

Sub-MAS	Agent	Initial Condition
Distribution MAS	LB MD CPL	Load 2MW, priority 1, Fault Load 3MW, priority 3 Normal Load 4MW, priority 2, Fault
	LBBus SSIM MDBus MC0 CPLBus BC	Normal
	SSIMBus MCBus BCBus	Normal
	SSCM1 SSCM4	Fault, connected Normal, disconnected
	SSCM2 SSCM5	Normal, disconnected Fault, connected
	SSCM3 SSCM6	Fault, connected Normal, disconnected
	PortBus StarboardBus	Normal
	PS1	Normal
	PS2	Normal
	DTransformer	Normal
Generation MAS	G1	Power 36MW, fault, connected
	G2	Power 36MW, normal, disconnected
	AG1	Power 2MW, normal, connected
	AG2	Power 8MW, normal, connected
	PCommon	Normal
	DCommon	Normal
Propulsion MAS	MC MC1 Inverter Inverter1 Rectifier Rectifier1 Transformer Transformer1	Normal
	Bus	Normal, connected
	Bus1	Normal, disconnected
	SPC	Normal
	Motor	Load 36MW, priority 4, normal

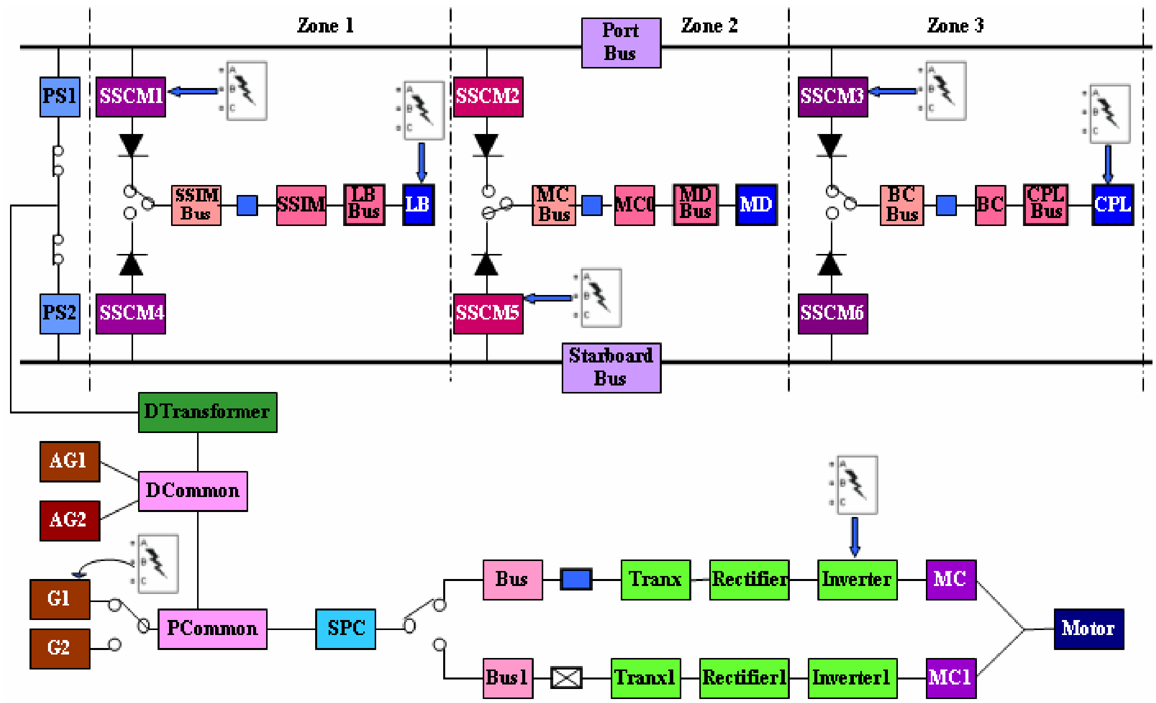


Figure 8.3 Initial Condition of the Entire MAS

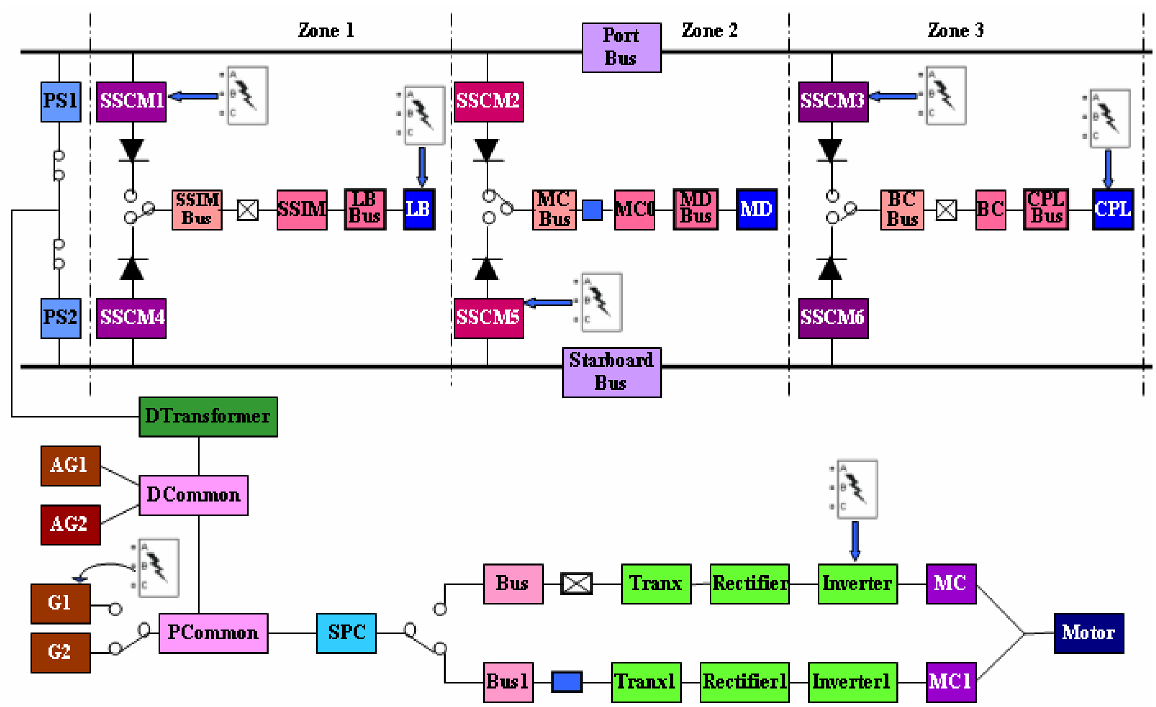


Figure 8.4 Post-Reconfiguration of the Entire MAS

In simulation, the forty-one agents receive their start-up arguments specified on the command line as their inputs, as illustrated in dark area of Figure 8.5.

```
C:\Program Files\MAS_distribution>java jade.Boot LB:LoadAgent(LBBus Fault 2 1) L
BBus:intermediateAgent(SSIM LB Normal) SSIM:intermediateAgent(SSIMBus LBBus Norm
al) SSIMBus:converterBusAgent(SSCM1 1 SSCM4 0 SSIM Normal) SSCM1:SSCM14Agent(Por
tBus SSIMBus Fault 1) SSCM4:SSCM14Agent(StarboardBus SSIMBus Normal 0) MD:LoadAg
ent1(MDBus Normal 3 3) MDBus:intermediateAgent(MC0 MD Normal) MC0:intermediateAg
ent(MCBus MDBus Normal) MCBus:converterBusAgent(SSCM2 0 SSCM5 1 MC0 Normal) SSCM
2:SSCM25Agent(PortBus MCBus Normal 0) SSCM5:SSCM25Agent(StarboardBus MCBus Fault
 1) CPL:LoadAgent1(CPLBus Fault 4 2) CPLBus:intermediateAgent(BC CPL Normal) BC:
intermediateAgent(BCBus CPLBus Normal) BCBus:converterBusAgent(SSCM3 1 SSCM6 0 B
C Normal) SSCM3:SSCM36Agent(PortBus BCBus Fault 1) SSCM6:SSCM36Agent(StarboardBu
s BCBus Normal 0) PortBus:PSbusAgent(PS1 SSCM1 SSCM2 SSCM3 Normal) StarboardBus:
PSbusAgent(PS2 SSCM4 SSCM5 SSCM6 Normal) PS1:PS1Agent(Dtransformer PortBus Norma
l) PS2:PS2Agent(Dtransformer StarboardBus Normal) Dtransformer:distransformerAge
nt(DCommon PS1 PS2 Normal) DCommon:CommonBusAgent(Dtransformer PCommon Normal) A
G1:agen1Agent(DCommon Normal 2) AG2:agen2Agent(DCommon Normal 8) PCommon:PCommon
BusAgent(G1 1 G2 0 DCommon Temp Normal) G1:maingenAgent(PCommon Fault 1 36) G2:m
aingenAgent(PCommon Normal 0 36) Motor:PropLoadAgent(MC MC1 Normal 36 4) MC:prop
intermediateAgent(Inverter Motor Normal) Inverter:propintermediateAgent(Rectifie
r MC Fault) Rectifier:propintermediateAgent(Transformer Inverter Normal) Transfo
rmer:propintermediateAgent(Bus Rectifier Normal) Bus:propbusAgent(Temp Transform
er Normal 1) MC1:propintermediateAgent(Inverter1 Motor Normal) Inverter1:propint
ermediateAgent(Rectifier1 MC1 Normal) Rectifier1:propintermediateAgent(Transform
er1 Inverter1 Normal) Transformer1:propintermediateAgent(Bus1 Rectifier1 Normal)
 Bus1:propbusAgent(Temp Transformer1 Normal 0) Temp:tempBusAgent(Bus 1 Bus1 0 PC
ommon Normal)
Aug 2, 2008 12:43:27 PM jade.core.Runtime beginContainer
INFO: ----------------------------------
    This is JADE 3.6 - revision 6032 of 2008/05/05 14:07:10
    downloaded in Open Source, under LGPL restrictions,
    at http://jade.tilab.com/
----------------------------------------
Aug 2, 2008 12:43:29 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
```

Figure 8.5 Agents in the MAS Get Their Inputs from Command Line

Figure 8.6 is a snapshot of agent registration in the Yellow Pages provided by the DF agent in the main container. It can be seen that total forty-one agents in the MAS are registered in the Yellow Pages, and one agent can find all other active agents at any time.

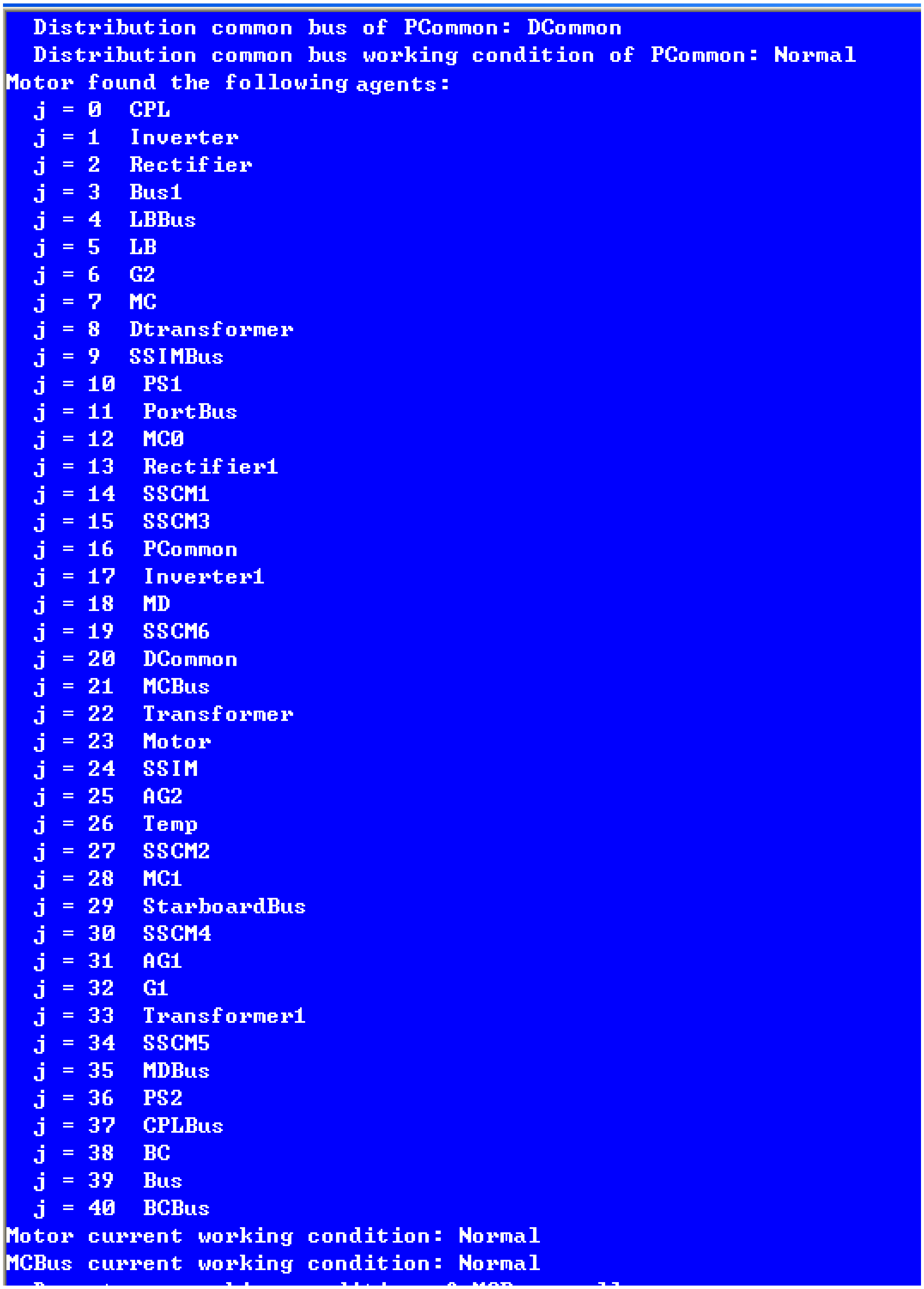
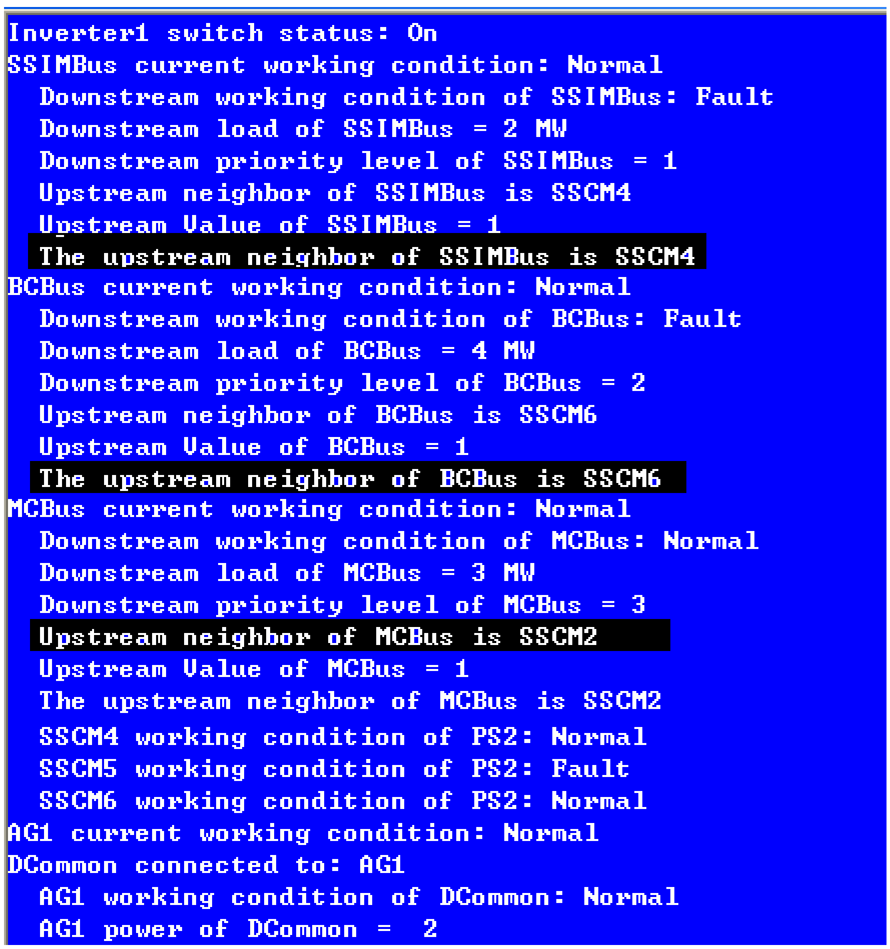


Figure 8.6 Agent Registration in the Yellow Pages

It is assumed that, initially the SSIM Bus, MC Bus, and BC Bus connect to SSCM1, SSCM5, and SSCM3, respectively, and SSCM1, SSCM5, and SSCM3 all are in fault condition. After reconfiguration, the SSIM Bus should switch to SSCM4, and MC Bus to SSCM2, BC Bus to SSCM6. Simulation result, as illustrated in dark area of Figure 8.7 snapshot.



```
Inverter1 switch status: On
SSIMBus current working condition: Normal
  Downstream working condition of SSIMBus: Fault
  Downstream load of SSIMBus = 2 MW
  Downstream priority level of SSIMBus = 1
  Upstream neighbor of SSIMBus is SSCM4
  Upstream Value of SSIMBus = 1
  The upstream neighbor of SSIMBus is SSCM4
BCBus current working condition: Normal
  Downstream working condition of BCBus: Fault
  Downstream load of BCBus = 4 MW
  Downstream priority level of BCBus = 2
  Upstream neighbor of BCBus is SSCM6
  Upstream Value of BCBus = 1
  The upstream neighbor of BCBus is SSCM6
MCBus current working condition: Normal
  Downstream working condition of MCBus: Normal
  Downstream load of MCBus = 3 MW
  Downstream priority level of MCBus = 3
  Upstream neighbor of MCBus is SSCM2
  Upstream Value of MCBus = 1
  The upstream neighbor of MCBus is SSCM2
  SSCM4 working condition of PS2: Normal
  SSCM5 working condition of PS2: Fault
  SSCM6 working condition of PS2: Normal
AG1 current working condition: Normal
DCommon connected to: AG1
  AG1 working condition of DCommon: Normal
  AG1 power of DCommon =  2
```

Figure 8.7 ConverterBusAgent Automatic Switching

Initially the main generator, G1, connects to the propulsion common bus and is in fault condition. After reconfiguration, the propulsion common bus, PCommon, should switch to the main generator G2. Figure 8.8 snapshot shows the switching results, as illustrated in dark area.

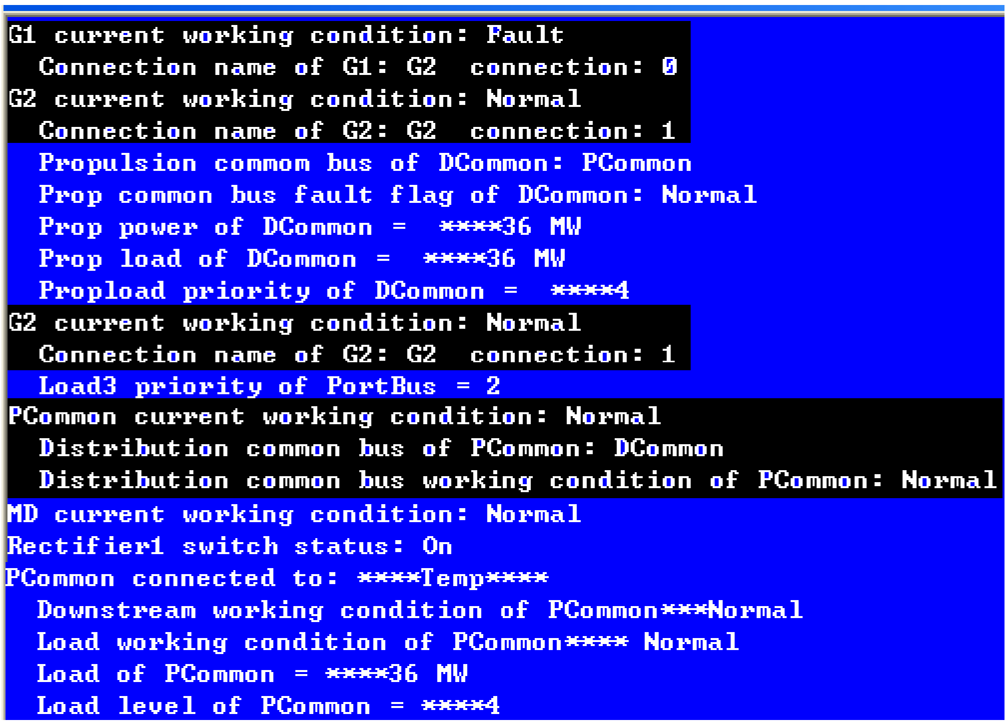


Figure 8.8 Propulsion Common Bus Automatic Switching

Figure 8.9 shows distribution common bus receiving information on tree distribution loads, such as load magnitudes and priority levels.

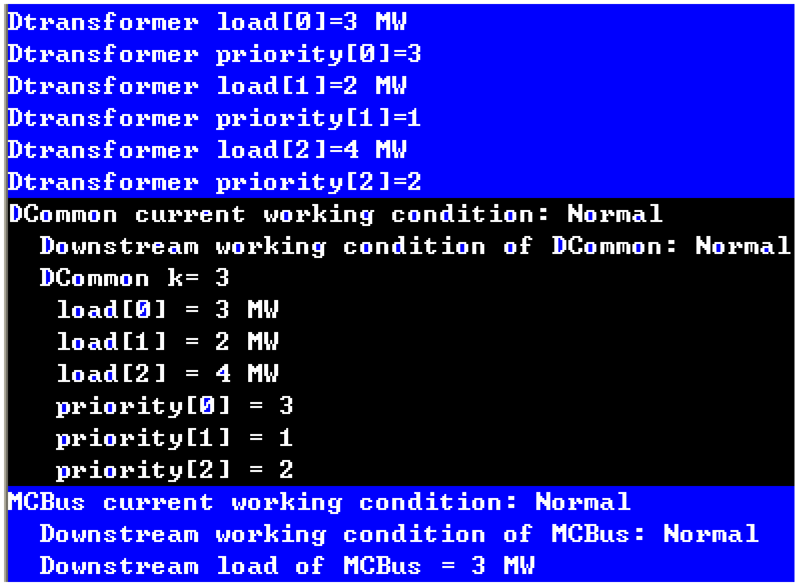


Figure 8.9 Distribution common bus receiving load information

Overall, simulation results, as illustrated in Figure 8.5-8.9, demonstrate the effectiveness of the whole MAS functionality.

8.3 Summary

This chapter describes the topology of the whole MAS, introduces the simulation platform, designs a test case and simulates the entire MAS. The next chapter will summarize conclusions and contributions of this dissertation, and future research topics.

CHAPTER IX

CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

To meet the Navy's increasing demand for more propulsion power and service power, the whole power system onboard is moving toward the integrated power system (IPS) for future all-electric warships. This research focused on analyzing and designing an IPS with AC-DC zonal distribution system based on the Naval Combat Survivability (NCS) DC Distribution Testbed (DCDT).

Multi-agent systems (MAS) are now being developed and applied in many fields of power systems, including power system monitoring and diagnostics, power system protection and reconfiguration, power market simulation, power network control and automation [27]. To increase fighting through and survivability of all-electric ships, the reconfiguration for IPS favors a decentralized system over a centralized system, and the MAS technique is an important candidate for reconfiguration of IPS. This research designed and developed a complicated MAS, which consists of three sub-MAS such as propulsion MAS, generation MAS, and distribution MAS, for reconfiguration of an IPS. The whole MAS can implement reconfiguration functions, including fault area isolation, automatic switching, and load shedding. In this research the MAS had been simulated in

JAVA/JADE platform. Simulation results of test cases demonstrated reconfiguration functionality of the MAS.

The first main contribution of this dissertation is the design of an IPS model, which can be a platform for ship board power system researchers and designers to perform fault protection and reconfiguration, IPS voltage stability and control, and power system management. The second main contribution of this dissertation is the implementation of MAS for reconfiguration of an IPS with AC-DC zonal distribution system. The MAS can execute several reconfiguration functions, such as fault area isolation, automatic switching, and load shedding, specifically for the IPS which combines electric power generation system, electric propulsion system, and AC-DC zonal distribution system.

9.2 Future Work

In this research MAS-based reconfiguration for an IPS is based on the assumption that a traditional power protection system exists and can detect the fault when a fault happens. A traditional power protection system is a centralized control system that has a picture of whole power system and utilizes protective relay coordination to implement fault detection and isolation. If intelligent and distributed agents can be used in IPS to detect faults, replacing protective relays, the protective system will be more reliable and increase the survivability of warships. Thus MAS-based protection system for an IPS will be a good and worthwhile topic in the future.

In this research the MAS includes over forty agents. When the MAS is scaled to a bigger size, agent communication in the PC that runs MAS simulation will be a bottleneck. And it is necessary and possible to scatter these agents on different PCs. These PCs may be distributed in different locations. Thus design and implementation of distributed MAS for reconfiguration of an IPS will become another research direction.

REFERENCES

[1] http://en.wikipedia.org/wiki/Steam_engine

[2] http://en.wikipedia.org/wiki/Internal_combustion

[3] R. Izadi-Zamanabadi and M. Blanke, "A Ship Propulsion System As A Benchmark for Fault-Tolerant Control," *Control Engineering Practice* 7 pp. 227—239, 1999.

[4] Optimal electric ship propulsion system, *Marine Reporter*, 2002.

[5] S.D. Sudhoff, K.A. Corzine, S.F. Glover, H.J. Hegner, and H.N. Robey, Jr., "DC Link Stabilized Field Oriented Control of Electric Propulsion Systems," *IEEE Transactions on Energy Conversion*, Vol. 13, No. 1, March 1998.

[6] T.J. McCoy, "Trends in Ship Electric Propulsion," *Proceedings of the IEEE Power Engineering Society Summer Meeting*, Vol. 1 pp. 343-346, 2002.

[7] Online Material, "Electric Warships and Combat Vehicles," http://www.onr.navy.mil/media/extra/fncs_fact_sheets/electric_ships.pdf

[8] J. Prousalidis and D. Muthumuni, "Power Quality On Electric Ships," www.cedrat.com/software/gallery/embedded/Power_quality_on_electric_ships.pdf

[9] http://en.wikipedia.org/wiki/Intelligent_agents

[10] K.P. Sycara, "Multi-agent Systems," *AI Magazine* 19(2), pp. 79-92, Summer 1998.

[11] K.L. Butler-Purry, "Multi-Agent Technology for Self-Healing Shipboard Power Systems," *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, pp. 207-211, Nov. 2005.

[12] J. M. Solanki and N.N. Schulz, "Using Intelligent Multi-Agent Systems for Shipboard Power Systems Reconfiguration," *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, pp. 212-214, Nov. 2005.

[13] J.M. Solanki, N.N. Schulz, and W. Gao, "Reconfiguration for Restoration of Power Systems Using A Multi-Agent System," *Proceedings of the 37th Annual North American Power Symposium*, pp. 390-395, 2005.

[14] K.Huang, D.A. Cartes, and S.K. Srivastava, "A Multi-Agent Based Algorithm for Mesh-Structured Shipboard Power System Reconfiguration," *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, pp. 188-193, Nov. 2005.

[15] K.Huang, D.A. Cartes, and S.K. Srivastava, "A Multi-Agent Based Algorithm for Ring-Structured Shipboard Power System Reconfiguration," *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 530-535, 2005.

[16] K.Huang, D.A. Cartes, and S.K. Srivastava, "A Novel Algorithm for Agent-Based Reconfiguration of Ring-Structured Shipboard Power System," *Proceedings of the 40th IAS Annual Meeting,* pp. 1311-1316, 2005.

[17] G. Morejon, S.K. Srivastava, and D.A. Cartes, "Integrated Virtual Test Bed and JADE Agent Platform Using CORBA for Reconfiguration of Shipboard Power Systems," *Proceedings of the IEEE Power Engineering Society General Meeting,* 2006.

[18] J. M. Solanki, "Multi-Agent Based Control and Reconfiguration for Restoration of Distribution Systems With Distributed Generators," *Ph.D. dissertation*, Dept. of Electrical and Computer Engineering, Mississippi State University, 2006.

[19] R. Jayabalan and B. Fahimi, "Naval Shipboard Power System," *Proceedings of the IEEE Vehicle Power and Propulsion Conference*, pp. 86-90, Sept. 2005.

[20] K. L. Butler-Purry and N. D. R. Sarma, "Intelligent Network Reconfiguration of Shipboard Power Systems," *Proceedings of the IEEE Power Engineering Society General Meeting*, Vol. 4, pp. 13-17, July 2003.

[21] S.D. Sudhoff, S.F. Glover, S.H. Zak, S.D. Pekarek, E.J. Zivi, D.E. Delisle, and D. Clayton, "Stability Analysis Methodologies for DC Power Distribution Systems," *Proceedings of the Thirteenth International Ship Control Systems Symposium*, Florida, Apr. 2003.

[22] A. Ouroua, L. Domaschk, and J.H. Beno, "Electric Ship Power System Integration Analyses Through Modeling and Simulation," *Proceedings of the IEEE Electric Ship Technologies Symposium*, pp. 70-74, 2005.

[23] S.D. Pekarek, J. Tichenor, S.D. Sudhoff, J.D. Sauer, D.E. Delisle, and E.J. Zivi, "Overview of A Naval Combat Survivability Program," *Naval Engineering and Research Consortium* paper, http://nerc.aticorp.org/papers/overview.pdf

[24] T. Ericsen, N. Hingorani, and Y. Khersonsky, "Power Electronics and Future Marine Electrical Systems," *IEEE Transactions on Industry Applications*, Vol. 42, No. 1, Jan./Feb. 2006.

[25] J.Langston, S.Suryanarayanan, M.Steurer, M.Andrus, S.Woodruff, and P.F.Ribeiro, "Experiences With the Simulation of A Notional All-Electric Ship Integrated Power System on A Large-Scale High-Speed Electromagnetic Transients Simulator," *Proceedings of the IEEE Power Engineering Society General Meeting*, June 2006.

[26] Virtual Test Bed, Online Material, http://vtb.engr.sc.edu/objectives/index_new.asp

[27] S.D.J. McArthur, E.M. Davidson, V.M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications—Part I: Concepts, Approaches, and Technical Challenges," *IEEE Transactions on Power Systems*, Vol. 22, No. 4, Nov. 2007.

[28] E. M. Davidson, S. D. J. McArthur, J. R.McDonald, T. Cumming, and I. Watt, "Applying multi-agent system technology in practice: Automated management and analysis of SCADA and digital fault recorder data," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 559–567, May 2006.

[29] S. D. J. McArthur, S.M. Strachan, and G. Jahn, "The design of a multiagent transformer condition monitoring system," *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1845–1852, Nov. 2004.

[30] T. Nagata and H. Sasaki, "A multi-agent approach to power system restoration," *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 457–462, May 2002.

[31] S. E.Widergren, J. M. Roop, R. T. Guttromson, and Z. Huang, "Simulating the dynamic coupling of market and physical system operations," in *Proceedings of IEEE Power Engineering Society General Meeting, 2004*, Jun. 2004, pp. 748–753.

[32] D. Koesrindartoto, S. Junjie, and L. Tesfatsion, "An agent-based computational laboratory for testing the economic reliability of wholesale power market designs," in *Proceedings of IEEE Power Engineering Society General Meeting, 2005*, Jun. 2005, pp. 931–936.

[33] S. J. Park and J. T. Lim, "Modelling and control of agent-based power protection system using supervisors," *Proc. Inst. Elect. Eng., Control Theory Appl.*, vol. 153, no. 1, pp. 92–98, Jan. 2006.

[34] S. Sheng, K. K. Li, W. L. Chan, Z. Xiangjun, and D. Xianzhong, "Agent-based self-healing protection system," *IEEE Transactions on Power Delivery*, vol. 21, no. 2, pp. 610–618, Apr. 2006.

[35] D. V. Coury, J. S. Thorp, K. M. Hopkinson, and K. P. Birman, "An agent-based current differential relay for use with a utility intranet," *IEEE Transactions on Power Delivery*, vol. 17, no. 1, pp. 47–53, Jan. 2002.

[36] C. Fukui, H. Kudo, J. Koda, K. Yabe, and Y. Tomita, "A cooperative protection system with an agent model," *IEEE Trans. Power Delivery*, vol. 13, no. 4, pp. 1060–1066, Oct. 1998.

[37] R. Giovanini, K. Hopkinson, D. V. Coury, and J. S. Thorp, "A primary and backup cooperative protection system based on wide area agents," *IEEE Transactions on Power Delivery*, vol. 21, no. 3, pp. 1222–1230, Jul. 2006.

[38] A.L. Dimeas, and N.D. Hatziargyriou, "Operation of A Multi-Agent System for Microgrid Control", *IEEE Transactions on Power Systems*, Vol. 20, No. 3, pp. 1447-1455, Aug. 2005.

[39] A. Korand, S.D.J. McArthur, G.W. Ault, G.M. Burt, and J.R. McDonald, "Enabling Active Distribution Networks Through Decentralized, Autonomous Network Management," *Proceedings of 18th International Conference on Electricity Distribution* (CIRED), Turin, June 2005.

[40] Q. Yu and N. N. Schulz, "Modeling and Simulation of MAS-Based Reconfiguration for an Integrated Power System," *GCMS08-International simulation multi-conference*, Edinburgh, Scotland, June 16-19, 2008

[41] Online material: AC-DC Converter Using a Three-Phase Three-Level PWM Voltage-Sourced Converter, http://www.mathworks.com/access/helpdesk/help/toolbox/physmod/powersys/index.html?/access/helpdesk/help/toolbox/physmod/powersys/ref/threelevelbridge.html&http://www.google.com/search?hl=en&q=AC-DC+converter+using+a+three-phase+three-level+pwm+voltage-sourced+converter

[42] Q. Yu and N. N. Schulz, "Design, Modeling, and Simulation of Power Generation and Electric Propulsion System for IPS for All-Electric Ships", *Proceedings of 2007 ASNE Automation and Control Conference*, Biloxi, Mississippi, Dec.10-11, 2007

[43] Brenner et al, "*Intelligent software agents*," Springer, 1998.

[44] Q. Yu and N. N. Schulz, "Multi-Agent Based Reconfiguration of an Electric Propulsion System for All Electric-Ships", *Proceedings of 2007 IEEE North American Power Symposium*, Las Cruces, New Mexico, Sept. 30--Oct. 2, 2007

[45] Online material, "JADE Tutorial—JADE Programming for Beginners," http://jade.tilab.com/doc/JADEProgramming-Tutorial-for-beginners.pdf

APPENDIX

JAVA/JADE CODE FOR INTERMEDIATE AGENT CLASS

```
//-------------------- Intermediate Agent Class ----------------------------
import jade.core.Agent;
import jade.core.AID;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;

public class propintermediateAgent extends Agent
{
  private AID[] distributionAgents;
  private String upstream, downstream, faultFlag, loadFlag, downstreamfaultFlag, switchStatus=
"On";
  private int load, priority_level, power;

  protected void setup()
  {
        // Printout a welcome message
        System.out.println("Hello! Intermediate-agent "+getAID().getName()+" is ready.");

        Object[] args= getArguments();

        if (args != null && args.length > 0)
        {   upstream = (String)args[0];
                downstream = (String)args[1];
                faultFlag = (String) args[2];
        }
        else
        {       System.out.println("Input error, we need three string inputs!");
                doDelete();
        }

        System.out.println("  The upstream neighbor of "+getAID().getLocalName()+" is " +
upstream);
        System.out.println("  The downstream neighbor of "+getAID().getLocalName()+" is " +
downstream);
        System.out.println("  Initial operating condition of the "+getAID().getLocalName()+" is
" + faultFlag);
        System.out.println("  Initial switch status is " + switchStatus);
        //Register the intermediate agent service in the yellow page
        DFAgentDescription dfd = new DFAgentDescription();
        dfd.setName (getAID());
```

```
ServiceDescription sd = new ServiceDescription();
sd.setType("distribution agent");
sd.setName("JADE distribution agent");
dfd.addServices(sd);

try
        {
        DFService.register(this, dfd);
        }
catch (FIPAException fe)
        {
        fe.printStackTrace();
        }

//Add the behaviour: receiving signals
addBehaviour(new receivingSignal());


//Add a genetic behaviour
addBehaviour(new TickerBehaviour(this, 5000)
{
   protected void onTick()
   {

//Update the list of distribution agents
DFAgentDescription template = new DFAgentDescription();
ServiceDescription sd = new ServiceDescription();
sd.setType("distribution agent");
template.addServices(sd);
try
{
  DFAgentDescription[] result = DFService.search(myAgent, template);
  distributionAgents = new AID[result.length];
  for (int i = 0; i < result.length; ++i
  {
     distributionAgents[i] = result[i].getName();
  }

 }
 catch (FIPAException fe)
 {
   fe.printStackTrace();
 }

 //Communication
 //System.out.println(getAID().getLocalName()+" current working condition: "+faultFlag
;

 myAgent.addBehaviour(new sendingSignal());
   }
```

```
        });

  }

  /* Inner class sendingSignal
     This is the behaviour used by intermediate agent to send signals to its upstream propulsion
agent */

  private class sendingSignal extends Behaviour
  {
    public void action()
    {
      ACLMessage msg = new ACLMessage (ACLMessage.REQUEST);
      for (int i=0; i<distributionAgents.length; ++i)
      {
        if (distributionAgents[i].getLocalName().equalsIgnoreCase (upstream))
                { msg.addReceiver(distributionAgents[i]);
                  msg.setLanguage("English" ;
                  msg.setOntology("Power system condition" ;

                  if (faultFlag.equalsIgnoreCase("fault"
                  msg.setContent(faultFlag);
                  else msg.setContent downstreamfaultFlag);
                  myAgent.send(msg);

                  msg.setContent(loadFlag);
                  myAgent.send(msg);
                  msg.setContent(String.valueOf(load));
                  myAgent.send(msg);
                  msg.setContent(String.valueOf(priority_level));
                  myAgent.send(msg);
                }
         }
    }
    public boolean done()
    {
    return true ;
    }
  }

  //----------------------------------------
  protected void takeDown()
  {
    System.out.println("Load-agent "+getAID().getName()+"terminating." ;
  }

//-------
  private class receivingSignal extends CyclicBehaviour
  {
    public void action()
    {
        MessageTemplate mt = MessageTemplate.MatchPerformative(ACLMessage.REQUEST);
```

```
                    ACLMessage msg = myAgent.receive(mt);

                if (msg != null
                {
                  // REQUEST Message received. Process it
                  downstreamfaultFlag = msg.getContent();
//                System.out.println(getAID().getLocalName()+" current working condition:
"+faultFlag );
                  System.out.println(getAID().getLocalName()+" switch status: "+switchStatus );
//                System.out.println("  Downstream working condition of "+getAID().getLocalName()+":
"+downstreamfaultFlag );
                }

            msg = myAgent.receive(mt);
            if (msg != null
                {
                   loadFlag = msg.getContent();
//                 System.out.println("  Downstream load of "+getAID().getLocalName()+":
"+loadFlag ;
                }

            msg = myAgent.receive(mt);
            if (msg != null
                {
                   load = Integer.parseInt ((String msg.getContent());
//                 System.out.println("  Downstream load of "+getAID().getLocalName()+" = "+load+"
MW");
                }

                msg = myAgent.receive(mt);
            if (msg != null
                {
                   priority_level = Integer.parseInt ((String msg.getContent());
//                 System.out.println("  Downstream priority level of "+getAID().getLocalName()+" =
"+priority_level);
                }

                    if (priority_level != 0)
                {  if (downstreamfaultFlag.equalsIgnoreCase("fault"  ||
loadFlag.equalsIgnoreCase("fault"
                       switchStatus = "Off";
                }
        }
    }
//-------
}
```

//--------------------- Intermediate Agent Class End------------------------------