

THE MOB SHIFT SIMULATION FRAMEWORK

J. Borges de Sousa¹, A. Girard² and N. Kourjanskaia³
Faculdade de Engenharia da Universidade do Porto, Portugal¹
The University of California at Berkeley²
California PATH, The University of California at Berkeley³

ABSTRACT

MOB SHIFT is a domain specific simulation framework enabling the timely and cost-effective analysis, design and evaluation of competing Mobile Offshore Base concepts. It was developed in SHIFT, a specification language for describing dynamic networks of hybrid automata. The expressive power of SHIFT provides a compact notation for modeling spatial and logical relationships in a dynamic environment and for modeling and analyzing control strategies governing object interactions, such as the coordination and control of multiple vehicles.

The simulation architecture consists of a structure of layers and of the respective interfaces. The layers of the simulation architecture correspond to building blocks of the underlying physical and control models.

This paper describes the MOB SHIFT simulation framework and the accompanying tool set. MOB SHIFT permits a realistic estimate of dynamic positioning capabilities in different environments and provides a tool to uniformly design and evaluate competing multi-module dynamic positioning control systems for the MOB.

Keywords: Dynamic positioning (DP), SHIFT, Simulation, Mobile Offshore Base.

1 INTRODUCTION

The Mobile Offshore Base (MOB) is a Science & Technology Program conducted by US Office of Naval Research to advance technologies essential for establishing the feasibility of building a Mobile Offshore Base (MOB) and to determine whether the MOB concept represents a credible system for Naval and Marine forces [1, 11, 18]. The envisioned system will consist of three to five interconnected modules and would accept cargo from conventional take-off and

landing (CTOL) aircraft, as well as from container ships. Overall, it will provide approximately 3 million square feet of reconfigurable internal storage, 10 million gallons of fuel, and it will house up to 3,000 troops (which is equivalent to an Army heavy brigade) [12]. It will be able to project these resources to the shore via landing craft.

Other key general characteristics of a MOB include:

- Length up to 2 km and width of approximately 120 ft.
- Low ocean wave-induced motion to support CTOL operation of cargo aircraft up to Sea State 6.
- High throughput, open-ocean ship to MOB and MOB-to-cargo ship transfers through Sea State 3.
- Platform survivability in any incident storm (e.g., hurricane and typhoon).
- Maintainability of 40 years between overhauls.
- Long-term station-keeping in deep water.

The last characteristic, long-term station-keeping, is the most common function usually ascribed to dynamic positioning (DP) control of semi-submersibles. For the "Independent Semi-Submersible Modules" MOB concept illustrated in Figure 1, however, three nearly 500-m long semi-submersible steel modules are functionally connected by drawbridges which provide a continuous airplane runway. The DP system must therefore substitute for structural load-bearing connectors to maintain overall orientation and relative position between modules. Variations of this concept include a hybrid DP-physical connector scheme, where the structural connector suffices for low magnitude environmental disturbances, and as the Sea State increases, a combination of the connector and DP is invoked to maintain the desired interconnected state until the physical connection is broken. At a certain point in disturbance magnitude, the entire

¹ DEEC/FEUP, R. dos Bragas, 4099 Porto Codex, Portugal. E-mail: jtasso@fe.up.pt

² 5141 Etcheverry Hall, University of California at Berkeley, Berkeley, CA 94720, USA. E-mail: anouck@robotics.eecs.berkeley.edu

³ 1357 S. 46th St RFS Bldg. 452, Richmond CA 94804. E-mail: natasha@path.berkeley.edu

interconnectivity is terminated, and air operations are aborted.

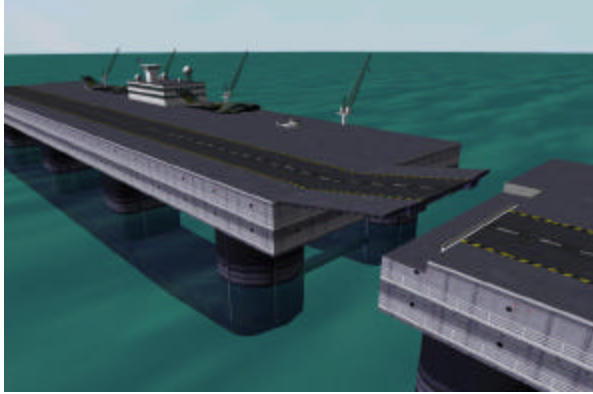


Figure 1 Bechtel Concept: Independent Semi-Submersible Module MOB

This paper is organized as follows. In Section 2, we discuss the virtual demonstration of the MOB control and evaluation method that is being developed at Partners for Advanced Transit and Highways (PATH) UC Berkeley under the CASSETTE project, and in particular, the MOB SHIFT simulation framework [9]. In Section 3, we briefly discuss how to model multiple interacting systems with dynamic networks of hybrid automata, the underlying model of the SHIFT simulation language [4, 5, 19]. In Section 4, we present a detailed description of MOB SHIFT, emphasizing the reconfigurable simulation architecture and physical models. In Section 5, we discuss the utilization of this environment to uniformly support control designs and performance evaluations. In Section 6, we discuss the conclusions and future work.

2 THE CASSETTE PROJECT

2.1 Objectives

The CASSETTE (Development And Demonstration Of Control And SyStem EvaluaTion TEchniques) project from UC Berkeley is part of the overall MOB feasibility evaluation. In this project we are developing an automated Multi-Module Dynamic Positioning Control System (MMDPCS) method for the MOB (see companion paper [15]), and a tool template to uniformly support MMDPCS designs and performance evaluations.

Under this project the team was tasked to virtually demonstrate the MOB control and evaluation method, and physically validate the key design issues with scale models of a MOB.

Figure 2 describes the identification and development of analysis and simulation tools, and the interactions between the elements of the project.

A SHIFT model of the platforms has been developed

using the standard equation of motion for ships [13, 14] and outputs from the WAMIT [21] software package. WAMIT inputs the physical characteristics of the platforms and generates the hydrodynamic parameters that are needed to model the platform dynamics and wave forces. In parallel, the control architecture and a set of evaluation tools were developed, and physical models were designed and built in order to conduct an experiment. The physical models will be used to validate the MOB SHIFT simulation framework, eventually leading to the reformulation of the SHIFT model. Upon validation of the SHIFT model, we will be able to go through the usual iteration cycle of design, evaluation and optimization.

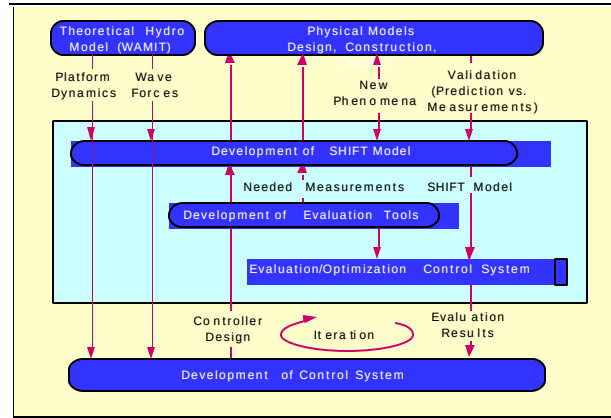


Figure 2 Interactions between elements of the project

2.2 Design and evaluation perspective

The MOB SHIFT simulation framework is the kernel of the MOB SHIFT evaluation framework. The simulation framework provides SHIFT models of MOB concepts. These will be tested in the scenarios specified according to the MOB SHIFT evaluation framework (see Figure 3).

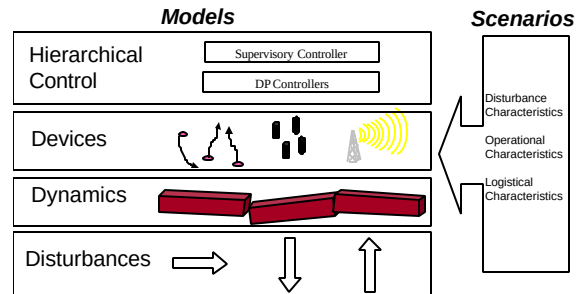


Figure 3 Design and evaluation perspective

For the purpose of the CASSETTE MMDPCS design and performance evaluation, the MOB system comprises a hierarchical control architecture and

independent systems. Each independent system consists of a semi-submersible platform, a power plant, thrusters, sensors, flexible bridges and/or connectors. In the context of the feasibility evaluation of the MMDPCS, the MOB SHIFT evaluation framework comprises:

- 1) A test plan and a set of evaluation criteria.
- 2) An Evaluation Toolbox that automates the execution of the test plan and the evaluation of the test runs.

The test plan, consisting of a set of test cases, is organized in terms of a combination of operational scenarios and utilization environments. For each operational scenario there are several utilization environments. Scenarios are organized into phases. Some scenarios share exactly the same phases, thus reducing the need for replications and for different warm-up phases.

Under the assumptions described in [10], the three types of factors, natural or induced, that may affect the system performance or render the MOB inoperable are:

- 1) Environmental disturbances.
 - a) Wind.
 - b) Waves:
 - i) Slow drift forces.
 - ii) Steady drift forces.
 - iii) Wave frequency motions.
 - c) Ocean currents.
 - d) Solitons.
- 2) Thruster failures.
- 3) Relative position sensor failures (that may induce drive-off).

Hence, an utilization environment is a combination of instances of the three types of factors.

The evaluation criteria consist of several Measures Of Effectiveness (MOE) that reflect the overall expectations. These include, among others, operability and performance (see [10]). Operability is defined in terms of uptime for air operations. This evaluation framework is mainly focused on control system designs. Accordingly, the measures of performance are independent of particular designs and reflect the underlying functional requirements. There are several measures of performance (MOP) for the control systems.

Obviously, the evaluation framework provides objective conditions for the refinement and fine-tuning of the MMDPCS. These are intended to provide a level of abstraction that enables to refine the understanding of the DP operation without cluttering the analysis with details that are not mission critical, or are still unknown and under investigation.

2.3 The tool set

The tool set to uniformly support MMDPCS designs and performance evaluations consists of the MOB

SHIFT simulation framework and the Evaluation Toolbox. This tool set can be used in several ways to:

- Test DP controllers and specific MOB configurations. The specific MOB configuration is automatically generated by MOB SHIFT from a user supplied file with the MOB parameters. The DP controller can be coded as a c-function or as a plug-and-play SHIFT component. The **DP_Controller** SHIFT component can be reused for this purpose. In terms of Figure 3 this means using the Evaluation Toolbox to run test cases with models of disturbances, dynamics and devices from MOB SHIFT and user provided DP controllers.
- Test and evaluate alternative control architectures. This task requires the development of the hierarchical control structure in SHIFT. In terms of Figure 3 this means using the Evaluation Toolbox to run test cases that use the first three layers of models from MOB SHIFT and the user provided hierarchical control structure. The CASSETTE MMDPCS SHIFT implementation can be reused in this process.

3 SHIFT LANGUAGE AND MODELING

3.1 Simulation of the MOB operation

The description requirements for the virtual demonstration of the MOB operation are very complex since the whole system includes both continuous activities and discrete-event features (i.e., constitutes a hybrid system) evolving in several time scales. The dynamic nature of the problem stems from the existence of multiple vehicles whose roles, relative positions, and dependencies change during operations.

The MOB SHIFT simulation environment was implemented in SHIFT, a programming language that describes dynamic networks of hybrid automata [5]. Networks of hybrid automata constitute a powerful modeling framework for this problem domain that is suitable for incorporating the discrete event and continuous dynamics that arise in the coordinated operation of multiple modules.

The expressive power of SHIFT, in terms of the coordination and control of multiple vehicles, has been successfully illustrated in several problem domains:

- SMART-AHS, a simulator of Automated Highway Systems, developed at PATH [22].
- The simulation of the coordinated operation of multiple Autonomous Underwater Vehicles (AUV) for coastal oceanography [17].
- The design and implementation of an architecture for the coordinated operation of multiple autonomous air vehicles [20].

3.2 SHIFT, a brief description

SHIFT users define types (classes) with continuous and discrete behavior. A simulation starts with an initial

set of components that are instantiations of these types. The world-evolution is derived from the behavior of these components.

A type consists of numerical variables, link variables, a set of discrete states, and a set of event labels. The variables are organized into input, state, and output variables. The inputs and outputs of different components can be interconnected. Each discrete state has a set of differential equations and algebraic definitions (flow equations) that govern the continuous evolution of numerical variables. These equations are based on numerical variables of this type and outputs of other types accessible through link variables. The transition structure of the hybrid automata may involve synchronization of pairs or sets of components.

The system behavior alternates between the continuous mode, where the evolution is governed by the flow equations, and the discrete mode, where simulation time is stopped and all possible transitions, determined by guards on transitions and/or by event synchronization among components, are taken. During a discrete step components can be created, interconnected, and destroyed. Currently the continuous mode is implemented by a fixed step Runge-Kutta integration algorithm and the step size determines the accuracy of the simulation.

The SHIFT prototype-based description is summarized in Figure 4.

Prototype based description

```

type Vehicle {
  input ... what we feed to it
  output ... what we see on the outside
  state ... what's internal
  discrete ... discrete modes of behavior
  export ... event labels
  flow ...
  transition ...
  setup ... actions executed at create time
}

```

} data model

} continuous and discrete evolution

Figure 4 SHIFT prototype based description

A SHIFT type provides a prototype of entities evolving in the world. An instance of a type is called a component. SHIFT allows for the construction of sets and arrays of components. The world evolution can be described in terms of components, sets and/or arrays of components and the respective interactions. Components can be created and destroyed during the simulation, while the overall set/array structure is maintained. There are two types of interactions between components. These are given in terms of:

- Data dependencies, specified as input/output links.
- Event synchronization dependencies.

These interactions can change dynamically. Figure 5

illustrates these concepts. For example, event a, in the automaton B, is synchronized with event h in automaton C, meaning that the corresponding transitions will take place at the same time.

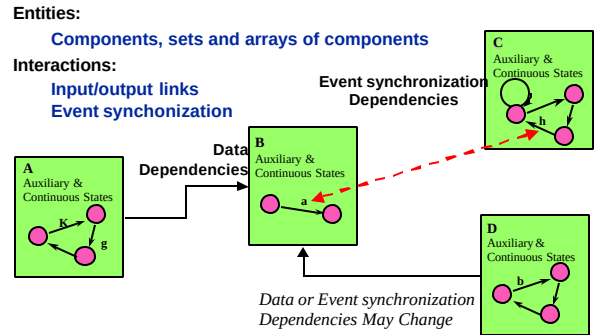


Figure 5 Dynamic Networks of Hybrid Automata

3.3 Modeling dynamic interactions

There are several modes of operation for the MOB (see [10]):

- 1) Unassembled mode. Each individual module maintains a prescribed position and orientation.
- 2) Assembled mode. The MOB assembly maintains a general position and orientation, and the individual modules maintain relative positions and orientations.
- 3) Transitioning modes. Independent modules are in preparation for mechanically connecting or disconnecting the other modules end-to-end.

Each mode requires different control configurations from the MMDPCS. Transitions between modes are triggered by commands during nominal operation, or by uncontrolled events. These transitions may require the dynamic reconfiguration of the interactions, input/output and/or synchronization dependencies between the elements of the MMDPCS. Accordingly, the system behavior can be described by the co-evolution of continuous and discrete dynamics.

Now the role of the SHIFT modeling capabilities in terms of simulating the MOB operations becomes clear. In order to illustrate these capabilities consider Figure 6.

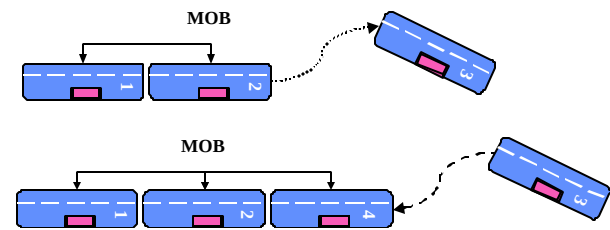


Figure 6 Dynamic interactions

For the sake of simplicity, and just in this example,

we define the MOB to be a group of two or more linked independent modules. The figure represents two snapshots of the MOB operation. In the first one, module 3 is maneuvering in order to disassemble from the other two modules that compose the MOB. Before the disassemble command the operation of the three MOB modules was governed by a dynamic positioning controller. Afterwards, another configuration of the DP controller controls the two-module MOB. Meanwhile, module 3 is under independent control action. The second snapshot is taken at a later stage, where a three to four MOB configuration change is about to take place. Upon meeting the docking requirements (with respect to the MOB), module 3 is commanded to join the MOB. Upon successful completion of this maneuver, module 3 is under the control of the DP controller, possibly in a different configuration. All configuration changes and coordination of the modules are handled in SHIFT through event synchronization and dynamic reconfiguration of input/output links.

4 MOB SHIFT

4.1 Notation

Throughout the rest of the document we use the following convention. Text formatted in **Bold** indicates SHIFT components.

4.2 The simulation architecture

The simulation architecture is represented in Figure 7 and is generally described below.

The MOB SHIFT type (not explicitly represented in Figure 7) models the MOB. The MOB type is composed of two main entities:

- **Supervisory_Controller**: The SHIFT type that models the supervisory control of the MOB.
- **Platforms**: Set of MOB_Platform as defined below.

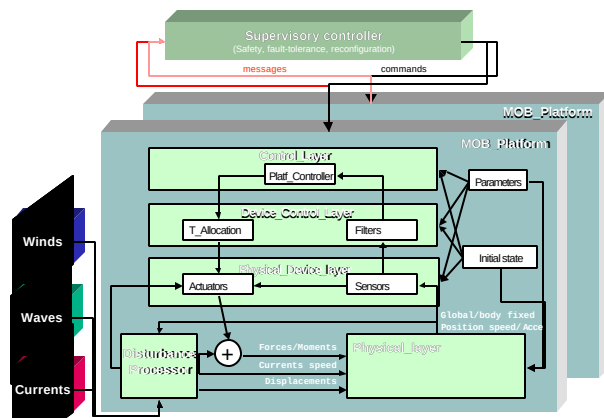


Figure 7 MOB SHIFT Simulation Architecture

The **Supervisory_Controller** sends commands to the set **Platforms** and receives messages from the

corresponding controllers. The set operations provided by SHIFT are essential for representing time-varying physical and control configurations. Although the composition of each set may vary with time, the coherency of the overall input/output link structure is maintained.

The **MOB_Platform** type is composed of the following types:

- **Physical_Layer**: encapsulates the dynamic model of the platform.
- **Disturbance_Processor**: models transfer functions from environmental disturbances into perturbations to the dynamic model of the platform.
- **Parameters**: provides the parameters describing each platform.
- **Initial_state**: provides the initial state of all the variables for the simulation.
- **Summation unit**.
- **Physical_Device_Layer**: models the set of sensors and actuators for a particular platform configuration.
- **Device_Control_Layer**: models the thrust allocation logic, filters and observers.
- **Control_Layer**: provides a set of controllers for all modes of operation.

The inputs to **Physical_Layer** are:

- Force/moment vector $F = (X, Y, N)$. The resulting forces on the platform are obtained by adding the corresponding outputs from the **Physical_Device_Layer** (generated by the corresponding actuators) and from the **Disturbance_Processor**.
- Speed of currents. Generated by the **Disturbance_Processor**.
- Wave frequency motions. Generated by the **Disturbance_Processor**.

The outputs are:

- Position and speed in inertial coordinates.
- Speed and acceleration in body fixed coordinates.

The inputs to the **DisturbanceProcessor** are the instantaneous platform position and speed (outputs from the **Physical_Layer**) and the environmental disturbances (winds, currents and waves). These inputs are transformed into the corresponding effects (outputs) for a specific platform (determined by the platform response operators):

- Forces and moments.
- Displacements or accelerations.
- Local speed of ocean currents.

The **Physical_Device_Layer** models the sensors and actuators available in a specific MOB concept. The inputs are:

- Commands for each actuator (specified by the **Device_Control_Layer**).
- Position and speed in inertial coordinates; speed

and acceleration in body fixed coordinates (from **Physical_Layer**).

The outputs are:

- The resulting force system generated by all actuators.
- Sensor outputs for a particular sensor configuration.

The **Device_Control_Layer** consists of the **T_Allocation** (thrust allocation) component and a set of filters and observers.

The inputs are:

- A commanded force system (specified by the **Control_Layer**).
- Sensor outputs (from **Physical_Device_Layer**).

The outputs are:

- Commands for each actuator.
- Position, speed and acceleration estimates as determined by the layer configuration for a particular MOB concept.

The **Control_Layer** consists of a set of **Platf_Controller** that are invoked by commands from the Supervisory Controller. The inputs are:

- Controller settings and references.
- Position, speed and acceleration estimates.

The outputs are:

- Commanded force system.
- Position, speed and acceleration estimates.

4.3 Implementation issues

The simulation architecture consists of an organized structure of layers and the respective interfaces. These layers correspond to building blocks of the underlying physical and control models. This same structure was used to organize the two thrusts of the implementation effort:

- The overall structure was implemented in SHIFT.
- Disturbance models and some of the DP controllers were implemented in c-code using the SHIFT external function interface.

The utilization of external c-functions is particularly convenient to implement:

- Computationally intensive algorithms, such as the disturbance models.
- Controllers that will be ported to a real-time operating system for the physical demonstration of the CASSETTE MMPDCS.

In terms of the SHIFT implementation:

- The **Supervisory_Controller** is initially supplied as an empty type in MOB SHIFT. Specific implementations of MMDPCS concepts are user-defined.
- The **Physical_Layer** and **Disturbance_Processor** types are supplied as ready-to-use parameterized components. The corresponding parameters are read from configuration files at creation time.

- Instances of the **Physical_Device_Layer** are configured at creation time according to a file specification. For example, the number and location of thrusters, for a particular MOB concept, may be changed from one simulation to another without the recompilation of the SHIFT code.
- The **Device_Control_Layer** and the **Control_Layer** types are provided as SHIFT templates that allow the incorporation of controllers and filters as plug-and-play components.

4.4 CASSETTE hierarchical control architecture

We developed the CASSETTE MMDPCS concept for the MOB as a particular instance of MMDPCS and **Supervisory_Controller** (see companion paper [15]).

4.5 Physical models

The equations of motion of a platform when derived in their complete form are quite complex, not only because of the sheer number of equations and forces, but also because the hydrodynamic forces are complex. In this section, we present the low frequency model of the platform dynamics that was implemented in SHIFT. This model describes the motions in the three primary axes of the dynamic positioning system. The three body-fixed axes are taken on the platform as defined in [Fossen], and are x position (directed from aft to fore), y position (directed to starboard) and heading.

DOF		Body-fixed Velocities	Inertial positions
1	Motion in x-direction	u	x
2	Motion in y-direction	v	y
6	Rotation about z-axis	r	ψ

The motion of the body-fixed frame is described relative to an inertial frame. The equations of motion can be written in the body-fixed frame as:

$$M\dot{n} + C(n)n + D(n)n = t$$

$$\dot{h} = J(h)n$$

Where:

- $n = \begin{bmatrix} u \\ v \\ r \end{bmatrix}$ represents the velocity of the platform in the body-fixed frame
- $h = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix}$ is the velocity of the platform in the inertial frame
- M is the mass + added mass matrix of the platform

- $C(v)$ is the Coriolis and centripetal matrix
- $D(v)$ is the damping matrix
- τ represents the body-fixed forces from the actuators, viscous drags and the environment
- $J(\eta)$ is a transformation matrix with:

$$J(h) = \begin{bmatrix} \cos(y) & -\sin(y) & 0 \\ \sin(y) & \cos(y) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4.6 Environmental modeling

The environmental effects accounted for in MOB SHIFT are due to wind, current, waves and solitons (see Figure 8). As most of the models are computationally intensive, the disturbances were implemented as c-functions that are called from SHIFT using the external function interface. For modularity purposes, the c-functions were then encapsulated into the **Disturbance_Processor** components.

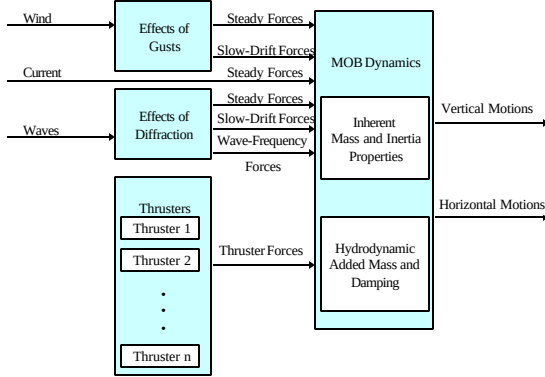


Figure 8 Environmental inputs

- Forces due to wind

Wind forces and moments on a MOB module are described in terms of a mean wind speed in combination with a wind spectrum that characterizes the variations of the wind speed [13]. We used the Davenport (1961) spectrum in the spectral formulation of the wind gust:

$$S(w) = k \frac{916700w}{[1 + (191w / V_w(10))^2]^{4/3}}$$

where

- $k=0.05$ is the turbulence factor
- $V_w(10)$ is the average wind speed at a level of 10m above the water surface (in knots)
- ω is the frequency of the wind oscillations (in rad/s)

Once the random portions of the wind are generated, they are added to the appropriate mean values to obtain the total wind speed V_T and used to compute the wind forces and moments to be applied to the platform:

$$\begin{bmatrix} F_x \\ F_y \\ N \end{bmatrix} = \begin{bmatrix} A_T C_x(a) \\ A_L C_y(a) \\ A_L L C_n(a) \end{bmatrix} \frac{\rho_w}{7.6} V_T^2$$

where ρ_w is the density of air in kg/m^3 , and the $C(\alpha)$ are coefficients depending on the angle of attack of the wind α .

- Impact of currents

Ocean currents are assumed to be steady and known or measured accurately. They are taken into account directly as velocities acting on the module.

- Wave effects

The effects of ocean waves on the platform are of three kinds: steady wave diffraction forces, slow drift forces and wave-frequency forces.

- *Steady wave diffraction forces – Slow drift forces*

The platform motions due to waves are of two kinds:

- 1) Vertical motions (roll, pitch, heave). These motions occur at the incoming wave frequency. The forces that cause these motions are too large to control.
- 2) Horizontal motions (surge, sway, yaw). These motions occur at the wave frequency and can have steady/slow drift components. Only the horizontal forces causing the steady and slow-drift components have been considered in the controller design.

The wave frequency excitation is approximately linear with wave amplitude and zero-mean. The steady drift and slow drift forces are quadratic with wave amplitude. In both cases, the mean forces were estimated using WAMIT.

- *Wave frequency motions*

Ocean surface waves are taken to follow the Bretschneider (1959) spectrum for wave spectral density functions [6, 7, 8]:

$$S(w) = \frac{1.25}{4} \frac{w_0^4}{w^5} H_s^2 \exp \left[-1.25 \left[\frac{w_0}{w} \right]^4 \right]$$

where ω_0 is the modal frequency and H_s is the significant wave height.

Waves have a dominant direction; directional characteristics were described by using a spreading

function that varies as a $\cos^2$ (angle between platform orientation and wind direction).

The actual seaway is modeled as the superposition of many regular waves using the principle of superposition. The response of a platform to an actual random seaway can be predicted from the response of the platform to regular waves. To obtain the regular wave response of the platform, WAMIT was used to compute the Response Amplitude Operators and phase components. The ship motions are then obtained from:

$$h^k(x, y, t) = \sum_i \sum_j a_{ij} RAO_{ij}^k \cos(k_i(x \cos \psi_j + y \sin \psi_j) - \omega_i t + \phi_{ij})$$

where:

- $i = 1..M$, M being the number of frequencies considered in the seaway
- $j = 1..N$, N being the number of directions considered in the wave directional spectrum
- $k = 1..6$ represents the mode of motion
- a_{ij} is the wave amplitude
- ω_i is the wave frequency
- k_i is the wave number
- ψ_j is the direction of the wave with respect to the platform
- ϕ_{ij} is a phase component.

- Solitons

Solitons have been modeled as described by [6].

4.7 Performance issues

MOB SHIFT consists of approximately 4800 lines of SHIFT code and 2200 lines of c-code. The SHIFT implementation of the CASSETTE hierarchical control architecture represents some additional 1400 lines of code.

An integration time step of 1 second is adequate, in terms of accuracy requirements, for a simulation run of the CASSETTE MMDPCS hierarchical control architecture. The simulated operation of 3 independent modules under this control scheme runs three times faster than real time on a SUN ULTRA-2 workstation with 245 MB of memory and Solaris 5.5.1 operating system.

5 UTILIZATION OF MOB SHIFT

Next, we briefly describe the utilization of MOB SHIFT to support control design and performance evaluations. This is a 4-stage process.

- 1) Implementation of the SHIFT simulation model for the particular MMDPCS concept.
- 2) Design of the test plan and evaluation criteria. The test plan specification consists in filling a test plan template.
- 3) Running the test plan under the Evaluation Toolbox. The Evaluation Toolbox runs the test

plan according to the user-supplied specifications and generates all the required data for posterior analysis.

4) Data analysis.

The SHIFT simulation model of a specific MMDPCS concept is assembled from the user supplied SHIFT implementation of the concept and from the following libraries (partly depicted in Figure 9):

- MOB SHIFT simulation.
- Control.
- Disturbances.
- Model parameters.

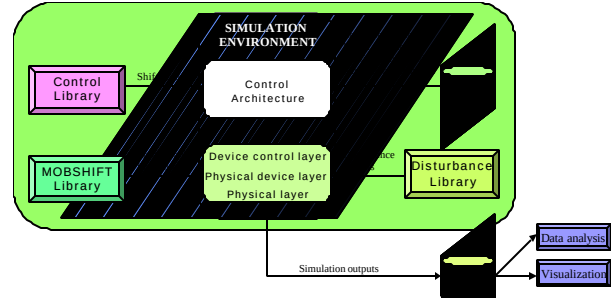


Figure 9 Utilization of MOB SHIFT

6 CONCLUSIONS

In order to uniformly evaluate MMDPCS designs, a SHIFT-based simulation and evaluation tool set has been developed and presented in this paper. It is complete with environmental disturbance models and plug-and-play capability for control systems and platform models.

The unique features of the SHIFT language, particularly the hybrid automata model, the dynamic reconfiguration of the model, the powerful synchronization constructs and the operations on sets were instrumental in the process of modeling the problem domain and in the specification of a structure where control architectures can be tested.

A SHIFT-based evaluation is currently under way for DP controllers developed by UC Berkeley [2] and Scientific Systems Company, Inc. [3]. The MOB SHIFT simulation framework will be validated by experiments to be conducted in the upcoming months.

Future work might entail testing and comparing different MMDPCS, and/or extending the existing hierarchical architecture (according to [15]).

The MOB SHIFT framework can easily be expanded into a more powerful design and evaluation framework by incorporating models and concept implementations concurrently developed by independent teams. Hence, MOB SHIFT can be used as a generic software package to facilitate the timely and cost-effective analysis, design, evaluation and implementation of the MMDPCS.

Acknowledgements: The authors would like to acknowledge the support for this work by the Office of Naval Research under grant N000114-98-0744. The authors also thank Prof. William Webster, Prof. Karl Hedrick, Dr. Akash Deshpande and Mr. Jim Misener for stimulating discussions and invaluable comments, insights and contributions.

REFERENCES

- [1] R. Zueck, P. Palo and R. Taylor. Mobile Offshore Base: Research Spin-Offs. In *Proc. Of the ISOPE Conference, Brest, France*, pp 10-16, 1999.
- [2] K. Hedrick, A. Girard and B. Kaku. A Coordinated DP Methodology for the MOB. In *Proc. Of the ISOPE Conference, Brest, France*, pp 70-75, 1999.
- [3] V. Manikonda, M. Gopinathan, P. Arambel and R. K. Mehra. Nonlinear Model Predictive Control Design for Coordinated Dynamic Positioning of a Multi-Platform Mobile Offshore Base. Accepted for the *VLFS'99 Conference, Honolulu, Hawaii, September 1999*.
- [4] L. Semenzato, A. Deshpande and A. Gollu. *SHIFT Reference Manual*. PATH Report. 1997.
- [5] A. Deshpande, A. Gollu and L. Semenzato. The SHIFT Programming Language and Run-time System for Dynamic Networks of Hybrid Automata. *California PATH Research Report UCB-ITS-PRR-97-7*
- [6] E. Lewis (ed.). Principles of Naval Architecture. 2nd revision. Jersey City, N.J.: Society of Naval Architects and Marine Engineers, 1989.
- [7] R. F. Beck, W. E. Cummins, J. F. Dalzell, P. Mandel and W. C. Webster. Motion in Waves. *Principles of Naval Architecture, Vol.3, Chapter 8*, 1989.
- [8] O. M. Faltinsen. Sea Loads on Ships and Offshore Structures. *Cambridge University Press*, 1990.
- [9] The MOB SHIFT Simulation Framework. *CASSETTE Technical Progress Report #2*, 1999.
- [10] *CASSETTE Technical Progress Report #3*, 1999.
- [11] G. Remmers, R. Zueck, P. Palo and R. Taylor. Mobile Offshore Base. In *Proc. Of the ISOPE Conference, Montreal, Canada*, p. 1-5, 1998.
- [12] J. Turner, M. Murdoch, D. Wilkins, and P. Lopez. Functional Requirements Definition for a Mobile Offshore Base (MOB). *Naval Logistics Conference*, 1998.
- [13] T. Fossen. Guidance and Control of Ocean Vehicles. *Wiley, New York*, 1994.
- [14] J. Newman. Marine Hydrodynamics. *MIT Press, Cambridge, MA*, 1977.
- [15] A. Girard, J. K. Hedrick, and J. Borges de Sousa. A Hierarchical Control Architecture for Mobile Offshore Bases. Accepted for the *VLFS'99 Conference, Honolulu, Hawaii, September, 1999*.
- [16] J. Misener, A. Girard, J. Borges de Sousa and Karl Hedrick. Control and Evaluation of Mobile Offshore Base Operations. In *Proc. of the SPIE Vol. 3693 Conference Unmanned Ground Vehicle Technology, USA*, 1999.
- [17] J. Borges de Sousa and A. Gollu. A SHIFT Simulation Environment for the Coordinated Operation of Multiple Autonomous Underwater Vehicles. In *Proc. of the Winter Simulation Conference, Atlanta, USA*, 1997.
- [18] <http://mob.nfesc.navy.mil/>
- [19] <http://www.path.berkeley.edu/shift/>
- [20] <http://robotics.eecs.berkeley.edu/~simsek/SEMINAR/>
- [21] T. Korsmeyer, T. Klemas, J. White and J. Phillips, Fast Hydrodynamic Analysis of Large Offshore structures. In *Proc. Of the ISOPE Conference, Brest, France*, pp. 27-34, 1999.
- [22] M. Antonioti, A. Deshpande, A. Girault. Microsimulation Analysis of a Hybrid System Model of Multiple Merge Junction Highways and Semi-Automated Vehicles. *Proceedings of the IEEE Conference on Systems, Man and Cybernetics, Orlando, FL, U.S.A. October, 1997*.