2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Trajectory Planning with Look-Ahead for Unmanned Sea Surface Vehicles to Handle Environmental Disturbances

Petr Svec, Maxim Schwartz, Atul Thakur, and Satyandra K. Gupta

Abstract— **We present a look-ahead based trajectory planning algorithm for computation of dynamically feasible trajectories for Unmanned Sea Surface Vehicles (USSV) operating in high seas states. The algorithm combines A* based heuristic search and locally bounded optimal planning under motion uncertainty using a variation of the minimax game-tree search. This allows the algorithm to compute trajectories that explicitly consider the possibility of the vehicle safely deviating from its original course due to the ocean waves within a specified look-ahead region. The algorithm can adapt its search based on the user-specified risk thresholds. Moreover, the algorithm produces a contingency plan as a part of the computed trajectory. We demonstrate the capabilities of the algorithm using simulations.**

INTRODUCTION

Planning of collision-free trajectories for Unmanned Sea Surface Vehicles (USSV) operating in high sea states presents a challenge due to significant environmental disturbances. The USSV movements are difficult to predict with sufficient precision because of the vehicle's interaction with the ocean waves. Therefore, an applied control action to the vehicle does not exactly lead to the desired motion. The variation in the resulting motion of the vehicle and its dynamics constraints makes the task of the trajectory planning particularly challenging in environments with dynamic obstacles. This typically includes planning in search, rescue, recovery, surveillance, or combat missions in which the unmanned vehicles may need to operate in the close vicinity of other dynamic objects to successfully fulfil their tasks. To be able to accomplish the tasks autonomously, they must have robust trajectory planning and high-level decision making capabilities to generate efficient and yet safe trajectories.

Let us illustrate two situations that can arise during navigation planning in an environment with high ocean disturbances (e.g., planning in sea state 4 for which the average wave height is defined to be between 1.25 to 2.5 m [1]). First, consider a situation in which the vehicle needs to quickly find a trajectory that is not overly conservative but still safe enough for traversal given the environmental conditions (see Fig. 1). Second, suppose that in some other situation the estimated collision risk is non-zero and the USSV needs to quickly decide which trajectory to follow to minimize this risk (see Fig. 2a).

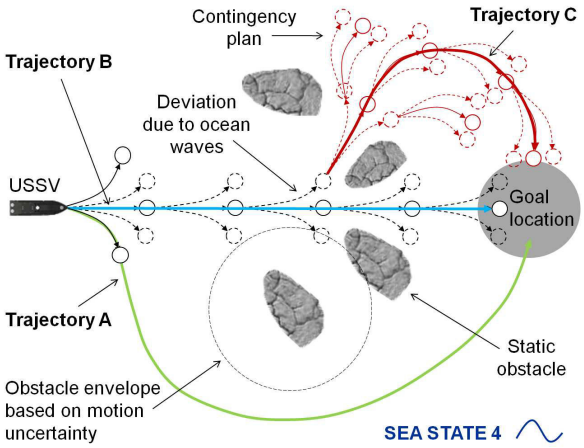

Fig. 1. Trajectory planning under motion uncertainty. The desired state transitions are shown as full arrows and the nondeterministic transitions are shown as dashed arrows. For clarity, we do not show all possible transitions.

As far as the first example is concerned, Fig. 1 shows the trajectories A, B, and C. The first trajectory A is computed by the overly conservative trajectory planning that puts large envelopes around each obstacle to prevent possible collisions. The second trajectory B is computed by a finite horizon look-ahead planner. This planner predicts near-term deviations of the vehicle from its nominal trajectory due to the ocean waves and uses these predictions for estimation of potential collisions. The planner assumes that the vehicle acts rationally i.e., it does its best to get back on the nominal trajectory or find a new collision-less trajectory C to the goal when the vehicle gets deviated from the trajectory B.

In the second example (see Fig. 2a), the USSV has to choose between executing action A or action B to approach its goal as fast as possible with the minimal probability of collision with dynamic obstacles. The predicted worst-case collision probability when the action A is executed is found to be 0.09, whereas execution of the action B leads to the collision probability of 0.001. Hence, by executing action A, the USSV has a higher probability of collision compared to executing action B. However, action A takes the USSV to the goal faster than the trajectory realized by the action B. Therefore, depending on the user-specified risk tolerance, the planner should decide to execute action A if the probability of collision 0.09 is acceptable, otherwise it executes action B.

To solve the above mentioned issues effectively for an

Petr Svec and Atul Thakur are with the Department of Mechanical Engineering, University of Maryland, College Park, Maryland 20742, USA, {petrsvec,athakur}@umd.edu

Maxim Schwartz is with the Energetics Technology Center, P. O. Box 601, La Plata, MD, USA, mschwartz@etcmd.org

Satyandra K. Gupta is with the Department of Mechanical Engineering and the Institute for Systems Research, University of Maryland, College Park, Maryland 20742, USA, skgupta@umd.edu

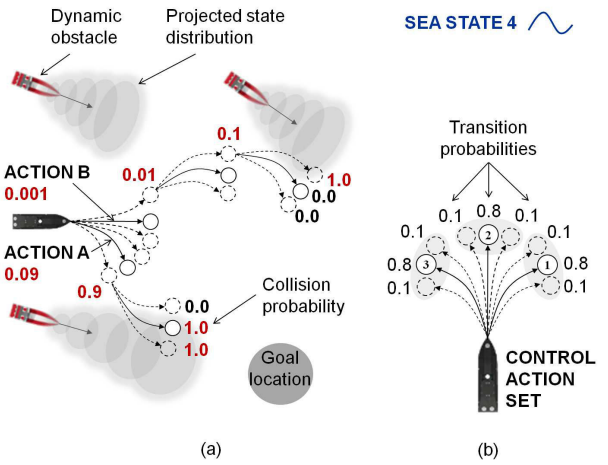

Fig. 2. Trajectory planning under motion uncertainty. (a) Minimizing collision probability. (b) A discretized set of dynamically feasible control actions and their corresponding nondeterministic transitions.

arbitrary model of the vehicle and sea state we developed a trajectory planner that can explicitly account for possible future deviations of the vehicle from its intended course to find a trajectory that is not overly conservative and still has a low probability of collision. The planner (1) utilizes the estimated uncertainty effects of the ocean waves on the motion of the vehicle, and (2) determines the probability of future collisions by projecting the state of the USSV under given control actions a few steps forward while considering possible nondeterministic state transitions.

The planner utilizes a fast heuristic based search algorithm to find a trajectory by incrementally searching in the space of candidate trajectories that consist of dynamically feasible control actions [2]. The algorithm combines A* based heuristic search and locally bounded optimal planning under uncertainty. It iteratively expands a tree of candidate trajectories according to the expected execution time and collision probability objectives incorporated into the cost function. Hence, a lower-risk candidate trajectory from two trajectories of the same expected execution time will be preferred for expansion during the search. In order to estimate the collision probability of a single trajectory segment, we employ a variation of the game tree minimax search [3]. In this type of search the USSV is a player executing control actions, whereas the nature, represented by the ocean waves, nondeterministically decides the outcomes of these actions. During the forward projection the planner builds a contingency plan. The contingency plan allows the vehicle to reactively make decisions to approach the target during on-line execution. If the contingency plan is not available for the current state of the vehicle, the planner recomputes the trajectory to acquire a new plan.

LITERATURE REVIEW

The trajectory planning under differential constraints [4] can be divided into (1) sampling based trajectory planning, (2) decoupled trajectory planning, (3) trajectory planning based on the Maneuver Automaton (MA), (4) mathematical programming, and (5) Model Predictive Control (MPC). The optimal planning under motion uncertainty in general entails solving a Markov Decision Process (MDP) using the dynamic programming (DP) algorithms [3]. However, since the state space of a planning problem under motion uncertainty is usually very large, most of the practical algorithms that have been developed [5], [6], [7] compute an optimal or close-to-optimal solution to the problem by running the value iteration over a carefully chosen subset of the state space.

In this paper, we focus on trajectory planning based on the Maneuver Automaton framework from which we use only maneuvers, not trims, similar to the lattice based planning in [2]. We extend this framework to consider the motion uncertainty. Recently, we utilized DP to compute an optimal policy over a lattice composed from dynamically feasible maneuvers and their corresponding nondeterministic transitions for a USSV operating in high sea states [8].

The work in [9] extends the MA framework by including the motion uncertainty directly into the cost function of a dynamic program. Our approach is similar to the approach developed by [10] in which the collision risk is directly incorporated into the cost function of the A* algorithm and is computed based on the trajectory-following error statistics. We, however, use look-ahead search in the discretized space of possible nondeterministic state transitions to estimate the collision risk and construct a high-level feedback in the form of a contingency plan. Our planner can handle arbitrarily large path-following errors (e.g., heading errors), can take care of non-Gaussian source of motion uncertainty, and is applicable to more general classes of robots.

USSV CONTROL ACTION SET GENERATION

The developed trajectory planner utilizes a discretized set of dynamically feasible control actions $U = \{u_1, \ldots, u_n\}$, or motion primitives [2] with a pre-set duration (see Fig. 2b). A control action is represented by forward velocity and heading angle profiles. We pre-computed each control action using a high-fidelity 6 DOF dynamics simulator of the USSV [11] (see the results section for details), so that it could be directly used in trajectory planning. The simulator is based on the potential flow theory [12]. The omnidirectional ocean waves are generated by superimposing wave components that are initialised using uniformly random phase lags. We used a PID controller for generating the control action set and for execution of control actions. In order to generate richer control actions, gradient-based optimization techniques [3] can also be used.

The resulting pose of the USSV starting from identical initial conditions varies when repeatedly executing the same action due to the presence of stochasticity in the ocean environment. Accordingly, each control action u is assigned a discrete transition probability distribution $P(x'|x, u)$ over a discretized set of possible resulting states to keep the number of nondeterministic transitions feasible for planning. The nondeterministic state transitions $S(u) = \{u_{s1}, \ldots, u_{sm}\}$ are defined for each control action u. The transition probability

distribution is computed using Monte Carlo simulations of the USSV dynamics model for 50 runs under sea state 4.

TRAJECTORY PLANNING ALGORITHM

A. Planning Problem Formulation

The planning task is to compute a trajectory τ that minimizes the time and the risk of collision during motion between a given initial x_{init} and a goal location x_{goal} in a stochastic environment with static and dynamic obstacles. The computed trajectory has to be dynamically feasible so that it can be directly followed by the USSV.

The search space is represented as a state lattice. The vehicle state is defined as $x = [x, y, \theta]^T$. The remaining dimensions such as roll, pitch, and angular velocities are suppressed to make planning feasible. We assume the velocity of the vehicle to be constant throughout the planning process. We discretize the action space U as well as the space of nondeterministic transitions $S(u)$ of each control action u.

B. Trajectory Search Algorithm Overview

The trajectory planner utilizes fast heuristic search algorithm (see Alg. 1) to find a trajectory by incrementally searching in the space of candidate trajectories using a discretized set of dynamically feasible control actions U and their corresponding nondeterministic state transitions S. The algorithm does not construct the entire lattice in advance, rather, it incrementally expands the lattice toward x_{goal} along the most promising direction. The algorithm combines A* based heuristic search and locally bounded planning under uncertainty using the game tree search [3]. It focuses its computation under uncertainty only into the most promising regions of the state space (driven by the heuristic function) as opposed to computation over the entire space. This substantially limits the computational requirements, which is important when planning in an environment with dynamic obstacles where fast replanning is a necessity.

C. Trajectory Cost Computation

The algorithm iteratively expands the nodes of the lattice with the least cost according to

$$f(x) = g(x) + h(x), \tag{1}$$

where $f(x)$ is the cost estimate of the trajectory starting from x_{init} to x_{goal} through x, $g(x)$ is the optimal cost-to-come of the trajectory from x_{init} to x, and $h(x)$ is the heuristic cost estimate of the trajectory between x and x_{goal}. The heuristic function $h(x)$ is used to reduce the total number of expanded nodes in the lattice, which is in accordance with the graph search A* algorithm that guarantees optimality of the resulting plan. In order to guarantee the optimality, the heuristic must be admissible. In other words, it cannot overestimate the true cost to the goal.

We define $p_{col,min}$ as the minimum trajectory collision risk the planner can tolerate, and $p_{col,max}$ as the maximum tolerable risk. During the search for the trajectory, if no trajectory is found with the collision risk less than $p_{col,max}$, the planner throws an exception to the higher-level decision making layers of the vehicle to decide what should be the next action. On the other hand, the planner does not include the collision probability of the trajectory into the cost if it falls below $p_{col,min}$.

The optimal cost-to-come is computed according to

$$g(x) = \begin{cases} T & \text{if } p_{col,\tau} \leq p_{col,min} \\ \infty & \text{if } p_{col,\tau} \geq p_{col,max} \\ T + wp_{col,\tau} & \text{otherwise,} \end{cases} \tag{2}$$

where $p_{col,\tau}$ is the overall probability of collision when following the trajectory τ, T is the total time needed to arrive to state x from x_{init} and is computed as $T = \sum_{k=1}^{K} l(u_k)$ over K planning stages, where $l(u_k)$ is the execution time of the control action $u_k \in U$. If the control action u takes the vehicle to an obstacle, then $l(u) = \infty$.

Algorithm 1 COMPUTETRAJECTORY$(x_{init}, X_{goal}, U, la_{max})$: A* algorithm for risk-based trajectory planning with look-ahead

Input: An initial state x_{init}, a goal set X_{goal} with a radius r around the goal state x_{goal}, a control action set U, and a maximum look-ahead level la_{max}.

Output: A trajectory τ.

```
Let Q be a sorted priority queue of states in ascending order according to the cost function f (see (1)).
Initialize Q ← {x_init}.
while Q not empty do
    x ← Q.FIRST() and mark x as closed.
    if x ∈ X_goal then
        return A trajectory τ generated by recursively tracing the predecessors of x up to x_init.
    end if
    for all u ∈ U do
        x' ← f(x, u)
        if x' not already closed then
            {p_chance, h_chance} ← LOOKAHEAD(x, u, U, 1, la_max)
            Use h_chance to compute a contingency plan for the state-action pair (x, u).
            Let p_col,(x,u) ← p_chance be the local collision probability for the state-action pair (x, u).
            Compute p_col,τ' for a new candidate trajectory τ' with the terminal node in x' using p_col,(x,u).
            Compute f_new(x') according to (1) by using the heuristic h_chance.
            if (x' ∉ Q) OR ((x' ∈ Q) AND (f_new(x') < f_old(x'))) then
                Set x' as the best successor state of x.
                Insert/Update x' into/in Q.
            end if
        end if
    end for
end while
return No suitable trajectory τ has been found for the given input.
```

Algorithm 2 LOOKAHEAD(x, u, U, la, la_{max})

Input: A state x, a control action u, a control action set U, a look-ahead level la, and a maximum look-ahead level la_{max}.

Output: An estimate p_{chance} of a possible future collision after execution of u in x, an expected heuristic cost h_{chance} from x to x_{goal}.

$x_{min} \leftarrow x$, $p_{chance} \leftarrow 0$, $h_{chance} \leftarrow 0$
Let $S(u)$ be the discretized set of nondeterministic state transitions for the control action u.
for all $u_s \in S(u)$ **do**
  $x'_{min} \leftarrow f(x_{min}, u_s)$
  Let p'_{min} be the collision probability of x'_{min}.
  if $la < la_{max}$ AND $p'_{min} = 0$ **then**
    $h'_{min} \leftarrow \infty$, $p'_{min} \leftarrow 1$
    for all $u \in U$ **do**
      $\{p''_{chance}, h''_{chance}\}$
      $\leftarrow$ LOOKAHEAD$(x'_{min}, u, U, la+1, la_{max})$
      if $p''_{chance} < p'_{min}$ OR $(p''_{chance} = p'_{min}$ AND $h''_{chance} < h'_{min})$ **then**
        $h'_{min} \leftarrow h''_{chance}$
        $p'_{min} \leftarrow p''_{chance}$
      end if
    end for
  else
    Compute the estimated heuristic cost h'_{min} from x'_{min} to x_{goal}.
  end if
  $p_{chance} \leftarrow p_{chance} + p(u_s)p'_{min}$
  Let $l(u_s)$ be the cost of u_s.
  $h_{chance} \leftarrow h_{chance} + p(u_s)(h'_{min} + l(u_s))$
end for
return $\{p_{chance}, h_{chance}\}$

The user-specified weight $w \geq 0$ explicitly increases the cost of the trajectory by the collision probability $p_{col,\tau}$. Alternatively, according to [10], one can define $g(x) = T/(1-p_{col,\tau})$ to balance the time and risk objectives without the need to tune the weight parameter manually. This way the expansion of nodes with a high collision probability from the priority queue Q is strongly discouraged.

D. Trajectory Collision Probability Computation

The cost function $g(x)$ includes not only the cost of the trajectory τ from x_{start} to x, but it may also incorporate the overall probability of collision $p_{col,\tau}$ when following that trajectory. This probability of collision is defined as

$$p_{col,\tau} = 1 - \prod_{k=1}^{K}(1 - p_{col,(x_k,u_k)}), \tag{3}$$

(similar to the definition in [10]), where $p_{col,x,u}$ is the local probability of collision for each segment k of the trajectory τ. Since the vehicle may get deviated from its intended course due to the ocean disturbances, $p_{col,x,u}$ gives an estimate of possible near-term collisions within the specified number of look-ahead steps, provided that the vehicle is a rational decision maker i.e., it does its best to quickly get back on its intended trajectory or find a new way to approach the goal. In the presence of dynamic obstacles, the future projected state of the vehicle is checked for collisions against the time-projected positions of the obstacles. One of the possible approaches for estimation of future positions of the obstacles by computing their time-parametrized trajectories as a sequence of Gaussian distributions can be found in [13].

E. Look-Ahead Search Tree Computation

The local collision probability $p_{col,x,u}$ for a given u executed from x is estimated by a systematic game tree search based on the Depth First Search (DFS) algorithm [3]. The search incrementally constructs a directed look-ahead tree by forward projecting the state x of the vehicle using control actions U and their corresponding nondeterministic state transitions S for a specified number of planning stages. The look-ahead tree consists of MIN and CHANCE nodes, and edges that represent the set of control actions U and nondeterministic state transitions S of the vehicle. The MIN nodes represent the decisions of the USSV, while the CHANCE nodes represent the nondeterministic decisions of nature, which in our case is made by the ocean waves. The presented game tree search is a variation of the minimax game tree search algorithm [3].

An example of the look-ahead search tree with the depth of 2 is depicted in Fig. 3. In this tree, a directed edge leading from a CHANCE node to a MIN node represents one of the possible nondeterministic state transitions $u_s \in S(u)$ of the vehicle with an assigned transition probability $p(u_s)$. A directed edge leading from a MIN node to a CHANCE node represents one of the vehicle's control actions $u \in U$. By executing a control action from a MIN node, the vehicle's state transitions to the CHANCE node that nondeterministically decides in which one of the possible discrete states the vehicle will end up.

The look-ahead search starts with execution of the action u in the state x to transition to a state x' through the first CHANCE node (see Fig. 3). Next, the algorithm recursively expands the tree all the way down to a given depth, bounded by the user-provided number of look-ahead steps. The states corresponding to the MIN leaves of the tree are assigned a heuristic cost estimate h of the time needed for the vehicle to get to x_{goal}. If the leaf of the tree and the associated arc intersect with an obstacle, the collision probability of the leaf node is set to 1.0. Otherwise, it is set to 0.0. By climbing the tree up to the root in the DFS fashion, the collision probability and expected heuristic values are propagated back to the root. The expected heuristic value h_{chance} of a CHANCE node is computed as the average over all nondeterministic outcomes h'_{min} according to $\sum_{u_s \in S(u)} p(u_s)(h'_{min} + l(u_s))$, where $l(u_s)$ is the cost of the nondeterministic transition u_s. Similarly, the collision probability of a CHANCE node is computed as $\sum_{u_s \in S(u)} p(u_s)p'_{min}$. The heuristic and collision probability values of a MIN node are taken as the minimum values of its children CHANCE nodes. The

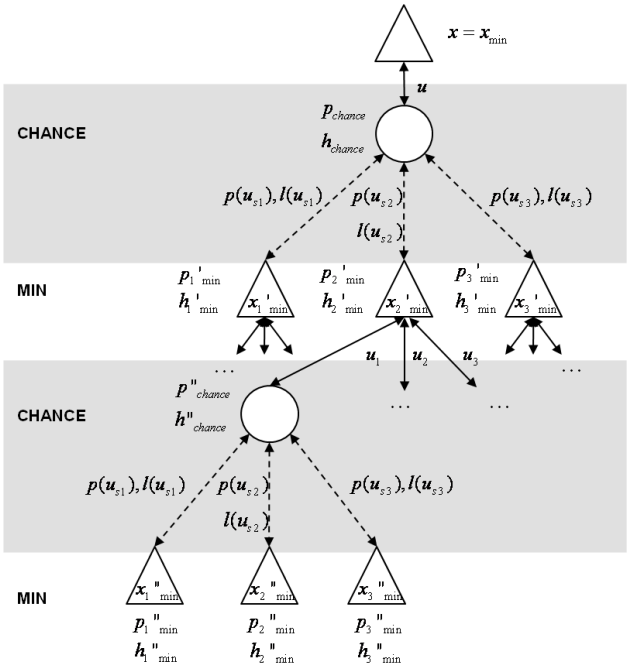

Fig. 3. Look-ahead search tree representation

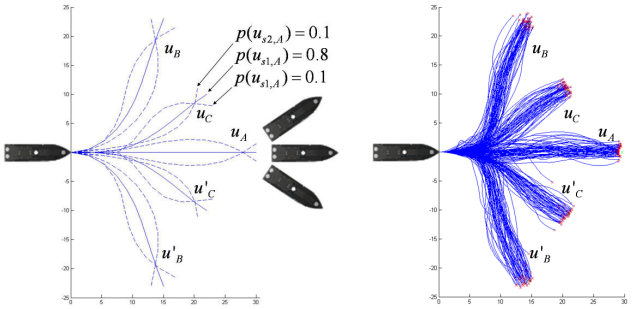

Fig. 4. (a) Control action set used in the experiment. Each control action u has assigned 3 nondeterministic transitions u_s varying in their final positions and orientations, and transition probabilities $p(u_s)$ extracted from the trajectory distribution. (b) Trajectory distribution for each control action u_A, u_B, u'_B, u_C, and u'_C generated using 50 simulation runs.

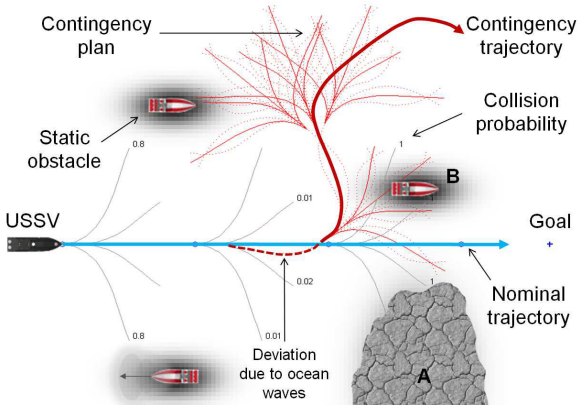

Fig. 5. Experimental case 1. The computed nominal trajectory (marked as blue) going through the narrow passage can be safely executed by the USSV due to the associated contingency plan (marked as red) which can reactively and robustly handle possible trajectory following exceptions.

main outcomes of the look-ahead search are $p_{col,(x,u)}$ and a heuristic estimate h_{chance} of the expected time to get to x_{goal} from x.

The look-ahead search expands an exponential number of states with respect to the depth of the tree. This, however, does not present a substantial problem for our application as the planner cannot look very far ahead due to the unpredictable movements of other obstacles; usually 2 to 3 look-aheads depending on the execution time of the specified control actions. However, the alpha-beta pruning [3] can be adapted to limit the number of the expanded nodes. Moreover, the tree stops expanding some of its branches once they align themselves with the nominal trajectory.

F. Contingency Plan

By following a node with the minimum heuristic value from the root node of the look-ahead search tree to its leaves, a sequence of local decisions is made for the vehicle to guide its movements towards the goal if it gets deviated from its original trajectory. In this way, the look-ahead search directly produces a contingency plan that is attached to the resulting trajectory. The contingency plan allows the vehicle to quickly make a decision by executing the best possible control action to move closer to the goal, either by getting back to the original trajectory or by finding a new promising direction to the goal. The plan is defined as a function $\pi : X \rightarrow U$ that produces an action $\pi(x) \in U$, for each $x \in X$, or node in the plan. At the representation level, the overall contingency plan is made by the expanded lattice itself and local contingency plans connected to all of its nodes.

EXPERIMENTAL RESULTS AND DISCUSSION

The control action set used in our experiment is shown in Fig. 4a. It consists of 5 control actions, where each action has assigned 3 nondeterministic state transitions with varying final orientations, positions, and transition probabilities ($p(u_s) = 0.8$ for the center nondeterministic transitions, $p(u_s) = 0.1$ for the remaining two). These nondeterministic transitions were manually extracted from a trajectory distribution generated using 50 Monte Carlo simulation runs for each control action as shown in Fig. 4b. In these simulation runs, each control action was executed under the sea state 4 consisting of 25 wave components. Given the velocity of the vehicle to be 3 m/s, the control actions u_A, u_B, and u_C and their symmetric counterparts have duration of 10, 9.6, and 8.3 seconds, respectively. The dimension of the boat is $12 \times 4 \times 4$ m. The planning space is represented as a gray scale image in which the obstacles are made by black 2D Gaussian regions.

In the designed computational experiments, the planner was supposed to find the shortest possible trajectory with a zero-collision probability between two given locations in an environment with obstacles under sea state 4. Fig. 5 shows

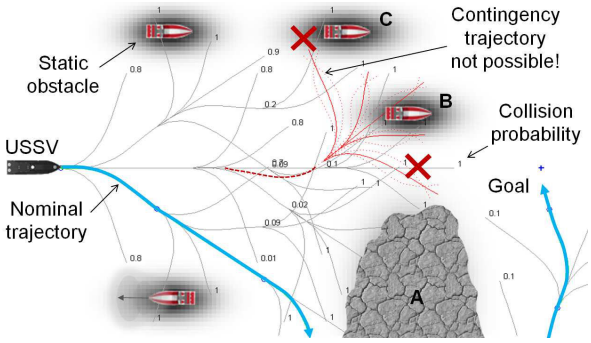

Fig. 6. Experimental case 2. In this case, the contingency plan that guarantees collision-less transitions for at least 3 look-ahead steps cannot be computed for the nominal trajectory from Fig. 5. The resulting nominal trajectory (marked as blue) has to lead around the obstacle A.

a situation in which the computed nominal trajectory leads through a narrow passage between the obstacles A and B with the expected execution time of 30 seconds in the best case. Although the vehicle may deviate from its course when following this trajectory, it will not collide with obstacles due to the precomputed contingency plan using which it can immediately execute a new safe trajectory leading it to the goal. The planner utilizes three-step look-ahead to verify that for each segment of the trajectory there is a contingency plan that does not lead to collisions. This particular example shows that in some situations it may not be the best strategy to immediately return to the original trajectory once deviated as it may cause a collision. Rather, it is safer to follow a new direction to the goal given by the contingency plan. In contrast, Fig. 6 shows a situation in which contingency planning is not possible for the nominal trajectory depicted in Fig. 5 due to the obstacle C blocking the possible passage. As a result, the only guaranteed collision-less trajectory to the goal leads around obstacle A with the expected execution time of 61 seconds.

In this experiment, the planner utilized three-step look-ahead and took 2 seconds on average to compute the shorter nominal trajectory (see Fig. 5) with the length of 90 m, whereas the computation time for the longer nominal trajectory (see Fig. 6) with the length of 183 m was 10 seconds on average on Intel(R) Core(TM)2 Duo CPU@2.66 GHz.

CONCLUSIONS AND FUTURE WORKS

In this paper we developed a trajectory search algorithm for planning under motion uncertainty in an environment with static and dynamic obstacles. The algorithm includes the risk of trajectory following directly into the cost function. It utilizes a multi-step look-ahead in the space of nondeterministic state transitions of the vehicle for each control action to compute this risk. The look-ahead search is implemented using a variation of the minimax algorithm. The algorithm is flexible due to the fact that it can utilize any dynamics model of the USSV and can handle any given sea state and non-Gaussian source of motion uncertainty. The product of the trajectory planning is not only the trajectory itself, but also a contingency plan for the USSV. This allows the USSV to effectively handle sudden deviations from its nominal trajectory. Using simulations, we have shown the capabilities of the planner to generate trajectories that are guaranteed to be safely executed even in high sea states. Currently, the number of look-ahead steps is limited as the tree grows exponentially with the number of required steps. Hence, in the future we want to increase the efficiency of the algorithm by intelligent pruning of this tree and reusing useful portions of the trajectory. The number of discrete control actions to be used is limited so we plan to enhance the algorithm to cope with this limitation. Also we plan to represent the state transition probability in a continuous form. Currently, the look-ahead search considers only the motion uncertainty but our aim is to handle the sensor uncertainty as well.

I. ACKNOWLEDGMENTS

This research has been supported by Office of Naval Research N00014-10-1-0585 and NSF Grant CMMI-0727380. Opinions expressed in this paper are those of the authors and do not necessarily reflect opinions of the sponsors.

REFERENCES

[1] T.I. Fossen, *Guidance and control of ocean vehicles*, Wiley, 1994.
[2] M. Pivtoraiko, R.A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009.
[3] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*, Prentice Hall, 2009.
[4] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1, pp. 65–100, 2010.
[5] M. Likhachev, G. Gordon, and S. Thrun, "Planning for Markov decision processes with sparse stochasticity," *Advances in Neural Information Processing Systems (NIPS)*, vol. 17, 2004.
[6] D. Ferguson and A. Stentz, "Focussed processing of MDPs for path planning," in *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04)*, 2004, pp. 310–317.
[7] S. Sanner, R. Goetschalckx, K. Driessens, and G. Shani, "Bayesian real-time dynamic programming," in *Proceedings of the 21st International Joint Conference on Artifical Intelligence*. Morgan Kaufmann Publishers Inc., 2009, pp. 1784–1789.
[8] A. Thakur, P. Svec, and S.K. Gupta, "Generation of state transition models using simulations for unmanned sea surface vehicle trajectory planning," in *ASME 2011 International Design Engineering Technical Conference (IDETC) & Computers and Information in Engineering Conference (CIE), Accepted for publication*, 2011.
[9] T. Schouwenaars, B. Mettler, E. Feron, and J.P. How, "Robust motion planning using a maneuver automaton with built-in uncertainties," in *Proceedings of the 2003 American Control Conference*. IEEE, 2003, vol. 3, pp. 2211–2216.
[10] M. Greytak and F. Hover, "Motion planning with an analytic risk cost for holonomic vehicles," in *Proceedings of the Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference CDC/CCC*, 2009, pp. 5655–5660.
[11] A. Thakur and S. K. Gupta, "A computational framework for real-time unmanned sea surface vehicle motion simulation," in *ASME 2010 International Design Engineering Technical Conferences (IDETC) & Computers and Information in Engineering Conference (CIE) August 15 -18, Montreal, Canada*, 2010.
[12] J. N. Newman, *Marine Hydrodynamics*, MIT Press, Cambridge, MA, 1977.
[13] A. Kushleyev and M. Likhachev, "Time-bounded lattice for efficient planning in dynamic environments," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation ICRA'09*, 2009, pp. 1662–1668.