Proceedings of OMAE'01
20th INTERNATIONAL CONFERENCE on OFFSHORE MECHANICS and ARCTIC ENGINEERING
June 3-8, 2001, Rio de Janeiro, Rio, Brasil

OMAE01/OSU-5022

A SHIP MANEUVERING CONTROL FRAMEWORK

João Borges de Sousa *
Dept.of Electrical Engineering and Computer Science
University of California at Berkeley
275M Cory Hall, Berkeley, California 94720
Email: sousa@robotics.eecs.berkeley.edu

J. Karl Hedrick
Department of Mechanical Engineering
The University Of California at Berkeley
6143 Etcheverry Hall, Berkeley, CA 94720-1740
khedrick@me.berkeley.edu

Anouck Girard
Ocean Engineering Graduate Group
The University Of California at Berkeley
230 Bechtel Engineering Center #1708
Berkeley, California 94720-1708
anouck@eecs.berkeley.edu

William Webster
The University Of California at Berkeley
222 California Hall, Berkeley, California 94720-1500
wwebster@socrates.berkeley.edu

ABSTRACT

A new ship maneuvering control framework is presented. This framework entails a three layered control architecture, a principled approach to design and implementation within the architecture, and hybrid systems design techniques. The formal model is dynamic networks of hybrid automata. The control architecture is structured according to the principle of composition of ship motions from a minimal set of elemental maneuvers, that are designed and verified independently. These are the building blocks for complex maneuvers that are also formalized in this framework. The principled approach is based on distributed hybrid systems techniques, and spans design and implementation because it uses the same model for both. Hybrid systems control techniques are used to synthesize the elemental maneuvers and to design protocols to coordinate the execution of elemental maneuvers within a complex maneuver. The architecture is fault-tolerant by design since it uses verified maneuvers. New control problems are introduced in this context. One is that of safety – to design a controller that always ensures safety in the presence of disturbances. The case study of the Mobile Offshore Base, that motivated the approach, is used to illustrate some of the main concepts. The novelty of the approach stems from the application of concepts and theories from distributed hybrid systems to the problem of ship maneuvering.

Keywords: Ship maneuvers, Hybrid Systems, Differential games, Mobile Offshore Base (MOB), Control Architecture.

INTRODUCTION

In the last decade we have witnessed unprecedented interactions between technological developments in computing, communications and control, and the design and implementation of networked multi-vehicle systems (Varaiya et al., 2001). New technological developments have enabled engineers to design new systems, and in turn, the implementation of those systems has led to a better understanding of the underlying technological issues, and to the formulation of new theories. One such example is the theory of distributed hybrid systems that has led to the development of a body of technology and tools for simulation, analysis and design. This is the enabling technology for an increasing number of networked multi-vehicle systems (for cars, helicopters and submarines see respectively (Varaiya, 1993; Varaiya et al., 2001; de Sousa and Deshpande, 1997). Here we present a formal ship maneuvering approach formulated in the framework of distributed hybrid systems. The approach is targeted at making ship maneuvering more robust and autonomous in the future.

We envision two levels of automation, that of maneuvers and

*Address all correspondence to this author.

complete automation. From the control perspective, the main difficulties come from maneuver automation. The problem is that traditional control techniques, that form the current practice in ship control, are not adequate for this purpose. The literature is abundant in low level control techniques for ship applications. These techniques are mainly tailored to solve low level control problems [1] that are formulated in the framework of continuous time and differential equations. Maneuver automation entails the realm of logic, and of discrete event models interacting with differential equations that model ship dynamics. This is why we need new control techniques. For example, we need to capture maneuver logic, to express it mathematically and to check it for consistency, correctness and safety.

Our developments have been motivated by the problem of maneuver coordination for the individual modules that compose the Mobile Offshore Base (MOB) (Remmers et al., 1999). This application involves some of the major ingredients of the generic ship maneuvering control problem.

The novelties are the introduction of concepts and theories from distributed hybrid systems and the use of systems engineering principles for architectural design (IEEE, 1999).

We organize ship motions in terms of maneuvers. As we examine the general ship maneuvering problem, we are faced with a whole range of maneuvers, of which some are simpler, and others more complex. We strive to define a basic set of ”elemental maneuvers”, from which all the maneuvers can be derived. Once we have found a minimal set of elemental maneuvers, we can verify their design for safety. We then compose the complex ship maneuvers, using the elemental maneuvers as building blocks. This enables us to always design correct maneuvers, that is maneuvers that meet the given specifications (which may include safety and ensured results), even in the presence of disturbances.

The key elements of our framework are:

Composition of ship maneuvers from a small set of elemental maneuvers. For each particular problem, one can find a set of basic maneuvers that the ship must be able to perform. These are called elemental maneuvers, and all other maneuvers are composed from them.

Formal approach to ship maneuvering. We provide a formal definition of elemental maneuvers. These are formalized within a three-level control architecture.

Built-in fault-tolerance. By design, the control architecture is fault-tolerant since it uses correct elemental maneuvers.

The design of elemental maneuvers is incremental. First each elemental maneuver is developed and verified independently, then the more complex maneuvers are formed and tested. For each elemental maneuver, we first synthesize a least restrictive controller - one which provides us with ”sets”, or ”windows” of possible actions. When safety or final objectives are not at stake, we are free to select one of the controls. This is convenient, since in controls, precision is costly, yet the ocean environment cannot be quantified exactly. To handle safety and performance issues, we then complement the design with rules or controllers that incorporate some logic. Our controllers ensure that, if feasible, the maneuver's objective is indeed attained.

This paper is organized as follows. In the first section we discuss the general ship maneuvering problem. In the second section we address the automation of ship maneuvering and discuss control design and implementation issues. In the third section we present our approach and relate our developments to the state-of-the-art. In the last section we draw some conclusions and discuss future work.

THE PROBLEM

Problem structure

Let us consider what happens under manual ship control. The motion of a ship is controlled with a basic set of commands, such as course keeping, turning, maintaining cruise speed, etc.. These are basically actuator commands (e.g., rudder, engine). The captain of the ship uses these commands to achieve some goal, e.g., reaching a certain position at sea. Observe that these goals can be typified. In fact, at each point in time, a ship is performing one of the following operations, designated here as *basic maneuvers*:

Problem 1. *Reaching a region in bounded time, while moving inside some prescribed boundaries* [2]*. Examples: navigation in a narrow channel, way-point navigation.*

Problem 2. *Same as above with additional constraints on the final position, and/or on the final speed. Examples: docking, mating with connectors with minimal speed to ensure connection.*

Problem 3. *Station-keeping within a bounded region. Examples: dynamic positioning (DP), thruster assisted mooring.*

Problem 4. *Following a prescribed path, with bounded errors. Examples: laying underwater cables.*

Problem 5. *Following a prescribed trajectory*[3]*, with bounded errors. Examples: at sea refueling, where each ship follows a prescribed trajectory, and the spacing between ships must remain within bounds.*

To execute each of these *basic maneuvers* the captain exercises a control activity that results in a sequence of commands, such as cruise, turn left, etc.. Observe that this control activity follows a specific logical pattern for each basic maneuver, let us call it the *control logic*. Obviously, the same control logic may result

[1] The level of control that directly interfaces with actuators such as, auto-pilots, etc...

[2] The problem of reaching a point is not included above since it is subsumed by that of reaching a region, the one with practical relevance.

[3] A trajectory can be informally described as a path parameterized by time.

in different sequences of commands. Why? Because this control activity results from the interactions with other entities and the environment, that are not known in advance. For example, this control logic takes into consideration additional information and rules, and the exchange of information with other entities, such as harbor authorities and other ships. Relevant information includes ship capabilities, geography and bathymetry features, traffic, weather forecasts and current profiles. Rules include rules of way, design recommendations concerning, for example, ship behavior in waves, etc..

At the level of ship motion planning, ship motions are planned in terms of these *basic maneuvers.* For example, the plan to *go from A to B* may involve a sequence of way-point following maneuvers.

The problem of the coordinated operation of multiple ships can, in some cases, be phrased as a simple extension of the previous ones. For example, cargo transfer between two ships in motion can be formulated as a problem of station keeping in a suitable reference frame. We will not go through these on this paper.

Motivation: Mobile Offshore Base

We addressed the design of a ship control framework in the MOB project (de Sousa et al., 2000; Girard et al., 1999; Webster and Sousa, 1999), where we designed a control architecture to govern the automated operation of the MOB.

The Mobile Offshore Base concept. The MOB is a large, self-propelled, floating, pre-positioned ocean structure formed of three to five modules and reaching up to 1,500 meters in length (Remmers et al., 1999). In most concepts, the structure is made of several modules, which have to be kept tightly aligned under large environmental loads. The alignment is maintained through the use of thrusters, connectors, or a combination of both. The modules forming the MOB must be able to perform long-term station keeping at sea, in the presence of waves, winds and currents. This is usually referred to as Dynamic Positioning (DP) control.

Here, and in order to motivate our formal developments, we give a preview of the automated operation of the MOB. To fix ideas consider 3 modules M1, M2 and M3 that perform the following sequence of operations: 1) the 3 modules are completely independent and located far away from each other; 2) modules M1 and M3 are commanded to join module M2 to form a MOB – the *assembly maneuver*; 3) upon successful assembly, the MOB is required to perform DP while optimizing fuel consumption – *MOB DP maneuver*. For reasons of safety, the assembly operation requires the modules to join the MOB (initially just module M2) one at a time. The assembly order and the selection of the optimization strategy emanate from the commander of the MOB. Otherwise, the operation is fully automated.

One possible execution of the `assembly maneuver` is as follows. Module 2 enters the mode `available for MOB`. In this mode it waits from a `request to join` message from any other module. In this case, Module 1 sends this message first. Upon the reception of this message, Module 2, accepts the request, enters the mode `blocked for join`, and sends a message to Module 1 giving it permission to start the join operation (see Figure 1). In this mode, Module 2 performs a DP maneuver at its current location. Meanwhile, Module 3 sends a `request to join` message to Module 2. Since Module 2 is in the mode `blocked for join`, it replies with a negative answer.

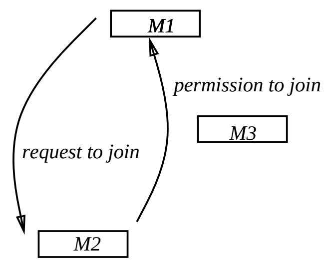

Figure 1. Excerpt of the join maneuver protocol

Upon the reception of the `permission to join` message, Module 1 enters an approach mode, where it executes an approach maneuver, that entails following an approach trajectory. The approach trajectory is calculated to bring Module 1 in a close vicinity, denoted V, of Module 2. While doing this Module 1 should keep a safety distance from Module 3 – in practice Module 1 is prevented from entering the section S (see Figure 2). Upon completion of this approach, Module 1 sends a `in vicinity to dock` message to Module 2. Upon checking if all conditions for safety are met, Module 1 and 2 transition to the docking mode, where both modules are jointly commanded to perform a docking operation.

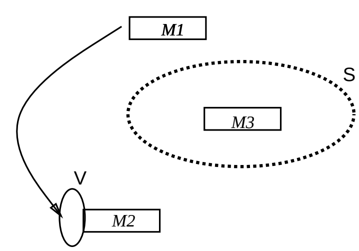

Figure 2. M1 joins M2

Upon successful docking Modules 1 and 2 enter the DP mode and are jointly commanded under a control strategy that minimizes fuel consumption – both modules perform a joint DP maneuver. In this mode, the ensemble is ready to accept more modules to join. This is what happens at a later stage with Module 3 (see Figure 3).

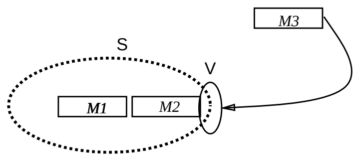

Figure 3. M3 joins MOB

The MOB is finally assembled. Then it is required to align with the dominant winds to minimize fuel consumption. In practice this means a transition to the `MOB DP maneuver`.

Towards automated operation

Introduction

The general problem of vehicle control automation is not a trivial one. Ship control automation is not an exception. From the above, we envision two levels of automation: 1) automation of the *basic maneuvers* and, 2) complete automation, from planning to operations. Let us describe what is involved in ship control automation.

Low level control

In terms of control automation, the set of commands such as turn to port, etc., is mapped onto the low level controllers. From the control systems perspective, the main difficulties concerning low level ship control are:

Non-linear dynamics. The dynamic behavior of a vessel can be highly non-linear, the effects of some disturbances are not thoroughly understood and some of the ship parameters are time-varying, e.g., the mass varies with loads.

Complexity of ship motions in a seaway. Ship motions result from the superposition of high and low frequency components (Lewis, 1989; Faltinsen, 1990) . The high frequency components cannot, in general, be compensated for by actuators. However, it is possible to attenuate their effects by controlling the ship's attitude with respect to the disturbance field.

Availability of models. Models of ship handling performance in some environments are either too complicated or are just not available. Often there is no clear consensus over which models to use. Examples include models for motions in restricted waterways and for thruster-to-thruster interactions in multi-thruster vessels.

Measurement of disturbances. The better the estimate of the disturbances, the better the controller performance. The problem is that it is quite difficult to obtain even a rough estimate of some of the disturbances.

Actuator saturation. This is not a modeling problem. However, some control design techniques do not take this physical fact into consideration. The result is that, when the actuators saturate the ship is no longer under closed-loop control.

Control design and implementation issues

From the control perspective, the main difficulties for automating ship control come from the first level of automation, that of the elementary maneuvers. The problem is that traditional control techniques, that form the current practice in ship control, are not adequate for this purpose. These techniques are mainly tailored to solve the low level control problems.

The problem of design for the second level of automation is not trivial, but it relies on the services provided by the first level. Let us look at what is involved in the design of a basic maneuver controller to understand why we need new techniques.

Formal methods. The captain of a ship uses some control logic to command the execution of a basic maneuver. We need to be able to capture this logic, to express it mathematically and to check it for consistency and correctness. This is why we need formal methods from computer science (see (van Leeuwen, 1990)).

Models. The control problems at this level of automation are different from the low-level control problems discussed in the previous sub-section. The realm of the low level control problems is that of continuous time and differential equations. Here, we enter the realm of logic, and of discrete event models interacting with differential equations that model ship dynamics. The example of the MOB maneuvers illustrates these interactions: the motion of a module is determined by the reception of some message that, in turn, is triggered by the fact that another module crosses some safety boundary.

Automation is not mimicking manual operation. In fact this can be counter-productive. First, it may prove difficult to encode all the control logic in a compact representation. Second, even if this is done there is no assurance that there are no flaws in the control logic. Third, automation enables complex quantitative reasoning, that can be used for optimizing operations or to implement complex control routines that ensure safe operations even in the presence of disturbances.

Control issues. Obviously the realm of interactions between continuous time dynamic models and discrete events requires the consideration of advanced control techniques. Let us discuss what is required from control design.

Communications and control. Control actions depend on interactions with other systems. Consider for example, the join operation from the previous section, where Module 1, which is not required to be stopped, asks Module 2 to join. What happens if

Module 2 does not reply in a reasonable time (there is a communication problem) or if Module 1 does not block for operations upon receiving the request to join message? All these issues are solved with a protocol design that can be formally verified for correctness (see (Varaiya et al., 2001) for further details).

Logic based control. The actions of each module follow some control logic. This control logic is complicated because it not only involves discrete event behavior but also complex continuous dynamics and interactions with other modules. The traditional practice of `if-then-else` programming is no longer adequate. It is not required to an expert programmer to realize that the amalgamation of `if-then-else` statements is very difficult to verify and often leads to unpredictable behavior.

Safety and predictability. Two important requirements for automated operation are predictability and safety. The controller has to perform according to what it is expected to do while respecting safety constraints. Disturbances play against predictability and safety. The question is then how to design predictable and safe controllers. This entails designing not only control laws but also regions for safe operation.

Flexibility. There is not need to over-constrain ship motions by imposing artificial constraints that are not used in practice. This suggests the consideration of least restrictive controllers. Informally, a least restrictive controller is one that outputs sets of control settings - from which the commanded control setting can be selected[4].

Models for the coordination of the operation of several vehicles. Models of coordination govern the joint operation of multiple vehicles (see (de Sousa, 2001; Girard et al., 2001a)). The join maneuver exemplifies a model of coordination for two vehicles – in this case a decentralized one – since each module is controlled independently, although controllers exchange messages. The MOB DP maneuver, in turn exemplifies a different type of coordination. The controllers operate under one of the following strategies proposed for the MOB: leader-follower, leaderless etc..

Software implementation. In our experience, a lot of control code is designed and implemented using techniques that are not particularly adapted to this level of automation. The problems are: 1) expressiveness and, 2) tool integration. We need compact representations to express the models and relations described above, and we need to interface the code with tools that facilitate design and verification. We have addressed some of these issues in a companion paper (Girard et al., 2001b).

It is now clear that control design at this level requires 1) *formal models* that span the design and implementation process; 2) a framework where we can study the overall structure and properties of control design that are not appropriately addressed within the constituent modules – *a control architecture*; 3) *a principled approach* to design and implementation; 4) *new control techniques*. These are the key elements of our approach.

Formal methods and logic theories introduce the possibility to express concepts and properties that are of importance for ship automation. Examples of those include fairness, liveness and least restrictive control. However these concepts are not part of the collection of idioms, patterns and styles of organization that control engineers have developed over the last decades and that serves as a shared, semantically rich, vocabulary among them (Shaw and Garlan, 1996). This means that ship automation also requires a new description language.

THE APPROACH

Introduction

This section is organized in terms of the key elements of our approach. For the sake of clarity we opted to skip the mathematical details that can be found for example in (de Sousa, 2001; Varaiya et al., 2001).

Formal models

Our approach is based on recent developments in the theory of distributed hybrid systems. Informally, a distributed hybrid system is a collection of dynamic systems – each of which includes both continuous time activities and discrete-event features – that interact through the exchange of data and messages.

A hybrid automaton consists of control locations with edges between the control locations. The control locations are the vertices in a graph. A location is labeled with a differential inclusion, and every edge is labeled with a guard, and a jump and reset relation. A hybrid automaton is H = (L,D,E) where:

L is a set of control locations.
D : L $\longrightarrow$ *Inclusions* where D(l) is the differential inclusion at location l.
E $\subseteq$ *L X Guard X Jump X L* are the edges - an edge e = (l,g,j,m) $\in$ E is an edge from location from l to m with guard g and jump relation j.

The state of a Hybrid Automaton is a pair (l,x) where l is the control location and $x \in R^n$ is the continuous state. Hybrid automata are classified according to the characteristics of L, D and E (see (Puri, 1995)).

Researchers have used dynamic networks of hybrid automata (DNHA) to model dynamic interactions (Deshpande et al., 1997). This is the *formal model* that we consider in our approach. Informally, dynamic networks of hybrid automata allow for interacting automata to create and destroy links among themselves, and for the creation and destruction of automata. Formally, for each hybrid automaton, there are two types of interactions: 1) the differential inclusions, guards, jump and reset relations are also functions of variables from other automata, 2) exchange of events among automata. Obviously, interactions are mediated by means

[4]Steering of a car is a good example of a least restrictive controller: the driver selects one of several possible steering angles at each moment.

of communication. Hence, a model for dynamic interactions has to include a description of the mechanisms by which automata interact. At the level of software implementation, the mechanisms by which software modules interact are called models of computation, or semantic frameworks. Hence, models of computation provide the formal basis for dynamic interactions. The choice of the model of computation for a specific implementation dependents on the properties of the underlying problem domain (Lee and Sangiovanni-Vincentelli, 1996).

Key concepts

The overall approach stems from the consideration of following principle:

The general ship maneuvering problem can be composed from the solution of modular elemental maneuvers that can be designed for safety and consistency.

The consideration of this principle enables us to partition a complex problem into a number of sub-problems that can be analyzed independently. This organization also introduces structure to the problem thus providing the structuring principle for the control architecture.

The motivation for the organization of ship maneuvering as the composition of elemental maneuvers comes from our work on the MOB project (de Sousa et al., 2000) that, in turn, was inspired by the PATH architecture (Varaiya and Shladover, 1991). The work of Varaiya was also quite influential in terms of architecture design (Varaiya, 1997). For related work on the coordinated motion of cars, helicopters and submarines see (Varaiya, 1993; Varaiya et al., 2001; de Sousa and Deshpande, 1997)

All ship motions must be composed from elemental maneuvers. This means that the first design step is to find one such a set. There are several ways to do this. We consider systems engineering principles to list and classify practical maneuvers, and concepts from hybrid control systems to group them, or to further refine these maneuvers. For example dynamic positioning (DP) and thruster assisted mooring may fall within the same category. From the control perspective, each elemental maneuver defines a category of problems that share the same structure, and the same solution method. This uniform representation allows for a compact specification and easy understanding. Moreover, it facilitates the corresponding design problem. We used these principles to define the taxonomy of elemental maneuvers presented in the *Requirements* section.

It may happen that an elemental maneuver is not enough to describe the actions involved in the execution of a practical maneuver at sea – the case with maneuvers involving more than one ship. This is where the composition of maneuvers enters the picture, and where we introduce the notion of complex maneuver. For example, the *assembly maneuver* for the MOB entailed the parallel execution of elemental maneuvers for each of the modules. The joint execution of these maneuvers is coordinated through the exchange of specific patterns of messages, that implement a cooperation protocol.

It is now clear that the elemental and complex maneuvers constitute the primitives for ship motion planning. At this level, we abstract from the details of rudder or engine commands, while maintaining an adequate level of description.

Each elemental maneuver is characterized by 1) An objective, 2) Hard constraints (those that cannot be violated), 3) Soft constraints (those that can be negotiated under special circumstances). For control design this characterization is complemented with: 1) dynamic models, 2) rules, 3) information sets (relevant information). By putting all these pieces together with a controller we get an:

Definition 1 (Elemental maneuver). [5] *Prototypical solution to a class of ship motion problems that cannot be obtained from the composition of other maneuvers. It is characterized by:*

1. *Objective.*
2. *Hard and soft constraints.*
3. *Information sets.*
4. *Dynamic model.*
5. *Controller.*

Informally, an elemental maneuver is like a computer program. For example, GOTO(A;B;C), with parameters A=(10,10), B=(20,20), C=navigation channel linking A and B, defines a concrete control problem: go from location A to location B while navigating inside channel C.

For the sake of clarity we now introduce the following definitions:

Definition 2 (Safe maneuver). *A maneuver whose execution does not violate the specifications.*

Definition 3 (Correct maneuver design). *A maneuver design that ensures the execution of a maneuver is always safe, even in the presence of disturbances, as defined in the information sets.*

Definition 4 (Complex maneuver). *Prototypical solution to a class of ship motion problems characterized by:*

1. *Objective.*
2. *Hard and soft constraints.*
3. *Information sets.*
4. *Controller that coordinates the execution of the elemental maneuvers that compose this maneuver.*

Control architecture

The control architecture is at the level of design that addresses the overall structure and the properties of control systems,

[5] This is an overloading of word maneuver, as defined in the dictionary (ENC, 2000): ”a controlled change of course of a vehicle or a vessel”.

hence providing a focus for certain aspects of design and development that are not appropriately addressed within the constituent modules (see (Garlan, 1995) for a discussion on software architectures). This structure is formalized in terms of layers and the respective interfaces. We consider a three-level control architecture.

Regulation Layer— the automated vehicles. The dynamical models of the vehicles are given in terms of nonlinear ordinary differential equations. This level deals with continuous signals, and interfaces directly with the vehicle hardware. Control laws are given as vehicle state or observation feedback policies for controlling the vehicle dynamics. Control laws at this level correspond to low level commands such as course keeping, turning, etc..

Maneuver Layer— control and observation subsystems responsible for safe execution of maneuvers – the first level of automation. The supervisory layer commands the execution of elemental and complex maneuvers according to the motion plan. Interactions with the regulation layer are mediated by the elemental maneuvers. Each elementary maneuver sends low-level commands to the regulation layer and receives events concerning their completion or failure. Elemental maneuver control is given in terms of hybrid automata. Complex maneuver control is formed of a control law, and a protocol that is used to coordinate the elemental maneuvers involved its the execution. The current design uses protocols in the form of finite state machines.

Supervisory Layer— control strategies that the ship follows in order to maximize safety and efficiency according to an operational plan – total automation. This layer supervises the execution of the motion plan.

Control design

Here we discuss control design for this architecture and related work. For reasons explained before we will concentrate this discussion on control synthesis for the maneuver layer. The literature is not abundant on ship control design at this level. The cause may be the novelty of distributed hybrid control techniques.

The generic problem of elemental maneuver control synthesis can be described as follows: given a dynamic system, or a collection of interacting dynamic systems, synthesize a controller so that the system(s) satisfies(y) the maneuver specifications. The type of maneuver specification dictates the type of control formulation. Examples include: 1) the problem of invariance – staying inside some region; 2) the problem of attaining a given target set while the trajectories of the system remain inside some other set; 3) the problem of optimizing some criteria; 4) the problem of stabilizing a system. Inherent to most of these control formulations is the problem of reach set computation – the set of all positions that the ship can reach from a given starting position. In fact, given the reach set, it is quite simple to solve most of these problems.

Reach set computation for hybrid systems has received considerable attention from the control systems community. For our developments we are interested in the approach proposed by Kurzhanskii and Varaiya (Kurzhanskii and Varaiya, 2000). They use dynamic programming techniques to describe reach sets and related problems of forward and backward reachability (Kurzhanskii and Varaiya, 2000). These problems are formulated as optimization problems that are solved through the Hamilton-Jacobi-Bellman equations. The reach sets are the level sets of the value function solutions to these equations.

The problem of controller synthesis and performance is highly dependent on the availability of good models (Lewis, 1989), (Faltinsen, 1990). The problem of devising models for ship handling performance in restricted waters is addressed in (Eda, 1986) and (Yeung and T., 1980).

The control formulations that we use fall within the categories of differential games and viability theory. Viability can be described as follows: given a dynamical system, a region K, and a set of initial conditions that lie within K, synthesize a control law that ensures the state of the system never leaves K (Aubin, 1991). The setting of differential games is that of a dynamic optimization problem where the control inputs are partitioned into two classes: 1) those available for controlling the system, 2) those available to the adversary or the disturbance. This setting extends that of deterministic optimal control to the case where a stochastic characterization of the disturbances is not available, preventing the formulation of a stochastic optimal control problem.

Let us discuss the design of controllers for an elemental maneuver in the framework of hybrid systems. This is done in three phases: 1) translation of the maneuver specifications into restrictions on the system's reachable sets of states; 2) formulation of a differential game of the appropriate type (example reaching a target set), and derivation of Hamilton-Jacobi-Bellman equations whose solutions describe the boundaries of the reachable sets and of the safe set – the set of all positions from which there is a controller that ensures the target is always reached; 3) synthesis of the hybrid controller from these equations. The hybrid controller consists of: 1) a specification of the safe set; 2) a least restrictive controller that assumes the form of a feedback control law for the continuous and discrete variables which guarantees that the hybrid system remains in the "safe subset" of the reachable set; 3) a controller that selects the final control setting from the set of control options given by the least restrictive controller. The least restrictive controller solves the problem of correctness while leaving room for other control considerations when safety (constraint violation) or the attainment of objectives are not at stake. Prior to execution the controller performs a feasibility check for a maneuver specification. Upon this check, it ensures correctness of the maneuver. It does this by generating a non-deterministic output – the set of feasible controls – when safety is not at stake. The second controller selects the commanded control setting from this set. This two-level control design allows for the integration of qualitative design techniques that may account for qualitative human decision-making.

Results from differential games and viability theory have been used to synthesize maneuvers and controllers for aircrafts. A game theoretic approach to the design of safe collision-avoidance maneuvers for air traffic management systems was proposed in (Lygeros et al., 1995). Results from viability theory were used in (Leitmann et al., 2000) to address the problem of aircraft take-off in windshear. The differential game formulation was also used in (Miloh and Pachter, 1989) to address the problem of modeling the encounter of two ships in a seaway from the dual points of view of collision-avoidance and pursuit-evasion maneuvers. The effect of speed-loss experienced by ships during a turn is incorporated into the model. The theory of differential games is used to establish the safe zone or the capture zone in collision avoidance or pursuit-evasion, respectively. The maneuvers are not integrated in a ship maneuvering control framework.

The literature is vast on applications of optimal control to the problem of prescribing optimal maneuvers for the motion of vehicles. For example, in (Tzeng, 1998), the steering control of a ship during a course-changing maneuver is formulated as a Bolza optimal control problem. The problem is solved via the sequential gradient-restoration algorithm. Nonlinear differential equations describing the yaw dynamics of a steering ship are employed as the differential constraints, and both amplitude and slew rate limits on the rudder are imposed. Two performance indices are minimized: one measures the time integral of the squared course deviation between the actual ship course and a target course; the other measures the time integral of the absolute course deviation. In a sequence of two papers, (Miele et al., 1999a; Miele et al., 1999b), optimal control methods are used to synthesize collision avoidance maneuvers with and without cooperation and course change maneuvers, under actuator saturation constraints. The course change maneuvers are closely related to problem 2) as defined in above. The problem of these approaches is that although optimal controls may exhibit some robustness in the presence of disturbances (Glad, 1987), there is no way of ensuring that trajectories of the system stay within a safety region. Maneuver design techniques are not limited to optimal control. For example, sliding mode controllers are used in (Zhang et al., 1998; Hedrick et al., 1999), backstepping approaches in (Grovlen and Fossen, 1996) and H_∞ control in (Hyakudome et al., 1999). Then again, it is not possible to guarantee safety by design. Some of these techniques do not take into consideration the problem of input saturation, that makes their application prone to exhibit unpredictable behavior.

For a brief discussion on protocol design for complex maneuvers see (Varaiya et al., 2001).

CONCLUSIONS

This paper reports an approach for maneuver design and control of ocean vehicles. The novelty of our approach stems from the consideration of systems engineering principles and of distributed hybrid systems techniques. The innovations are the introduction of safe elemental maneuvers in a three level coordination and control hierarchy, in order to manage complexity. We formalize the notion of elemental maneuver and synthesize safe elemental maneuvers using techniques from hybrid systems, differential games and viability theory.

Future work includes implementing a full set of safe elemental maneuvers and enriching the maneuver switching logic to accommodate faults and automated planning.

We envision using this approach to design and implement advanced control systems for new applications such as ship platooning in restricted waterways, cargo transfer between ships, thruster assisted mooring and automated docking.

ACKNOWLEDGMENT

The authors thank the reviewers for their comments and suggestions. This material is based upon work supported by the U.S. Office of Naval Research's MOB Program under grant N00014-98-1-0744. João Borges de Sousa was partially funded by Programa PRAXIS XXI - Medida 3.1b) (IES project) and by Ministério da Defesa, Portugal. Anouck Girard was partially funded by the Link Foundation.

REFERENCES

(2000). *Microsoft Encarta Encyclopedia Deluxe 2000*. Microsoft. CD-ROM.

Aubin, J.-P. (1991). *Viability theory*. Birkhauser.

de Sousa, J. B. (2001). *Coordination and control of interacting dynamic systems*. PhD thesis, Universidade do Porto.

de Sousa, J. B. and Deshpande, A. (1997). Real-time multiagent coordination using diadem: applications to automobile and submarine control. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 1769–74. IEEE.

de Sousa, J. B., Girard, A., and Hedrick, K. (2000). Realtime hybrid control of mobile offshore base scaled models. In *Proceedings of the American Control Conference*.

Deshpande, A., Gollu, A., and Semenzato, L. (1997). The shift programming language and run-time system for dynamic networks of hybrid automata. Technical Report UCB-ITS-PRR-97-7, California PATH.

Eda, H. (1986). A study of ship handling performance in restricted waters: Development and validation of computer simulation model. *Transactions SNAME*, 94.

Faltinsen, O. M. (1990). *Sea Loads on Ships and Offshore Structures*. Cambridge University Press.

Garlan, D. (1995). Research directions in software architecture. *ACM Computing Surveys*, 27(2):257–261.

Girard, A., de Sousa, J. B., and Hedrick, K. (2001a). Cooperative control in uncertain dynamic environments. In *submitted to European Control Conference 2001*.

Girard, A., de Sousa, J. B., Hedrick, K., and Webster, W. (2001b). Simulation environment design and implementation: an application to the mobile offshore base. In *submitted to OMAE 2001*.

Girard, A., Hedrick, K., and de Sousa, J. B. (1999). A hierarchical control architecture for mobile offshore bases. In *Proceedings of Third International Workshop on Very Large Floating Structures*, pages 447–456.

Glad, S. T. (1987). Robustness of nonlinear state feedback - a survey. *Automatica*, 23(4):425–35.

Grovlen, A. and Fossen, T. I. (1996). Nonlinear control of dynamic positioned ships using only position feedback: an observer backstepping approach. In *Proceedings of the 35th IEEE Conference on Decision and Control*, pages 1211–1217. IEEE.

Hedrick, K., Girard, A., and Kaku, B. (1999). A coordinated control methodology for the mobile offshore base. In *Proceedings of ISOPE*, pages 70–75.

Hyakudome, T., Nakamura, M., Kajiwara, H., and Koterayama, W. (1999). Experimental study on dynamic positioning control for semi-submersible platform. In *Proceedings of the 9th International Offshore and Polar Engineering Conference*, pages 76–82. ISOPE.

IEEE, editor (1999). *IEEE standard for application and management of the systems engineering process.* IEEE.

Kurzhanskii, A. B. and Varaiya, P. (2000). Ellipsoidal techniques for reachability analysis. In Lynch, N. and Krogh, B., editors, *Computation and control*, Lecture Notes in Computer Science, pages 202–214. Springer-Verlag.

Lee, E. and Sangiovanni-Vincentelli, A. (1996). Comparing models of computation, http://ptolemy.eecs.berkeley.edu/ eal.

Leitmann, G., Seube, N., and Moitie, R. (2000). Aircraft take-off in windshear: A viability approach. *Set-Valued Analysis*, 8:163–180.

Lewis, E., editor (1989). *Principles of Naval Architecture.* Society of Naval Architects and Marine Engineers. 2nd revision.

Lygeros, J., Godbole, D. N., and Sastry, S. (1995). A game theoretic approach to hybrid system design. Technical Report UCB/ERL M95/77, University of California, Berkeley. Electronics Research Laboratory.

Miele, A., Wang, T., Chao, S., and J.B.Dabney (1999a). Optimal control of a ship for collision avoidance maneuvers. *Journal of Optimization Theory and Applications*, 103(3):459–519.

Miele, A., Wang, T., Chao, S., and J.B.Dabney (1999b). Optimal control of a ship for course change and sidestep maneuvers. *Journal of Optimization Theory and Applications*, 103(2):259–82.

Miloh, T. and Pachter, M. (1989). Ship collision-avoidance and pursuit-evasion differential games with speed-loss in a turn. *Computers & Mathematics with Applications*, 18(1-3):77–100.

Puri, A. (1995). *Theory of hybrid systems and discrete event systems.* PhD thesis, University of California at Berkeley.

Remmers, G., Taylor, R., Palo, P., and Brackett, R. (1999). Mobile offshore base: A seabasing option. In *Proceedings of Third International Workshop on Very Large Floating Structures*, pages 1–7.

Shaw, M. and Garlan, D. (1996). Formulations and formalisms in software architecture. In van Leeuwen, J., editor, *Computer Science Today: Recent Trends and Developments*, pages 307–323. Springer-Verlag.

Tzeng, C. (1998). Optimal control of a ship for a course-changing maneuver. *Journal of Optimization Theory and Applications*, 97(2):281–97.

van Leeuwen, J., editor (1990). *Handbook of theoretical computer science.* Elsevier.

Varaiya, P. (1993). Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(3):195–207.

Varaiya, P. (1997). Towards a layered view of control. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 1187–90. IEEE.

Varaiya, P. and Shladover, S. E. (1991). Sketch of an ivhs systems architecture. In *Proceedings of the VNIS '91. Vehicle Navigation and Information Systems Conference*, pages 909–922. IEEE.

Varaiya, P., Simsek, T., and de Sousa, J. B. (2001). Communication and control in hybrid systems. In *accepted as a tutorial session for ACC 2001*.

Webster, W. and Sousa, J. (1999). Optimal allocation for multiple thrusters. In *Proceedings of the ISOPE*, pages 83–89.

Yeung, R. W. and T., T. W. (1980). Hydrodynamics interaction of ship with fixed obstacles. *Journal of ship research*, 24(1):50–59.

Zhang, R., Chen, Y., Sun, Z., Sun, F., and Xu, H. (1998). Path control of a surface ship in restricted waters using sliding mode. In *Proceedings of the 37th IEEE Conference on Decision & Control*, pages 3195–3200. IEEE.