Control and evaluation of Mobile Offshore Base operations

Anouck Girard\textsuperscript{a}, James Misener\textsuperscript{a}, Joao Sousa\textsuperscript{b}, Karl Hedrick\textsuperscript{a}

\textsuperscript{a}California PATH, University of California at Berkeley, 1357 S. 46\textsuperscript{th} St RFS Bldg. 452, Richmond CA 94804
\textsuperscript{b}Faculty of Engineering, University of Porto, Rua do Bragas, 4099 Porto Codex, Portugal

ABSTRACT

We are developing dynamic position (DP) control and evaluation systems for semi-submersible vessel system called a Mobile Offshore Base (MOB). In concept, the MOB is a self-propelled prepositioned floating base consisting of three to five vessels, and comprising a mile-long runway to accommodate C-17 take-off and landing operations and allow cargo transfer from container ships. Separate MOB barges would embark toward a preposition point about 100 km offshore, assemble along a line, then execute a military mission in a variety of sea states. Specific concepts call for them to be mechanically or electronically linked, while a concept refinement uses a hybrid approach, linking them mechanically during low sea states and electronically once the environmental disturbances increase.

We discuss issues and approaches with MOB control, with a focus on the overarching control architecture. We frame our discussion, however, on microsimulation techniques derived from a discipline best described as simulation of dynamically reconfigurable multi-agent hybrid dynamic systems. Specifically we describe the intended use of our microsimulation technique to evaluate various control concepts and ultimately, to test the feasibility of employing DP on the MOB.

Keywords: supervisory control, control architectures, dynamic positioning (DP), SHIFT simulation, microsimulation, 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 credible system for Naval and Marine forces\textsuperscript{1}. The envisioned system would 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)\textsuperscript{2}. It will be able to project these resources to the shore via landing craft.

Other key general characteristics of a MOB include\textsuperscript{3}:

- Length up to 2 km and width of approximately 120 m
- 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 also 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 interconnectivity is terminated, and air operations are aborted.

\textsuperscript{*} Correspondence: Email: anouck@robotics.eecs.berkeley.edu; WWW: http://path.berkeley.edu/~anouck/index.html; Telephone: 510 642 3248; Fax: 510 231 9565
In the remainder of this paper, we will cover the types of DP functions required for a generic independent semi-submersible MOB, focusing on the control architectures and discrete control logic.

2. THE DYNAMIC POSITIONING CONTROL PROBLEM

The multi-module DP system integrates propulsion hardware, sensors, and controller software to allow the system of MOB vessels to hold a desired position and orientation (surge, sway and yaw) under dynamic environmental forces of wind, wave and currents. Dynamic positioning system efficiency is crucial to the Independent Semi-Submersible Module and its variants. We are addressing this at a variety of levels, to include detailed nonlinear position control design, but to also encompass the overarching control architecture.

To briefly describe the position control design, we begin with the nonlinear equations that describe the low-frequency horizontal motion of a surface vessel. They may be written as

\[
(M_{RB} + M_A) \dot{\mathbf{v}} + C_{RB} (\mathbf{v}) \mathbf{v} + C_A (\mathbf{v}) \mathbf{v}_r = F_{env} + F_T
\]

where \( M_{RB} \) and \( M_A \) are the body mass and added mass matrices, \( C_{RB} \) and \( C_A \) are the body coriolis/centripetal and added coriolis/centripetal matrices, and \( \mathbf{v} = [u \ v \ r]^T \) is a vector of body-fixed velocity components in the surge, sway, and yaw directions. \( \mathbf{v}_r = [u_r \ v_r \ r]^T \) is a vector of body-fixed relative velocity components, where the relative velocity is the velocity of the vessel relative to the water. \( F_{env} \) contains the body-fixed components of wind forces, viscous current forces, and second-order wave forces. \( F_T \) contains the body-fixed thruster force components: surge force, sway force, and yaw moment. The body-fixed velocity components are related to the earth-fixed velocity components by:
\[ \eta = J(\eta)\nu \]  \hspace{1cm} (2)

where

\[
\eta = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} \quad \text{and} \quad J(\eta) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\]

Hedrick, et. al.\(^6\) describe the robust performance of a nonlinear control algorithm to operate on Equation 1. Hedrick also investigates single and multiple MOB vessels with a three-vessel case consisting of “follow-the-leader” control (i.e., any of the three vessels considered to be the leader) and “leaderless” control (i.e., each vessel is commanded to inertial coordinates corresponding with correct heading and relative spacing). The primary measure used was time to stabilize in formation. The scenarios investigated were station-keeping and aligning into the wind (with the latter scenario for optimal CTOL).

While these maneuvers address a (large) subset of the total DP functionality required for DP-interconnected MOBs and are part of our effort, they must be organized in an appropriately robust supervisory structure. This is described next.

3. HIERARCHICAL CONTROL ARCHITECTURE

The size, weight, and control requirements of the independent module MOB raises some complex control issues. To deal with mission handling and control as well as dynamic positioning of the platforms and safety issues, a three-layer software architecture that moves from discrete to continuous signals is conceived\(^7,8,9\). This is represented in Figure 2.

At the top, or the strategic or supervisory control layer, discrete commands are given to achieve high-level goals such as moving to a location, engaging/disengaging platforms, and assigning a coordinator/leader platform. Commands are issued in the form of a state machine type automaton that handles emergency control, remote off-site control, and planned control. The state machine formulation makes the architecture user-friendly, in that it reduces the apparent complexity of the system, allows for less repeats in the code, and provides more flexibility throughout the control architecture. This level can be decentralized if necessary.

The intermediate level is the tactical or maneuver coordination layer. It interfaces the high level commands with the actuator commands, and is MOB-based. It deals with navigation and generates maneuvering commands, such as cruising, station...
keeping, forming a runway, and disengaging. It also computes optimal real time trajectories and has fault diagnosis and management capabilities.

The lowest level is the execution or MOB stability and control level, and it deals with continuous signals, and interfaces directly with the platform hardware. It contains several DP algorithms, a thruster allocation scheme, and sensor data processing and monitoring for fault detection.

3.1. Top level: supervisory control layer

The supervisory control layer switches what controllers to use and what parameters they should run with according to high level commands. These commands can come from the commanding officers of the structure, be pre-programmed, or be remotely given. The supervisory layer also keeps track of failures on board and has some error handling logic embedded into it, but the reflexive handling of thruster failure is accomplished at the lowest (MOB stability and control) layer. Emergency commands would interrupt the execution of the supervisory control and would supersede all others. In the current operational concept of the MOB, these emergency commands would constitute manual intervention.

The supervisory control layer pertains to the discrete domain and is represented as a several interconnected state machines. One state machine represents the assembly, and each module has its own state machine. A preliminary version is shown in Figure 3, which shows transitions to four states – “disassembled”, “assembly: add a module”, “assembled (go to + align)”, and “disassembly: remove a module”.

![Figure 3. MOB State Machine](image)

3.2. Intermediate level: maneuver coordination layer (control of several dependent platforms)

The maneuver coordination layer is formed of different controllers that correspond to the different maneuvers. The specialized intermediate level maneuver controllers listed in Figure 4 below – merge, split, goto, align in wind, and collision avoidance – will act in logical sequences for connection of the platforms, movements of the assembled MOB, and disassembly of the MOB. (We are deferring work on the collision avoidance controller for now, with the explicit protocol being that this is will be accomplished by human intervention.)
All of these coordinated maneuvers are partly continuous and partly discrete. As an example, the discrete logic (denoted by the transition arrows) and the continuous actions (occurring within the circles) for the Merge Controller is presented below.

Figure 5 illustrates the state logic from a MOB 1 perspective, with a maneuver in which MOB 2 joins the already-merged two-platform MOB 1. There are three states, beginning with “leaderless operation” with transitions to checking for “available to join”, and a waiting state, “blocked waiting to join” which would be attained if, for example, a MOB 3 was joining MOB 1 from the other side.

Figure 6 illustrates the same maneuver but from the perspective of MOB 2. Starting with “leaderless operation”, it transitions to “dock” and eventually to three-platform leaderless operation only after passing through the checks and balances necessary to ensure that a safe docking can occur.
There are additional layers of logical complexity that will be tailored to the specific mission profile, and this is being developed, along with the full sequence of environmental disturbances.

3.3. Lowest layer: MOB stability and control level (control of independent platforms)

This layer is formed of three parts: motion and stability controllers, thruster allocation, and thruster control (with the latter being added later). Inputs to this level are a desired state of the MOB. Outputs of the layer are propeller speeds and motor currents for each of the on-board thruster. Inputs and outputs at each sub-level within this layer are shown in Figure 7.

Figure 6. Logic for the Merge Controller, MOB 2

Figure 7. Lowest Layer: MOB Stability and Control
Figure 8 illustrates where discrete maneuver controllers are contained within the control hierarchy. This ensemble of controllers is contained within “Motion and Stability Control” block, to include DP. The DP and Thruster Allocation levels are extensively described elsewhere; development is just underway for the other maneuvers.

![Motion and Stability Controllers Diagram]

**Figure 8. Motion and Stability Controllers**

### 4. DP EVALUATION

#### 4.1. Mission Profile

Setting up the appropriate mission and environmental characteristics are necessary up-front subtasks in evaluating the efficacy of DP. Environmental characteristics will include variations in wind speed and direction, current speed and direction, and wave characteristics and direction under five generically-described environmental conditions: max wind and waves, slow wind shift, sudden storm, internal wave, and high current.\(^1\)

The initial condition is illustrated by Figure 9. The MOB assembly will likely be three widely-spaced independent modules planning the coordinated execution of a join sequence. The first step would be to attain a “module-assembly-ready position”, with individual prescribed positions within 100 ft of a line and plus or minus 10-deg orientation on a line, and with an end to end spacing of 400 ft. From this, several independent modules will sequentially join to form a MOB.

![Independent Modules Preparing to Assemble]

**Figure 9. Independent Modules Preparing to Assemble**
The group then goes to and maintains a desired “global” position specified to within plus or minus 2.5 mi and orientation within plus or minus 10 degrees. The “local” relative sway and surge of adjacent module ends will be to within 20 ft, and relative yaw of adjacent modules to within 1 deg, all while aligning with shifting wind (which will be up to Sea State 6). To disengage, the MOB then breaks-up into individual platforms, and the modules return to independent operation; this is also shown in the figure. This entire sequence is illustrated in Figure 10.

![Figure 10. Station-keeping and Disassembly](image)

4.2. Simulation Process

Simulations are being run by a specialized MOB version of SHIFT, a hybrid systems programming language developed at California PATH. 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 grouped 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 numeric variables. These equations are based on numeric variables of this type and outputs of other types accessible through link variables. The transition structure of the hybrid automaton may involve synchronization of pairs or sets of components.

The system 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 synchronizations 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.

Because the evolution of the MOB mission profile contains these constructs, a MOB-specific version of SHIFT was developed to perform the evaluation of DP within the suggested mission profile. Environmental disturbances and ensuing wind, wave and current interactions is being developed for inclusion into MOB libraries for SHIFT.

An important feature of this version of SHIFT is the arrangement of components in a hierarchical manner consistent with the tenets of supervisory control described in Section 3 above. Hence, the simulation architecture consists of an organized structure of layers and the respective interfaces. The layers of the simulation architecture correspond to building blocks of the underlying physical and control models.

The simulation architecture is represented by Figure 11 and is generally described below.

The MOB SHIFT type (not explicitly represented in Figure 11) models the MOB. The MOB type is composed of two main entities (explicitly represented in Figure 11):

- **Supervisory Controller**: The SHIFT type that models the supervisory control of the MOB.
- **Platforms**: Set of MOBPlatform
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 physical and control configurations that vary in time. The structure of each set may vary with time, while 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.
- **DisturbanceProcessor**: represents transfer functions from environmental disturbances into perturbations to the dynamic model of the platform.
- **Parameters**: provides the parameters describing each platform.
- **Initialstate**: provides the initial state of all the variables for the simulation.
- **Summation unit**.
- **Physical_Device_Layer**: represents the sets 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 the Physical_Layer type 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 DisturbanceProcessor.
- Speed of currents. Generated by DisturbanceProcessor.
- Accelerations. Generated by DisturbanceProcessor.

The outputs are: position, speed and acceleration in body fixed and global coordinates.

The inputs to Disturbance Processor 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
The Physical_Device_Layer represents the sensors and actuators available in a specific MOB concept. The inputs are:
- Commands for each actuator (specified by the Device_Control_Layer).
- Position, speed and acceleration in body fixed and global 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:
- 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.

This concept allows for:
- Each MOB concept to be mapped onto a particular specification for each layer (SHIFT model).
- Each layer specification models different configurations for the same MOB concept. A particular configuration is instantiated at run-time, according to a configuration file (does not require the compilation of the SHIFT code). For example, the number and location of thrusters for a particular MOB concept may be changed without the recompilation of the SHIFT code.

Also, within this architecture users will:
- Provide libraries of supervisors, controllers, sensors and actuators (which are in development).
- Write the SHIFT specification, in terms of layer models, for each MOB concept (to include hierarchical control layers).
- Write configuration files for each MOB concept according to the evaluation framework specifications.

5. SUMMARY

We have described the DP problem, our control scheme (with special emphasis on the control architecture and how supervisory control would be applied), and the notional mission profile. We have also described the SHIFT-based evaluation would be executed with already-developed environmental disturbance models. To complete this work would entail building further intermediate and lower level controllers, then organizing them in the hierarchical arrangement described here. Finally, SHIFT-based evaluation of DP will be conducted.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support for this work by the Office of Naval Research under grant N000114-98-0744.
REFERENCES


