Protocol-Based COLREGS Collision Avoidance Navigation Between Unmanned Marine Surface Craft

Michael R. Benjamin
NAVSEA Newport RI 02841
Mechanical Engineering , MIT
Cambridge MA 02139
Email: mikerb@csail.mit.edu

John J. Leonard
Dept. of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge MA 02139
Email: jleonard@mit.edu

Joseph A. Curcio
Dept. of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge MA 02139
Email: jcurcio@mit.edu

Paul M. Newman
Dept. of Engineering Science
Oxford University
Oxford, OX1 3PJ, UK
Email: pnewman@robots.ox.ac.uk

February 22, 2006

Abstract

This paper is concerned with the in-field autonomous operation of unmanned marine vehicles in accordance with convention for safe and proper collision avoidance as prescribed by the Coast Guard Collision Regulations (COLREGS). These rules are written to train and guide safe human operation of marine vehicles and are heavily dependent on human common sense in determining rule applicability as well as rule execution, especially when multiple rules apply simultaneously. To capture the flexibility exploited by humans, this work applies a novel method of multi-objective optimization, interval programming, in a behavior-based control framework for representing the navigation rules, as well as task behaviors, in a way that achieves simultaneous optimal satisfaction. We present experimental validation of this approach using multiple autonomous surface craft. This work represents the first in-field demonstration of multi-objective optimization applied to autonomous COLREGS-based marine vehicle navigation.

1 Introduction

1.1 Motivation

Mobile robotic platforms deployed in the marine environment offer substantial benefits to society while bringing a multitude of policy and legal challenges. Introducing mobile robotic vessels into navigable waterways presents the risk of collision with other vessels (both manned and unmanned), personal injury and property damage. Until policy, law and specifications evolve to address these issues, one can only speculate on the requirements imposed on developers, owners and operators of mobile robotic marine vehicles. However, an inspection of the relevant legal standards concerning safe operation of vessels in navigable waters reveals the likely need of owners, operators and programmers to abide by the current "rules of the road" given by the "International Regulations for Prevention of Collision at Sea", or the "COLREGS" (Commandant, 1999). It is likely that as the use of mobile robotics continues to proliferate within the marine environment a new legal framework will evolve to address the ramifications of ownership and operation of these assets. A prudent operator might take the stance that, until the law catches up with the operation of these vehicles, the smart move is to make the vehicles compliant with the existing standards applicable to safe navigation (Brown and Gaskell, 2000), (Brown and Gaskell, 2000).

1.2 Solution Framework

Although the COLREGS is a document suitable for guiding human behavior, it is not suitable for direct input into a vehicle control system. In practice, there are often multiple rules simultaneously in effect, and to varying degrees. This is particularly true in congested waters. In many situations there are also multiple distinct vehicle maneuvers that would satisfy a given rule. Humans are fairly good at dealing with conflicting rules and capitalizing on the flexibility of the written language, but these situations present the harder challenges for autonomous vehicle control.

To address this problem, we have used a novel method of multi-objective optimization, interval programming (IvP), (Benjamin, 2004), within a behavior-based architecture for capturing

COLREGS rules. Each COLREGS rule corresponds to a behavior that produces an objective function over the vehicle's decision, i.e., actuator, space. The objective functions capture the behavior prescribed by the COLREGS rule (in the peak areas of the function), but also capture its flexibility (in the non-peak areas). Each iteration of the vehicle control loop then involves the creation and solution of a multi-objective optimization problem, where each module contributes one function. This approach is suitable for building additional mission modules, on top of a COLREGS foundation where the mission modules also produce additional functions alongside the COLREGS modules. Results from simulation and results from in-field experiments with multiple autonomous surface craft are reported to validate these algorithms and architecture.

2 Background

2.1 Behavior-Based Control

In behavior-based systems, robot or vehicle control is the result of set of independent, specialized modules working together to choose appropriate vehicle actions. It can be viewed as an alternative to the traditional sense-plan-act control loop as shown in Figure 1, where decision-making and planning are performed on a single world model that is built up and maintained over time.
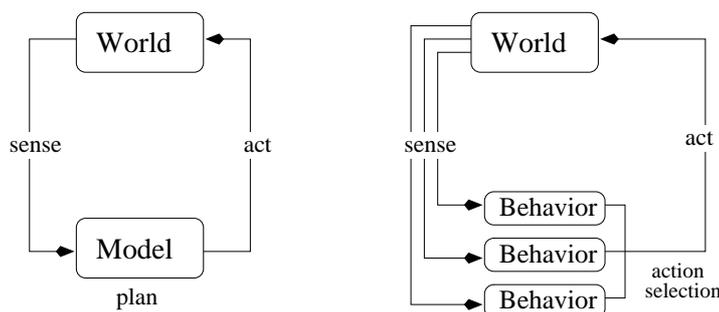

Figure 1: Behavior-based control differs from conventional control by composing overall vehicle behavior into distinct modules that are developed and operate largely in isolation, and coordinated through an action selection mechanism. In this case, action selection is in the form of a new multi-objective optimization technique to overcome known difficulties associated with behavior-based control.

Commonly cited virtues of behavior-based systems include: the ease of development of the

independent modules, the lack of a single complex world model, and the potential for a highly reactive vehicle with certain behaviors triggered by the appropriate events in a dynamic environment.

The origin of such systems is commonly attributed to Brooks' "subsumption architecture" in (Brooks, 1986). Since then, it has been used in a large variety of applications including: indoor robots, e.g., (Arkin, 1987), (Arkin, Carter, and Mackenzie, 1993), (Hoff and Bekey, 1995), (Lenser, Bruce, and Veloso, 2002), (Pirjanian, 1998), (Riekki, 1999), (Saffiotti, Ruspini, and Konolige, 1999), (Tunstel, 1995), (Veloso, Winner, Lenser, Bruce, and Balch, 2000), land vehicles, e.g., (Rosenblatt, 1997), planetary rovers, e.g., (Ju, Cui, and Cui, 2002), (Pirjanian, Huntsberger, and Schenker, 2001), (Singh et al., 2000), and marine vehicles, e.g., (Lee, Kwon, and Joh, 2004), (Benjamin, 2002b), (Bennet and Leonard, 2000), (Carreras, Batlle, and Ridao, 2000), (Kumar and Stover, 2001), (Rosenblatt, Williams, and Durrant-Whyte, 2002), (Williams, Newman, Dissanayake, Rosenblatt, and Durrant-Whyte, 2000). Action selection, as indicated in Figure 1, is the process of choosing a single action for execution, given the outputs of the behaviors. The "action space" is the set of all possible distinct actions. For example, all combinations of rotational and linear velocity for a robot, or all speed, heading and depth combinations for a marine vehicle.

2.2   Known Difficulties in Behavior-Based Control

The primary difficulty often associated with behavior-based control concerns action selection - namely how to ensure the chosen action really is in the best overall interest of the robot or vehicle. An action generally is a vector of values, one for each actuator being controlled. For example, the rotational and angular velocity for a land robot, or heading, speed and depth for a marine robot.

Generally there are two techniques used in practice. The simplest method is to pick (at every iteration of the control loop) a single behavior to have exclusive control of the vehicle. Some approaches, like (Bennet and Leonard, 2000), (Brooks, 1986), (Newman, 2003) assign a set of fixed priorities to behaviors, and conditions for their activation. The priorities do not change dynamically. In other implementations, like (Kumar and Stover, 2001), priorities may be determined

4

dynamically. Although using strict priority scheme is appealing due to its simplicity, it is problematic in applications where the outright ignoring of the "secondary" behaviors leads to gross vehicle inefficiency, as is the situation with task described in this work.

The other common form of action selection, known variably as "action averaging", "vector summation" etc., takes the output of each behavior in the form of a vector and uses the average numerical value as the action sent to the vehicle's actuators. Summation is typically weighted to reflect behavior priority. This method has been used effectively in a number of applications, (Arkin, 1987), (Arkin and Balch, 1997),(Balch and Arkin, 1998), (Carreras et al., 2000), (Khatib, 1985).

When the preferred actions of two distinct behaviors disagree, this approach rests on the idea that the alternative actions degrade in effectiveness in a manner depicted in Figure 2.
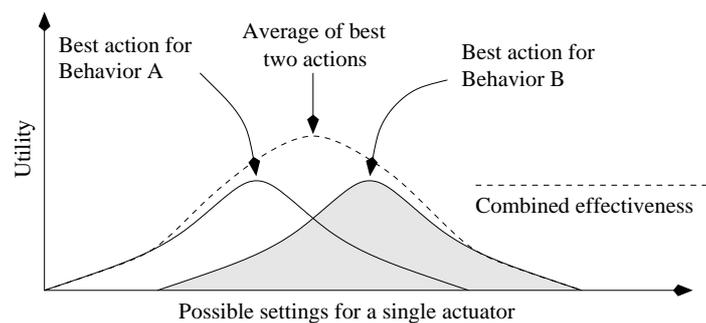

Figure 2: In action-averaging, each behavior outputs a *single* best action. The best action presumably is the most effective among alternative actions for that particular behavior. The effectiveness levels of alternative actions are rendered here only for illustration and do not participate in the action averaging process. When two behaviors are non-mutually exclusive and share common action choices with high levels of effectiveness, as shown here, then action averaging typically reflects an appropriate compromise between behaviors.

In such a case, the action, or actuator setting, in between the two individually preferred actions may indeed be the most effective action overall. However, action averaging is problematic in cases when alternative actions degrade in effectiveness in a manner depicted in Figure 3, where the numerical average does not represent an effective compromise between two behaviors that are, in effect, mutually exclusive.
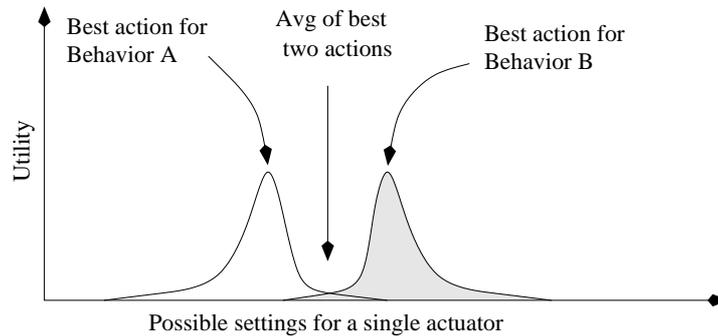
Figure 3: The average of the best action produced by two behaviors may have poor value for *both* behaviors. The chooser of the action is oblivious to the error since the behaviors output a single preferred action and do not communicate the underlying effectiveness of their alternatives, rendered here only for illustration. In this case, interests being pursued by the two behaviors are mutually exclusive, and the "compromise" is detrimental to both.

3 The "IvP" Architecture

3.1 Behavior-Based Control with Interval Programming

By using multi-objective optimization in action selection, behaviors produce an *objective function* rather than a single preferred action (Pirjanian, 1998), (Benjamin, 2002a) and (Rosenblatt, 1997). In the examples in Figs. 2 and 3, the objective functions are what distinguish opportunities for compromise. Note the overall preferred action in Figs. 2 and 3 are virtually the same despite the differences in utility of secondary alternatives.

An interval programming problem consists of a collection of IvP functions, each with an associated priority weighting. Each function typically corresponds to an aspiration of the decision maker, or autonomous agent, and maps each point in the decision space to a value that reflects the degree to which that decision supports the corresponding aspiration. The priority weightings reflect the degree to which the decision maker is willing to trade off achievement in one aspiration for another based on the overall context at the moment.

For a problem defined over a decision space with n decision variables $(x_1, \ldots, x_n)$, and having k objective functions $f_1(x_1, \ldots, x_n), \ldots, f_k(x_1, \ldots, x_n)$, with k priority weights $(w_1, \ldots, w_k)$, the general form is given by:

$$\begin{aligned} \text{maximize} \quad & w_1 f_1(x_1, \ldots, x_n) + \ldots + w_k f_k(x_1, \ldots, x_n) \\ \text{such that} \quad & f_i \text{ is an IvP piecewise defined function,} \\ & w_i \text{ is a positive number.} \end{aligned}$$

The k objective functions are effectively combined into a single objective function, which begs the question as to whether or not this constitutes "multi-objective" optimization. This term is applied here to discern a subclass of single-objective optimization problems where the single objective function to be optimized is composed of components that are themselves meaningful objective functions. By taking the sum of the set of contributing objective functions, the pitfall of action averaging shown in Figure 3 is avoided, but the benefit of combining behavior output shown in Figure 2 is captured.

There are two practical challenges in producing and using objective functions as the primary behavior output: (1) the method must be fast enough to accommodate the vehicle control loop, typically 1-20Hz. On each iteration new functions are created and a new problem solved. (2) if the method of solving the optimization problem depends on a strict type of objective function (such as linear or quadratic), then either certain behaviors cannot plug in, or such behaviors must produce an objective function of the right type that approximates its true objective function.

This work differs from (Pirjanian, 1998) and (Rosenblatt, 1997) in that the process of multi-objective optimization is not handled by a brute force evaluation of all possible decisions. The use of brute force is typically too slow and poorly scalable, and often, as in (Rosenblatt, 1997), is only practically achievable by decoupling control variable decisions, i.e., first deciding the best heading of a vehicle and then based on that heading, then deciding the best speed and so on for other variables. In this work the interval programming (IvP) model is used to implement multi-objective optimization over a coupled n-dimension action space, by letting each behavior contribute an IvP objective function over this space.

3.2   Interval Programming Functions

The interval programming model specifies (1) a scheme for representing functions of unlimited form and (2) a set of algorithms for finding a globally optimal solution. All functions are piecewise linearly defined, and thus are an *approximation* of a behavior's true underlying utility function. The piecewise functions need not use uniform shaped pieces between functions or even within a single function. The quality of the approximation is primarily determined by the number of pieces one chooses to generate as well as the algorithms for refining and allocating pieces to areas of the underlying function that are less amenable to local linear approximation. Furthermore, the function produced by a behavior need only be defined over the subset of the decision or action space relevant to the behavior.

Practically speaking, there are two IvP C++ libraries. The core library defines the data structures for representing IvP functions and problems, and the solution algorithms for solving syntactically legal instances of an IvP problem. The second library is more dynamic and contains a growing set of tools for casting or converting the utility functions native to a vehicle behavior into IvP functions. Each behavior produces an IvP function on each iteration of the control loop, so this process needs to be fast, and the behavior designer needs to be mindful about the tradeoff of function quality and CPU time. The existing library utilities for building an IvP function from an underlying function rely only on access to a function call for sampling or evaluating any point in the decision space for the underlying utility function. The underlying utility function is something unique to each vehicle behavior module. For the developer of the behavior, the utility function is the primary thing to determine and implement. To summarize, the behavior developer using the IvP model needs to address the following four aspects:

1. Define the underlying utility function correlating vehicle actions to the utility with respect to the overall behavior goal.

2. Provide a method (in our case in C++) for rapidly evaluating a given point in the function domain given certain current information about the world in which the vehicle is operating.

3. Make a decision about what is typically a good enough approximation of this utility function with an IvP function.

4. Determine a policy for how the priority of that behavior changes with respect to aspects in the world.

Examples of this process are given with behavior descriptions in Section 4.

3.3   Action Selection with Interval Programming

The action taken by the vehicle in each iteration of the control loop is the solution to the interval programming problem instance composed of the objective functions from each behavior. IvP search is over the weighted sum of individual functions and uses branch and bound to search through the combination space of pieces rather than the decision space of actions. The only error introduced is in the discrepancy between a behavior's true underlying utility function and the piecewise approximation produced to the solver. This error is preferable compared with the error of restricting all behaviors to a quadratic function for example. Furthermore, the search is much faster than brute force evaluation of the decision space, as done in (Rosenblatt, 1997), since each piece implicitly evaluates many points in the decision space. The larger the piece size, the greater it will outperform brute force in terms of speed, but at a cost of accuracy. The choice of piece size and distribution is a tradeoff between speed and accuracy. The solution speed is primarily correlated to the number of pieces contributed from each objective function, but as the number of dimensions or variables grow, more pieces are likely to be needed to achieve a satisfactory function approximation. The decision regarding function accuracy is a local decision to the behavior designer, who typically has insight into what is sufficient. The solver guarantees a globally optimal solution and this work validates that such search is feasible in a vehicle control loop of 4Hz on a 600MHz computer.

Like many branch and bound algorithms, the search can be greatly enhanced by seeding the problem with a good initial solution. The solution algorithms implemented for the IvP model

employ a few application-independent heuristics such as examining first the best decision in the highest weighted function. When using IvP for autonomous vehicle control, the initial solution used is always the best decision derived from the previous control cycle - typically what was a good decision a fraction of a second prior is also a pretty good (if not best) decision in the present unless something has changed in the world in the meanwhile. In fact, when something *does* change dramatically in the world, such as hitting a waypoint or a nearby vehicle changes trajectory, the solve time has been observed to be roughly 50% longer (but still comfortably under practical constraints). See (Benjamin, 2004) for more on IvP and search algorithms.

4   The Vehicle Helm and "Bread and Butter" Behaviors

A primary motivation for applying multi-objective optimization to the COLREGS navigation problem is that COLREGS behaviors serve to augment other behaviors without mutual design consideration. We present here two "bread and butter" behaviors sufficient for illustrating the subsequent description of the COLREGS behaviors. We also describe the decision space used in our particular helm and behaviors, and describe the notion of "closest point of approach", a key element of utility functions related to relative vehicle motion.

4.1   The Decision Space and Vehicle Helm

The *helm* is the module consisting of the behaviors and the optimization (action selection) engine. The action space in our helm consists of deciding the variables, heading (θ), speed (v), and time-on-leg (t). The latter is the "intended" duration of the chosen action. The helm is not committed to executing an action $\langle \theta, v, t \rangle$ for the intended time, but the time duration is used to further distinguish the utility of actions. The helm produces a tuple $\langle \theta, v, t \rangle$ on every iteration of the control loop, and the values of heading and speed are fed into PID control to produce rudder and thrust commands.

The helm, through the Global Positioning System (GPS), has access to its own position and trajectory (x, y, θ, v), and through wireless communication has access to the position, heading, and

speed of a given vehicle (x_b, y_b, θ_b, v_b). Each helm behavior has access to these variables if need be, and they comprise all the necessary input to the behaviors described below for this work.

Helm behaviors can be configured with initial priority weight values that may stay constant during the execution of a mission. A behavior may also use world state information garnered from the MOOSDB to calculate its own priority dynamically, perhaps in combination with its original configuration weight. A priority weight of zero effectively disables influence of a behavior. Although the use of objective functions is designed to coordinate multiple simultaneously active behaviors, helm behaviors can be easily conditioned on variable-value pairs in the MOOS database to run at the exclusion of other behaviors. Likewise, behaviors can produce variable-value pairs upon reaching a conclusion or milestone of significance to the behavior. In this way, a set of behaviors could be run in a plan-like sequence, or run in a layered relationship as originally described in (Brooks, 1986). If two behaviors are both simultaneously active, with competing or conflicting objective functions, resolution is achieved via the multi-objective optimization solver.

4.2   Closest Point of Approach

For COLREGS behaviors, an important quality of a candidate action $\langle \theta, v, t \rangle$, is the *closest point of approach (CPA)* between two vehicles during a candidate leg. A Behavior producing an objective function with CPA as a component of its utility function needs to perform many variations of this calculation on each new call to generate an IvP objective function. Thus, the algorithm with notes on efficiency measures are given here.

Our own current position is known and given by (x, y), and the other vehicle's current position and trajectory is given by (x_b, y_b, θ_b, v_b). To compute the CPA distance for a given $\langle \theta, v, t \rangle$, first the time t_{min} when the minimum distance between two vehicles occurs is computed. The distance between the two vehicles at the current time can by determined by the Pythagorean theorem. Generally, for any given time t (where the current time is $t = 0$), and assuming the other vehicle stays on a constant trajectory, the distance between the two vehicles for any chosen $\langle \theta, v, t \rangle$ is given by:

$$dist^2(\theta, v, t) = k_2 t^2 + k_1 t + k_0, \tag{1}$$

where

$$
\begin{aligned}
k_2 =\ & \cos^2(\theta)v^2 - 2\cos(\theta)v\cos(\theta_b)v_b + \cos^2(\theta_b)v_b^2 + \\
& \sin^2(\theta)v^2 - 2\sin(\theta)v\sin(\theta_b)v_b + \sin^2(\theta_b)v_b^2 \\
k_1 =\ & 2\cos(\theta)vy - 2\cos(\theta)vy_b - 2y\cos(\theta_b)v_b + \\
& 2\cos(\theta_b)v_b y_b + 2\sin(\theta)vx - 2\sin(\theta)vx_b - \\
& 2x\sin(\theta_b)v_b + 2\sin(\theta_b)v_b x_b \\
k_0 =\ & y^2 - 2yy_b + y_b^2 + x^2 - 2xx_b + x_b^2
\end{aligned}
$$

The stationary point is obtained by taking the first derivative with respect to t:

$$dist^2(\theta, v, t)' = 2k_2 t + k_1.$$

Since there is no "maximum" distance, this stationary point always represents the closest point of approach, and therefore:

$$t' = \frac{-k_1}{2k_2}.$$

The value of t_{min} may be in the past, i.e., less than zero, if the two vehicles are currently opening range. Or t_{min} may be well beyond t, the time length of the candidate maneuver $\langle \theta, v, t \rangle$. Therefore the value of t_{min} is clipped by $[0, t]$. Furthermore t_{min} is zero when the two vehicles have the same heading and speed (the only condition where k_2 is zero). The actual CPA value is then obtained by plugging t_{min} back into (1).

$$CPA(\theta, v, t) = \sqrt{k_2 t_{min}^2 + k_1 t_{min} + k_0}. \tag{2}$$

As mentioned before, this calculation is a common component in the underlying utility function for behaviors dealing with relative vehicle motion. A behavior, within a single iteration of

the control cycle, will perform a sequence of calculations on different $\langle \theta, v, t \rangle$ values. However, all calculations have the same values of current vehicle position (x, y), and current position and trajectory of the other vehicle $(x_b, y_b, \theta_b, v_b)$. To make this overall sequence of calculations faster, all terms in (1) comprised exclusively of x, y, x_b, y_b, θ_b, v_b are calculated once and cached for later calculations.

4.3   A Collision Avoidance Behavior

Each of the COLREGS behaviors described in the next section are also collision avoidance behaviors, each based on a particular COLREGS protocol regarding the relative position and trajectory between two vehicles. The collision avoidance behavior described here differs from the COLREGS behaviors only in that it doesn't care *how* collisions are avoided. Such a behavior may be used in an environment where the other vehicles are not following COLREGS protocol for example. (We use this behavior to describe general aspects of behavior creation as well as to provide a head start for later descriptions of the COLREGS behaviors.)

The underlying utility function, $f(\theta, v, t)$ utilized by this behavior, is based on the CPA distance for a candidate decision $\langle \theta, v, t \rangle$:

$$f(\theta, v, t) = g(CPA(\theta, v, t))$$

The utility varies linearly between a parameterizable "inner" distance and an "outer" distance. CPA distances lower than the inner-distance are treated as collisions, and values greater than the outer-distance have a plateau utility nominally set to 100. (Functions are normalized prior to the application of the priority weight, so actual utility ranges are insignificant). CPA distance in between the outer-distance and inner-distance degrade linearly, illustrated by the example in Figure 4.

The priority of the behavior is determined by the CPA distance of a hypothetical continuation of the current heading and speed out another n seconds. A simulation track is shown in Figure 5.
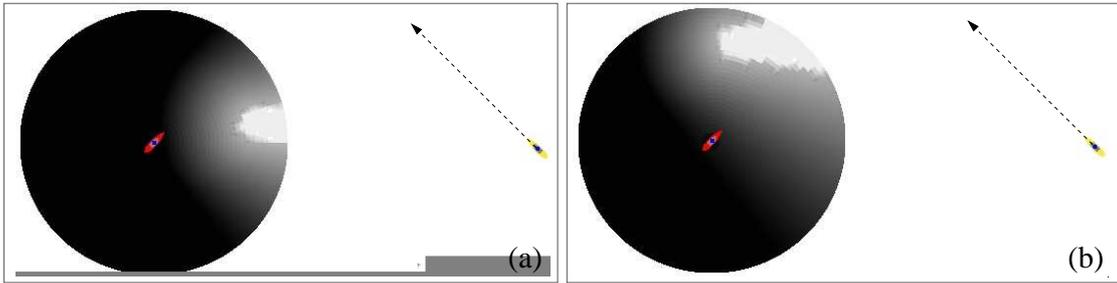
13

Figure 4: The objective functions produced by the AvoidCollision behavior for two situations. In both cases, the controlled vehicle has a top speed of 4 meters/second with the contact moving on the indicated heading. These are radial plots over heading and velocity. Darker colors represent more favorable actions, and larger radii on the plot indicate higher candidate speeds. The vehicles are 200 meters apart. CPA distances less than 10 meters are considered collisions (in white) and those greater than 75 meters are neutral (in black). Distances in between degrade linearly. In (a) the contact is moving at 3 m/s and in (b) the contact is moving at 5 m/s.

4.4 A Waypoint Behavior

The waypoint behavior is populated with a set of (x_i, y_i) waypoints, and has access to the vehicle's current position (x, y) via GPS. It ranks candidate legs $\langle \theta, v, t \rangle$ based on the proximity of the resulting position to the next waypoint. An example objective function is shown in Figure 6. The series of waypoint could be the result of either direct human entry at launch time, or the result of a your favorite path planning algorithm that runs either prior to launch or dynamically. This behavior also can be configured to perform a track-line by specifying a moving point on the track line between the current waypoints to steer toward, rather then steering directly toward the next waypoint. This behavior currently has no temporal aspect, but is rather guided solely by the deviation from the current calculated shortest path to the waypoint (or the track-line point).

5 The COLREGS Behaviors

There are nearly 40 rules that comprise the "COLREGS", nearly half of which concern lighting and sounds. We focus our attention on the four most challenging rules, from an autonomous navigation perspective, that cover "head-on" situations and "crossing" situation, rules 14-16. It is also worth noting rules 8(b), (d) which address collision avoidance generally (all excerpts are from
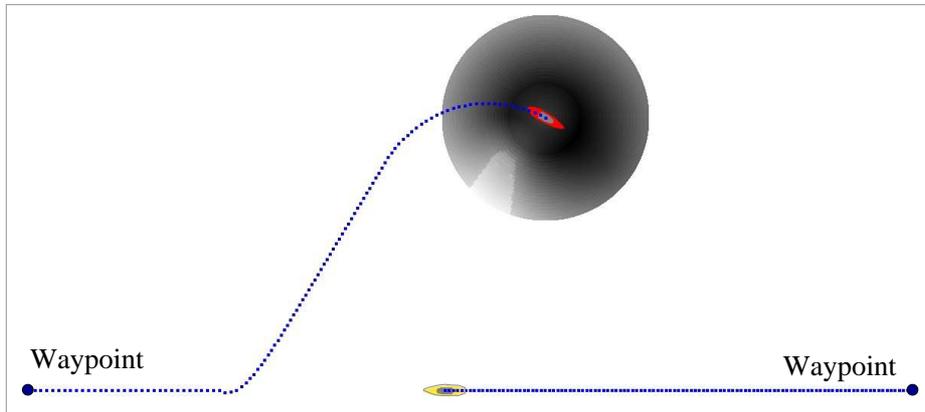
Figure 5: In simulation, the lefthand vehicle is guided by a waypoint and collision avoidance behavior to the point on the right. (Note this vehicle passes to the opposite side as would be prescribed by the COLREGS. Compare this trajectory with Figure 11.) The righthand vehicle is executing a waypoint behavior with no collision avoidance to the waypoint on the left. The function rendered represents the addition of the two objective functions at that point in time.
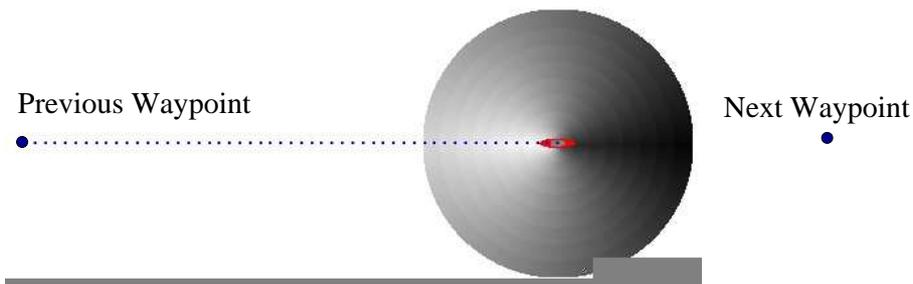

Figure 6: The objective function produced for the waypoint behavior rates decisions higher that bring the vehicle closer to the next waypoint and do not add detour distance. The utility drops linearly. This is a radial plot over heading and velocity. Darker shades represent higher utility. Typically about 600 linear pieces are used to represent this function.

(Commandant, 1999)):

Rule 8 :"Action to Avoid Collision"

> *(b) Any alteration of course and/or speed to avoid collision shall, if the circumstances of the case admit, be large enough to be readily apparent to another vessel observing visually or by radar; a succession of small alterations of course and/or speed should be avoided.*
> *(d) Action taken to avoid collision with another vessel shall be such as to result in passing at a safe distance. The effectiveness of the action shall be carefully checked until the other vessel is finally past and clear.*

15

This rule reveals a measure of the flexibility common in the rules, suitable for humans, but tricky for robots, such as "*large enough* to be readily apparent", and "*small* alterations of course". Generally the flexibility is found in both the *condition* of the rule and the *application* of the rule. Exploiting the latter is of paramount importance, since the rules need to at times co-exist with other rules as well as the efforts of the vehicle to complete its task.

5.1   The "Head-on" Behavior

The rule regarding two vessels approaching head-on is Rule 14 in (Commandant, 1999):

Rule 14 : "Head-on Situation"

 (a) Unless otherwise agreed, when two power-driven vessels are meeting on reciprocal or nearly reciprical courses so as to involve risk of collision each shall alter her course to starboard so that each shall pass on the port side of the other.

(b) Such a situation shall be deemed to exist when a vessel sees the other ahead or nearly ahead and by night she could see the mast-head lights of the other in a line or nearly in a line or both sidelights and by day she observes the corresponding aspect of the other vessel.

(c) When a vessel is in any doubt as to whether such a situation exists she shall assume that it does exist and act accordingly.

The objective function produced by this behavior is also based on the closest point of approach for a given candidate maneuver leg $\langle \theta, v, t \rangle$. The "head-on" condition referred to in the rule is interpreted to be in effect when the relative bearing between the two vehicles is within 15 degrees of the heading of the contact. To achieve the desired effect, the candidate heading is compared against the current relative bearing and starboard maneuvers are rated higher, and likewise lower for port maneuvers, as shown in Figure 7.

In addition, the behavior is given a range outside of which the priority of the behavior is zero and is inactive (see Figure 11(a).).
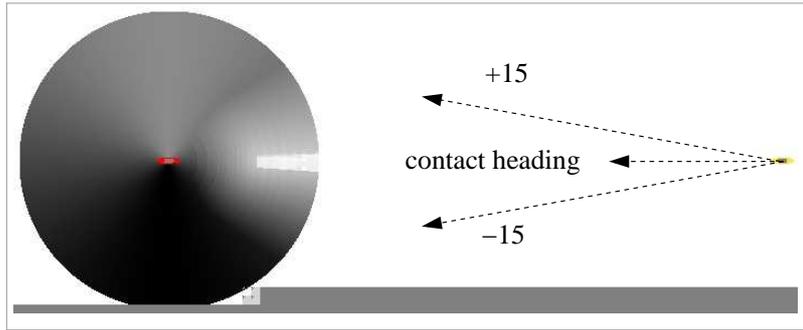
Figure 7: The "head-on" behavior produces objective functions based in part on the closest point of approach for a candidate maneuver and in part on a preference for starboard maneuvers passing the contact on the port side. Darker colors represent more favorable actions, and larger radii on the plot indicate higher candidate speeds. Compare against Figure 4(b) where maneuvers to either side of the contact are nearly equal in preference.

5.2 The "Crossing" Behaviors

COLREGS Rule 15 and 16 serve to define a "crossing" situation. These roles are depicted in Figure 8.
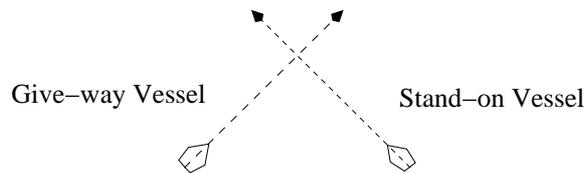
Figure 8: The Give-way vessel yields to the Stand-on vessel.

Rule 15 : "Crossing Situation"

(a) When two power-driven vessels are crossing so as to involve risk of collision, the vessel which has the other on her starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel.

Rule 16: "Action by Give-way Vessel"

Every vessel which is directed to keep out of the way of another vessel shall, so far as possible, take early and substantial action to keep well clear.

The objective function produced by this behavior also utilizes closest point of approach for a given candidate maneuver leg $\langle \theta, v, t \rangle$ in its objective function formulation. The "crossing" condi-

tion referred to in the rule is interpreted to be in effect when the relative bearing between the two vehicles is greater than 15 degrees of the heading of the contact, but less than 90 degrees. According to Rule 15, crossing ahead of the other vessel is to be avoided. To represent this preference in the objective function, a candidate leg, $\langle \theta, v, t \rangle$, is further evaluated to determine if it crosses ahead or behind the other vessel. The ranking of utility of an action is penalized further if it crosses ahead, as shown in Figure 9.
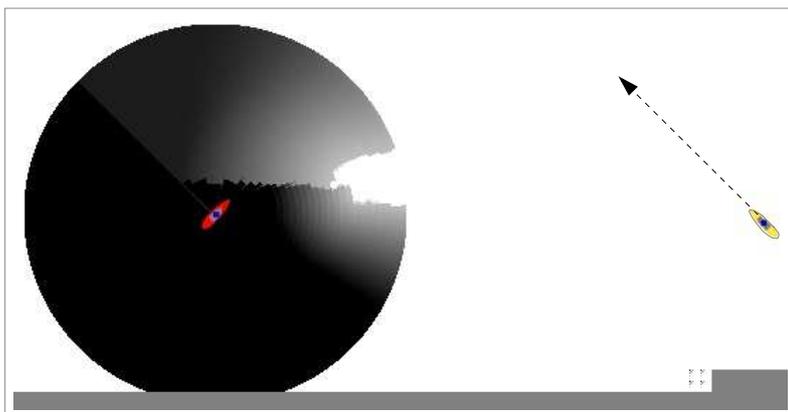

Figure 9: The "crossing" behavior produces objective functions based in part on the closest point of approach for a candidate maneuver and in part on a preference maneuvers that do not cross ahead of the other vessel. Darker colors represent more favorable actions, and larger radii on the plot indicate higher candidate speeds. (Compare with Figure 4(b).)

6 Experiments

Testing is done both in simulation and on two kayak-based autonomous surface crafts depicted in Figure 10. Each vehicle had access to a compass and Garmin 18 GPS, the latter with updates of 1Hz.

The GPS also provided the vehicle speed information, and at sufficiently high enough speed (> 0.5m/s), the GPS was preferred over the compass for heading measurements. Each vehicle communicated its position, heading and speed to the other vehicle at a rate of 4Hz, via a 802.11b wireless link. Each vehicle also had a unique ID that was known at launch time. For example, the Rule-14 collision avoidance behavior described in Figure 11 below, was parameterized explicitly
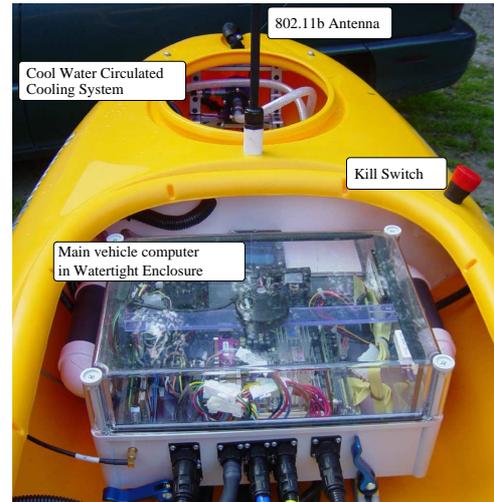
Figure 10: Two kayak-based autonomous surface craft for used for in-field experiments. Each had access to GPS and shared their current position and trajectory with the other.

with the ID of the other vehicle. A contact management module that automatically sorts incoming track data into distinct vehicle IDs is part of ongoing work. This feature we believe is orthogonal to testing the viability of the COLREGS behaviors.

Each vehicle is running MOOS, described in (Newman, 2003), which provides among other things the important capability of launching separate distinct on-board processes that communicate through a common database using sockets. Individual processes communicate in a subscribe and publish manner through this database. The database is a distinct process always running on-board. The helm that comprises the individual behaviors and the multi-objective optimization engine is a single process in a MOOS community of processes. In experiments reported here, the helm runs at 4Hz, and also contains the vehicle PID controllers. There is a PID controller for both "rudder" and "thrust". The helm passes control decisions in terms of "heading" and "speed" to the PID controllers.

Figure 11 shows a representative in-field experimental result that we have achieved using the behaviors and multi-objective optimization algorithms described in the previous section. This experiment was designed to test "Rule 14 (Head-on Collision)". The caption in the figure provides a detailed step-by-step account of how the correct behavior emerges based on the IvP optimized action selection strategy described above in Section 3. Each point on the plot represents a GPS

entry in the vehicle log file.

Figure 12 shows a representative experimental result regarding the Rule-15 (Crossing) be-havior. In this experiment, vehicle 2 is controlled by both the waypoint and COLREGS Rule-15 behavior. The waypoint behavior is configured to only penalize deviations from the shortest path to the next waypoint. It is not penalized for time delays. For this reason, the vehicle, when presented with a collision avoidance situation, simply slows down to let the situation resolve before resuming normal speed to the next waypoint.
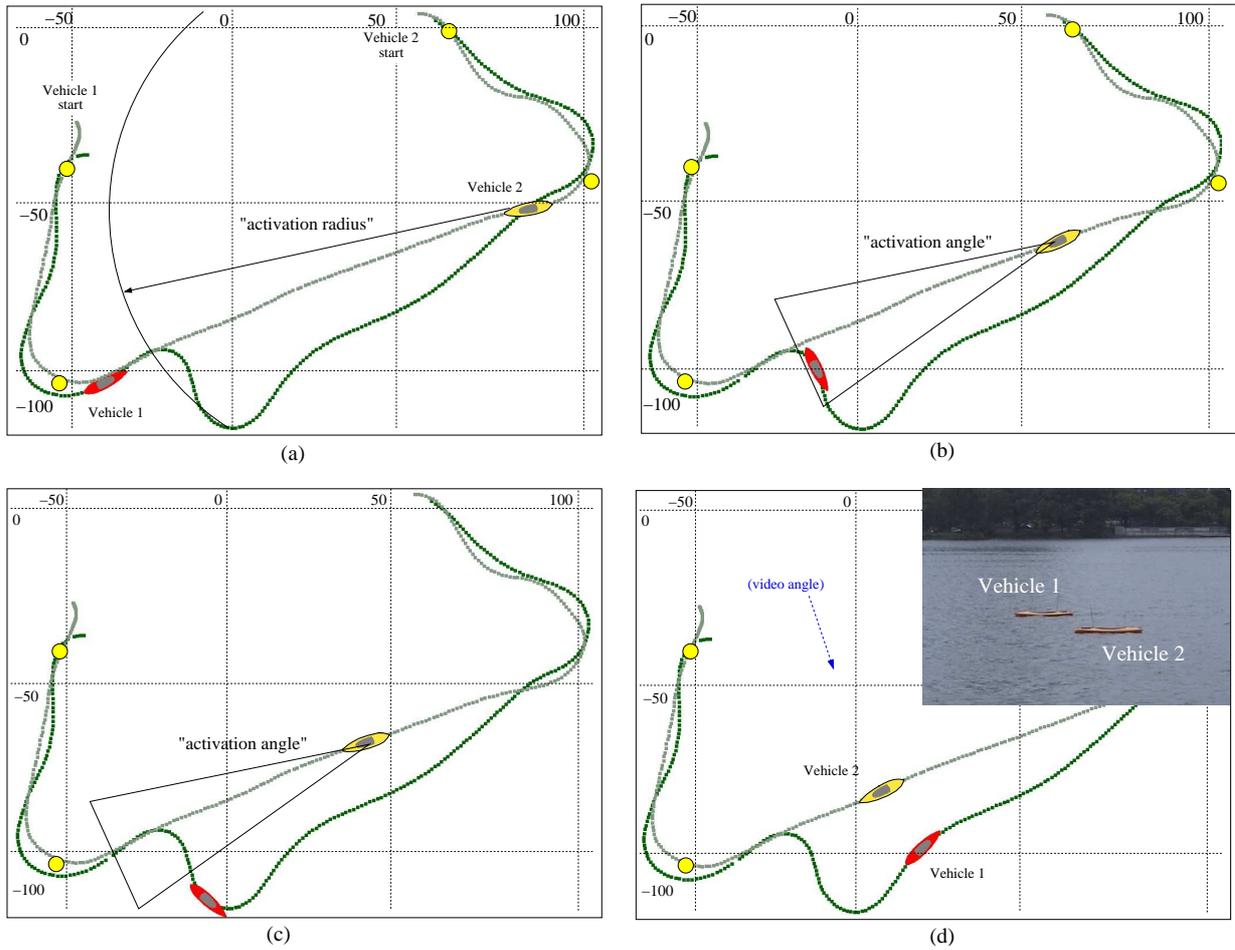
Figure 11: In-field experiments with two autonomous kayaks verifying the COLREGS "Head-on" Rule 14 behavior. Vehicle 1 and 2 are put on a head-on collision course through a series of waypoints. Vehicle 1 is utilizing a waypoint behavior and a "rule-14" behavior. Vehicle 2 is only using a waypoint behavior and does not make any attempt at collision avoidance with vehicle 1. In (a) the two vehicles are on a head-on collision course with vehicle 1 heading to waypoint $(105, -35)$, and vehicle 2 heading to waypoint $(-50, -110)$. Waypoints are shown in circles. In (a) only the waypoint behavior is active in vehicle 1 because vehicle 2 is still outside the activation range. In (b) vehicle 1 is within the activation range and within the activation angle specified to the rule-14 behavior and is thus making a starboard maneuver to avoid collision. In (c) vehicle 1 has just moved outside the activation angle and thus the rule-14 behavior becomes inactive, and the influence of the waypoint behavior begins to dominate again. In (d) vehicle 1 is proceeding uninhibited toward its destination. The image is from video shot during the experiment that produced the data shown here.
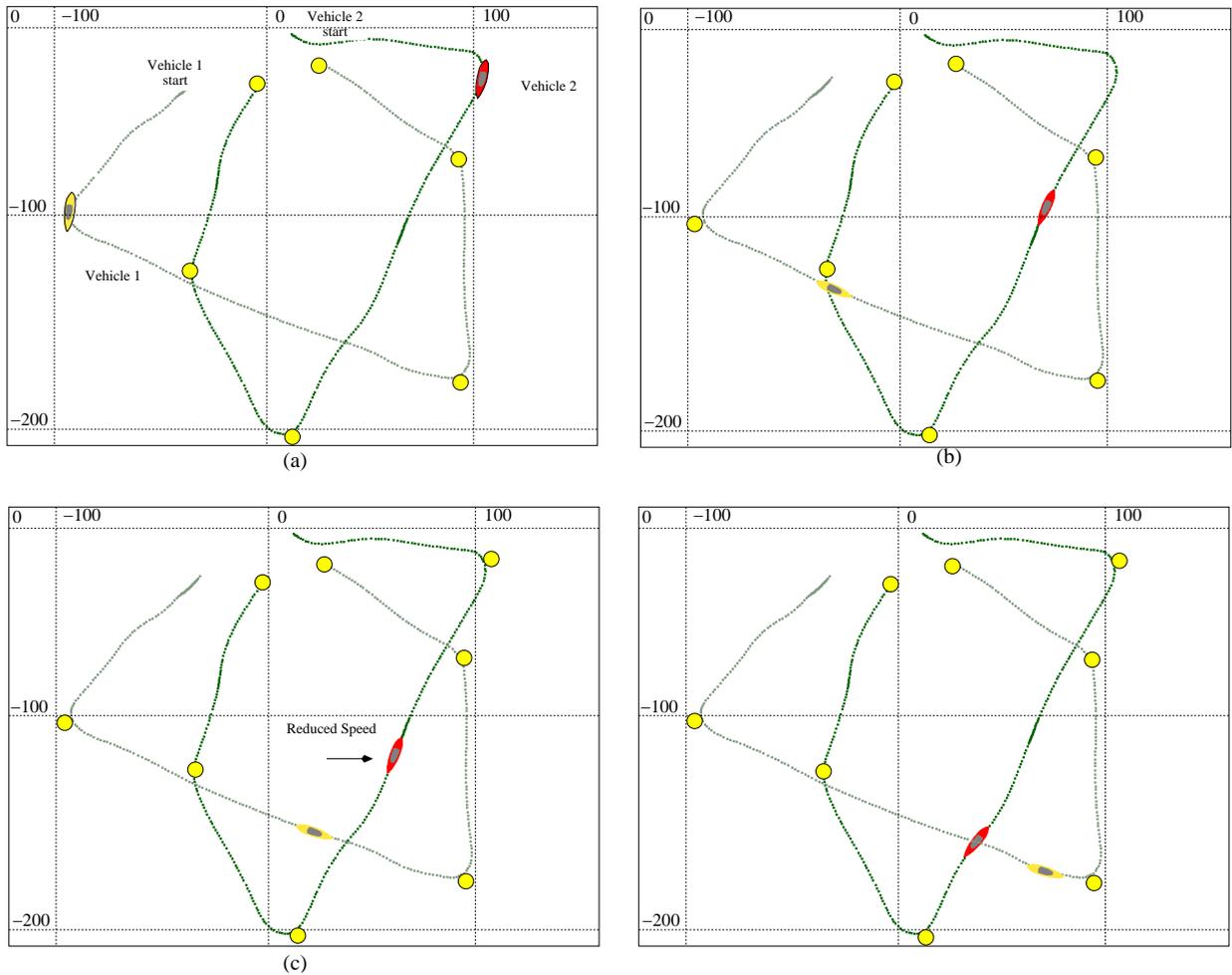
Figure 12: In-field experiments with two autonomous kayaks verifying the COLREGS "Crossing" Rule 15 behavior. Vehicle 1 and 2 are put on a collision course through a series of waypoints. Vehicle 2 is utilizing a waypoint behavior and a "rule-15" behavior. Vehicle 1 is only using a waypoint behavior and does not make any attempt at collision avoidance with vehicle 2. In (a) the two vehicles are on a collision course with vehicle 1 heading to waypoint $(100, -175)$, and vehicle 2 heading to waypoint $(0, -200)$. Waypoints are shown in circles. In (a) only the waypoint behavior is active in vehicle 2 because vehicle 1 is still outside the activation range. In (b) vehicle 1 is within the activation range specified to the rule-15 behavior and is thus begins to slow down to avoid collision. In (c) vehicle 1 has just progressed far enough so as to no longer be at risk for collision, and the rule-15 behavior becomes inactive, and the influence of the waypoint behavior begins to dominate again. In (d) vehicle 2 is proceeding uninhibited toward its destination.

7 Conclusion

This paper has investigated the problem of autonomous collision avoidance and navigation for unmanned marine surface craft. We have presented a novel method using IvP-based multi-objective optimization to coordinate distinct vehicle behaviors representing both task execution and established human protocol for safe navigation. This paper also provides, to our knowledge, the first ever demonstration of such a system on a physical marine platform.

An important goal in this work is to implement a technique capable of capturing the flexibility in the COLREGS rules. Not just the flexibility of *when* a rule is applied, but also *how* it is applied. We believe we have demonstrated such a technique by using objective functions to represent not only the preferred choice for a rule (the peak) but also the compromise choices (the off-peak areas of the function). A technique that allows the simultaneous influence of multiple behaviors also allows for tying the weight of that influence to the perceived developing situation.

The level of experimentation in this work has convinced us of the viability of this technique in terms of full end-to-end implementation of the algorithms on physical marine platforms in certain canonical collision risk situations. The current experimental results however do not yet suffice for claiming a "COLREGS compliant" system. A proof to support this claim may be as elusive as a proof that a human passing a written COLREGS test will never cause a collision. It is however an objective in this project to build a "COLREGS compliant" system and provide strong support for that claim.

To this end, our current work (Summer 2006) focuses on using six vehicles deployed to neighboring regions on the water with a shore-link that generates simultaneous randomly generated re-deployment commands that direct the vehicles to new deployment areas, causing vehicles to cross paths (in a wide variation of circumstances) to arrive at their new deployment area. We will log the trajectories for post analysis detection of collision, very near collision, near collision and so on. We can then compare performance between using (a) no collision avoidance, (b) non-protocol collision avoidance, and (c) COLREGS collision avoidance with different parameter settings. We expect to augment the on-water tests with simulation tests. While the on-water tests are more

realistic, the simulation tests can be run for many more hours with little effort.

In this work we relied on GPS on all vehicles for sharing very accurate information about position and trajectory over an 802.11b link. This allowed us to avoid fitting the vehicles with further sensors and implementing the algorithms to generate vehicle position from raw sensor data. At the conception of this project this was purely a decision of convenience to allow us to focus on decision making. In the meanwhile, Automatic Information Systems (AIS) have become cheaper and more prevalent on (manned) marine vehicles. These systems broadcast position, trajectory and vehicle identification to neighboring vehicles fitted with the proper receiver equipment. We are currently considering outfitting our platforms with such systems. We can envision the day when deployment of unmanned surface craft are limited to zones where all vehicles are AIS-compliant and COLREGS-compliant.

References

Ronald C. Arkin. Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behavior. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 264–271, Raleigh, NC, 1987.

Ronald C. Arkin and Tucker Balch. AuRA: Principles and Practice In Review. *Journal of Experimental and Theoretical Artificial Intelligence*, 9:175–189, 1997.

Ronald C. Arkin, William M. Carter, and Douglas C. Mackenzie. Active Avoidance: Escape and Dodging Behaviors for Reactive Control. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(1):175–192, 1993.

Tucker Balch and Ronald C. Arkin. Behavior-Based Formation Control for Multi-robot Teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, December 1998.

Michael R. Benjamin. *Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control*. PhD thesis, Brown University, Providence, RI, May 2002a.

Michael R. Benjamin. Multi-objective Autonomous Vehicle Navigation in the Presence of Cooperative and Adversarial Moving Contacts. In *OCEANS 2002*, Biloxi Mississippi, October 2002b.

Michael R. Benjamin. The Interval Programming Model for Multi-Objective Decision Making. Technical Report AIM-2004-021, Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA, September 2004.

Andrew A. Bennet and John J. Leonard. A Behavior-Based Approach to Adaptive Feature Detection and Following with Autonomous Underwater Vehicles. *IEEE Journal of Oceanic Engineering*, 25(2):213–226, April 2000.

Rodney A. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, April 1986.

E. D. Brown and N. J.J Gaskell. Report on the Law Relating to Autonomous Underwater Vehicles. Technical report, Society for Underwater Technology, 2000.

Marc Carreras, J. Batlle, and Pere Ridao. Reactive Control of an AUV Using Motor Schemas. In *International Conference on Quality Control, Automation and Robotics*, Cluj Napoca, Rumania, May 2000.

US Coast Guard Commandant. *International Regulations for Prevention of Collisions at Sea, 1972 (72 COLREGS)*. US Department of Transportation, US Coast Guard, August 1999. COMMANDANT INSTRUCTION M16672.2D.

Joel Hoff and George Bekey. An Architecture for Behavior Coordination Learning. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pages 2375–2380, Perth, Australia, November 1995.

He-Hua Ju, Ping-Yuan Cui, and Hu-Tao Cui. Autonomous Behavior Path Planning for Lunar Rover. *ACTA AUTO-MATICA SINICA*, 29(2):324–329, 2002.

Oussama Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 500–505, St. Louis, MO, 1985.

Ratnesh Kumar and James A. Stover. A Behavior-Based Intelligent Control Architecture with Application to Coordination of Multiple Underwater Vehicles. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Cybernetics*, 30(6):767–784, November 2001.

Sang-Min Lee, Kyung-Yub Kwon, and Joongseon Joh. A Fuzzy Logic for Autonomous Navigation of Marine Vehicle Satisfying COLREG Guidelines. *International Journal of Control, Automation, and Systems*, 2(2):171–181, June 2004.

Scott Lenser, James Bruce, and Manuela Veloso. A Modular Hierarchical Behavior-Based Architecture. In A. Birk, S. Coradeschi, and S. Takokoro, editors, *RoboCup-2001: The Fifth RoboCup Competitions and Conferences*. Springer Verlag, 2002.

Paul M. Newman. MOOS - A Mission Oriented Operating Suite. Technical Report OE2003-07, MIT Department of Ocean Engineering, 2003.

Paolo Pirjanian. *Multiple Objective Action Selection and Behavior Fusion*. PhD thesis, Aalborg University, 1998.

Paolo Pirjanian, Terry L. Huntsberger, and Paul S. Schenker. Development of CAMPOUT and its further applications to planetary rover operations: a multi-robot control architecture. In *Proceedings of SPIE Conference on Sensor Fusion and Decentralized Control in Robotic Systems IV*, Newton, MA, October 2001.

Jukka Riekki. *Reactive Task Execution of a Mobile Robot*. PhD thesis, Oulu University, 1999.

Julio K. Rosenblatt. *DAMN: A Distributed Architecture for Mobile Navigation*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1997.

Julio K. Rosenblatt, Stefan B. Williams, and Hugh Durrant-Whyte. Behavior-Based Control for Autonomous Underwater Exploration. *International Journal of Information Sciences*, 145(1-2):69–87, 2002.

Alessandro Saffiotti, Enrique H. Ruspini, and Kurt Konolige. Using Fuzzy Logic for Mobile Robot Control. In H. J. Zimmerman, editor, *Practical Applications of Fuzzy Technologies*, chapter 5, pages 185–206. Kluwer Academic Publishers, 1999.

Sanjiv Singh, Reid Simmons, Trey Smith, Anthony Stentz, Vandi Verma, Alex Yahja, and Kurt Schwehr. Recent Progress in Local and Global Traversability for Planetary Rovers. In *IEEE Conference on Robotics and Automation*, pages 1194–1200, San Francisco, CA, April 2000.

Edward Tunstel. Coordination of Distributed Fuzzy Behaviors in Mobile Robot Control. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 4009–4014, Vancouver, BC, Canada, October 1995.

Manuela M. Veloso, Elly Winner, Scott Lenser, James Bruce, and Tucker R. Balch. Vision-Servoed Localization and Behavior-Based Planning for an Autonomous Quadruped Legged Robot. In *Artificial Intelligence Planning Systems*, pages 387–394, April 2000.

Stefan B. Williams, Paul Newman, Gamini Dissanayake, Julio K. Rosenblatt, and Hugh Durrant-Whyte. A decoupled, distributed AUV control architecture. In *Proceedings of 31st International Symposium on Robotics*, pages 246–251, Montreal, Canada, 2000.