2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

A Receding Horizon Controller for Motion Planning in the Presence of Moving Obstacles

Bin Xu, Daniel J. Stilwell and Andrew J. Kurdila

***Abstract*— We address the minimal risk motion planning problem in a two dimensional environment in the presence of both moving and static obstacles. Our approach is inspired by recent results due to Vladimirsky [20] in which path planning on time-varying maps is addressed using a new level-set approach, and for which computational costs are remarkably low. Toward practical implementation of these results for path planning in unstructured environments, we develop a receding-horizon formulation in which path planning for moving and static obstacles is addressed locally, while path planning for static obstacles is addressed globally. This formulation reduces the overall computational burden of path planning and makes it suitable for very large domains. The result is a suboptimal receding horizon planner and a matching condition that connects local planning with global planning. We present a rigorous analysis from which convergence to a desired endpoint is guaranteed.**

I. INTRODUCTION

We propose a sub-optimal controller for motion planning in a two dimensional dynamic environment that possesses both static and moving obstacles. In our initial work ([21] and [22]), we have addressed path planning problems based on level set methods in which an autonomous vehicle navigates in a static environment for which the *a priori* map is incomplete, and our work focused efficient computation of the level set. In this work, we extend our results to the case of moving obstacles. Our general approach is based on the level set methods introduced in ([17] and [18]), and we employ the recent path planning approach in [20] to address the case of moving obstacles. Our contribution in this paper is to rigorously justify a receding horizon formulation, similar in spirit to the approach in ([21]), in which planning with respect to static and moving obstacles is computed locally at a fast rate, and planning with respect to only static obstacles is computed globally only occasionally. This approach reduces the overall computational burden, and enables path planning in the presence of moving obstacles to be accomplished in real-time and in potentially very large domains.

In this paper, the proposed approach to path planning does not account for vehicle dynamics or otherwise addresses path-following limitations. Thus our approach is suited to vehicles that can follow arbitrary paths at potentially slow speeds. This includes certain classes of autonomous surface vehicles, which motivates our work, but also includes classes of ground vehicles and ground hovercrafts. In addition, we make no attempt to explain how the trajectory of moving obstacles is estimated, although we assume that such estimates are available to the planner.

The methods to plan motions in the presence of moving obstacles has been addressed by a variety of approaches, including a global search in state-time space ([5], [4] and [11]), speed maneuvering along a predefined trajectory ([8] and [14]), estimation of a obstacle trajectory cone, [3], and perturbation approach that assumes a parameterized polynomial trajectory ([7], [16] and [19]). Some approaches assume either that both moving and static obstacles possess a specific geometry, such as spheres, or that moving obstacles travel along piecewise linear trajectories, for example [3], [7], [14], [16], and [19]. The state-time space search methods ([5] and [11]) do not require these assumptions, but incorporate time as an extra dimension and model the time-varying environment as a static environment with one extra dimension. These methods can compute global optimal paths, but impose additional computational requirements.

In this paper, we aim to find an optimal path for an autonomous vehicle that minimizes the overall risk for traversal in a dynamic environment. We first assume that the positions of both moving and static obstacles are completely known and thus our environment map is accurate. This part of the work is inspired by Vladimirsky [20], who proposes a partial differential equation (PDE) for the time optimal control problem for non-autonomous systems. We propose a very similar PDE whose domain is the original two dimensional environment. The suboptimal controllers that maneuver the vehicle to avoid both moving and static obstacles is indeed along the gradient of the level sets of this PDE's solution. Note that since the PDE is defined over the original two dimensional environment, the computational expense is much smaller than some other methods such as [5] and [11].

For the sake of the implementation in a more realistic scenario, we consider the case in which the *a priori* map is inaccurate, and moving and static obstacles are detected during the mission by an on-board sensor with limited range. Recall that we can express the minimal risk path by the level sets of the solution to a Eikonal equation [17] over the global domain which accounts for the static obstacles only. In order to make provisions for the newly detected moving and static obstacles within the sensor field of view, we propose a receding horizon control formulation that generates local path segments. Analysis of convergence to

Bin Xu and Daniel J. Stilwell are with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24061, USA. bxu, stilwell@vt.edu
Andrew J. Kurdila is with the Department of Mechanical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24061, USA. kurdila@vt.edu

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

the desired goal location is often a challenge within the receding horizon control framework (see e.g. [13]). In [9], it is shown that the appropriate choice of the terminal cost is a crucial factor which can guarantee convergence (or stability) of the vehicle state with respect to the goal location. In this paper, we select the globally computed solution of the Eikonal equation as the terminal cost for the receding horizon controller. We show that this choice allows us to choose end points for local paths for which we can establish a sufficient condition that guarantees vehicle convergence. In terms of the computational expense, one needs only to compute the solution of the PDE over a small, local domain due to the limited length of the planning horizon. The Eikonal equation would be computed over the entire domain only occasionally. Therefore, the computational expense of the proposed receding horizon control is further reduced.

II. Problem Formulation

Consider an autonomous vehicle navigating in $\overline{\Omega} \subset \mathbb{R}^2$, where Ω is a connected and bounded open set in $\mathbb{R}^2$ and $\overline{\Omega}$ is the closure of Ω. The vehicle can be regarded as a point mass since it is small relative to $\overline{\Omega}$. The task for the vehicle is to travel along an obstacle free path such that the vehicle can reach a predefined goal $\mathbf{z} \in \overline{\Omega}$.

Letting the vehicle position be $\mathbf{x} \in \mathbb{R}^2$, we model the motion of an autonomous vehicle as a point mass whose velocity is directly controlled

$$\dot{\mathbf{x}}(t) = \mathbf{u}(t),\ \mathbf{x}(t_0) = \mathbf{x}_0 \in \overline{\Omega} \tag{1}$$

where $\mathbf{x}(t)$ is the state of the vehicle and $\mathbf{u}(t)$ is the input. We assume that the vehicle moves on a relatively low speed, such that it can turn without forward motion and can make a sudden stop as soon as the vehicle arrives at $\mathbf{z}$. The assumption is reasonable since we are concerned with altering the heading and the speed of the vehicle such that it does not move toward obstacles. Then, we can model the admissible input as $\mathbf{u}(t) \in \mathbb{U}$, where

$$\mathbb{U} = \left\{ \mathbf{u} \in \mathbb{R}^2 :\ \|\mathbf{u}\| \le v_{\max} \right\} \tag{2}$$

and $v_{\max}$ is a constant scaler indicates the maximum speed of the vehicle.

To model both static and moving obstacles in $\overline{\Omega}$, we associate a risk for the vehicle to traverse a point at any time by a cost function $g \in C^1(\overline{\Omega} \times [t_0,\infty);\mathbb{R}^1)$. For a point $\boldsymbol{\xi} \in \overline{\Omega}$, $g(\boldsymbol{\xi},t)$ remains constant when there are no moving obstacles close to $\boldsymbol{\xi}$ at time t. It increases when $\boldsymbol{\xi}$ is occupied by or very close to a moving obstacle and decreases to the constant value after the moving obstacle leaves. In addition, we assume that there exist two positive scalers G_1 and G_2 such that

$$0 < G_1 \le g(\boldsymbol{\xi},t) \le G_2 < \infty,\ \forall \boldsymbol{\xi} \in \overline{\Omega},\ \forall t \in [t_0,\infty). \tag{3}$$

To ensure that the minimal risk problem is well-posed, we only search for optimal controllers in a given time interval $[t_0,t_1]$ where $t_0 \le t_1 < \infty$. Our goal is to find $\mathbf{u}(\cdot)$ that minimizes the cumulative minimal cost from $\mathbf{x}_0$ to $\boldsymbol{\xi} \in \overline{\Omega}$ satisfying,

$$J(\boldsymbol{\xi}) = \min_{\mathbf{u}(\cdot)\in\mathbb{U}} \int_{t_0}^{T} g(\mathbf{x}(\tau),\tau)\|\dot{\mathbf{x}}(\tau)\|d\tau \tag{4}$$

subject to

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{u}(t), \\ \mathbf{x}(t) \in \overline{\Omega}, \\ \mathbf{x}(t_0) = \mathbf{x}_0, \\ \mathbf{x}(T) = \boldsymbol{\xi}, \\ \mathbf{u}(t) \in \mathbb{U}, \\ T \in [t_0,t_1]. \end{cases} \tag{5}$$

Specially, when $\mathbf{z} = \boldsymbol{\xi}$, we obtain the optimal controller $\mathbf{u}(\cdot)$ that maneuvers the vehicle from the initial location $\mathbf{x}_0$ to the goal $\mathbf{z}$.

III. Suboptimal Solution

In this section we show the necessary condition that minimizes the cost function (4). Without loss of generality, we let $t_0 = 0$ in Proposition 1.

Proposition 1: Consider the optimization problem (4) that is subject to the dynamics (5). Consider a function $Q \in C^2(\overline{\Omega};\mathbb{R}^1)$ satisfying

$$\begin{aligned} &\|\nabla Q(\boldsymbol{\xi})\| = g\left(\boldsymbol{\xi}, \frac{Q(\boldsymbol{\xi})}{\gamma G_1 v_{\max}}\right), \\ &Q(\mathbf{x}(0)) = 0. \end{aligned} \tag{6}$$

where $\gamma \in (0,1]$ is a constant scaler. If $t_1 \ge \max_{\boldsymbol{\xi}\in\overline{\Omega}} \frac{Q(\boldsymbol{\xi})}{\gamma G_1 v_{\max}}$, the optimal controller that necessarily minimizes (4) is characterized by

$$\mathbf{u}(t) = \gamma \frac{G_1 v_{\max}}{g\left(\mathbf{x}(t), \frac{Q(\mathbf{x}(t))}{\gamma G_1 v_{\max}}\right)} \frac{\nabla Q(\mathbf{x}(t))}{\|\nabla Q(\mathbf{x}(t))\|}. \tag{7}$$

Along the trajectory generated by the controller (7), the following equation holds

$$g\left(\mathbf{x}(t), \frac{Q(\mathbf{x}(t))}{\gamma G_1 v_{\max}}\right) = g(\mathbf{x}(t),t). \tag{8}$$

In addition,

$$J(\mathbf{z}) = Q(\mathbf{z}) \tag{9}$$

is the suboptimal cost that is evaluated by (7).

The partial differential equation (6) is the same as that derived in [20]. In [20], the author employs the Hamilton-Jacobi-Bellman equation and finds sufficient conditions for the global minimal-time problem. The author also establishes existence and uniqueness of the viscosity solution of the partial differential equation (6).

Remark 1: Equations (6) and (7) indicate that the suboptimal controller can be found by going along the gradient of value $Q(\boldsymbol{\xi})$ whose domain is the original two dimensional environment $\overline{\Omega}$. Therefore, to search the suboptimal controller, we need only to compute $Q(\boldsymbol{\xi})$ over the original environmental domain $\overline{\Omega}$.

Remark 2: For the case in which there are no moving obstacles, the cost function $g(\boldsymbol{\xi},t)$ depends only on its first argument. Thus, the partial differential equation (6) reduces to the well-known Eikonal equation [17]. Then, the resulting optimal controller by (7) coincides with that proposed in [12].

Remark 3: Equations (7) and (8) show that the speed of the vehicle is $\gamma\frac{G_1 v_{\max}}{g(\mathbf{x}(t),t)}$. This indicates that the higher the risk for traverse is, the slower is the speed of the vehicle. This property captures the desirable characteristic of the vehicle motion. That is, the vehicle moves slower at the place that has higher risk to traverse.

Proof: [Proof of Proposition 1] In this proof, we will use the fact that (8) holds along the trajectory generated by the controller (7). Therefore, the first step is to show the connection between $Q(\mathbf{x}(t))$ and t. Since $Q(\boldsymbol{\xi})$ is a conservative vector field, we have

$$\begin{aligned}
&Q(\mathbf{x}(t))\\
=&\int_{\mathbf{x}(0)}^{\mathbf{x}(t)}\frac{\partial Q}{\partial \mathbf{x}}d\mathbf{x}\\
=&\int_0^t\frac{\partial Q(\mathbf{x}(\tau))}{\partial \mathbf{x}(\tau)}\dot{\mathbf{x}}(\tau)d\tau\\
=&\int_0^t\frac{\partial Q(\mathbf{x}(\tau))}{\partial \mathbf{x}(\tau)}\frac{\gamma G_1 v_{\max}}{g\left(\mathbf{x}(\tau),\frac{Q(\mathbf{x}(\tau))}{\gamma G_1 v_{\max}}\right)}\frac{\nabla Q(\mathbf{x}(\tau))}{g\left(\mathbf{x}(\tau),\frac{Q(\mathbf{x}(\tau))}{\gamma G_1 v_{\max}}\right)}d\tau\\
=&\int_0^t\gamma G_1 v_{\max}d\tau\\
=&\gamma G_1 v_{\max}t
\end{aligned}\tag{10}$$

Thus, we infer that

$$t=\frac{Q(\mathbf{x}(t))}{\gamma G_1 v_{\max}},\tag{11}$$

which indicates (8).

We use Euler-Lagrange method to find the necessary condition for the minimal risk for traversing. We adjoin the system differential equation (5) to $J(\boldsymbol{\xi})$ with multiplier $\boldsymbol{\lambda}(\tau)\in\mathbb{R}^2$ (see e.g. [2], pp. 72):

$$\bar{J}=\int_{t_0}^T g(\mathbf{x}(\tau),\tau)\|\dot{\mathbf{x}}(\tau)\|+\boldsymbol{\lambda}^T(\tau)(\mathbf{u}-\dot{\mathbf{x}}(\tau))d\tau\tag{12}$$

Since $\|\mathbf{u}\|\neq 0$, we perturb $\bar{J}$ with variations to $\delta\mathbf{x}$, $\dot{\delta}\mathbf{x}$, $\delta\mathbf{u}$ and δT. The variations in $\bar{J}$ satisfy

$$\begin{aligned}
\delta\bar{J}=&\int_{t_0}^T\left[\frac{\partial g}{\partial\mathbf{x}}\delta\mathbf{x}\|\mathbf{u}\|+g\frac{\mathbf{u}^T}{\|\mathbf{u}\|}\delta\mathbf{u}+\boldsymbol{\lambda}^T\delta\mathbf{u}-\boldsymbol{\lambda}^T\delta\dot{\mathbf{x}}\right]d\tau\\
&+g(\mathbf{x}(T),T)\|\mathbf{u}(T)\|\delta T\\
=&\int_{t_0}^T\left[\frac{\partial g}{\partial\mathbf{x}}\delta\mathbf{x}\|\mathbf{u}\|+g\frac{\mathbf{u}^T}{\|\mathbf{u}\|}\delta\mathbf{u}+\boldsymbol{\lambda}^T\delta\mathbf{u}+\dot{\boldsymbol{\lambda}}^T\delta\mathbf{x}\right]d\tau\\
&-\boldsymbol{\lambda}^T\delta\mathbf{x}|_{t_0}^T+g(\mathbf{x}(T),T)\|\mathbf{u}(T)\|\delta T\\
=&\int_{t_0}^T\left[(\frac{\partial g}{\partial\mathbf{x}}\|\mathbf{u}\|+\dot{\boldsymbol{\lambda}}^T)\delta\mathbf{x}+(g\frac{\mathbf{u}^T}{\|\mathbf{u}\|}+\boldsymbol{\lambda}^T)\delta\mathbf{u}\right]d\tau\\
&-\boldsymbol{\lambda}^T\underbrace{\delta\mathbf{x}}_{\delta\mathbf{x}=d\mathbf{x}-\dot{\mathbf{x}}dt}|_T+g(\mathbf{x}(T),T)\|\mathbf{u}(T)\|\delta T\\
=&\int_{t_0}^T\left[(\frac{\partial g}{\partial\mathbf{x}}\|\mathbf{u}\|+\dot{\boldsymbol{\lambda}}^T)\delta\mathbf{x}+(g\frac{\mathbf{u}^T}{\|\mathbf{u}\|}+\boldsymbol{\lambda}^T)\delta\mathbf{u}\right]d\tau\\
&-\boldsymbol{\lambda}^T(d\mathbf{x}-\dot{\mathbf{x}}\delta T)|_T+g(\mathbf{x}(T),T)\|\mathbf{u}(T)\|\delta T
\end{aligned}\tag{13}$$

To let $\delta\bar{J}=0$, we require that each term associated with the variation terms in (13) vanishes. Thus, we have

$$\begin{aligned}
\delta\bar{J}=&\int_{t_0}^T\left[(\underbrace{\frac{\partial g}{\partial\mathbf{x}}\|\mathbf{u}\|+\dot{\boldsymbol{\lambda}}^T}_{=\mathbf{0}^T})\delta\mathbf{x}+(\underbrace{g\frac{\mathbf{u}^T}{\|\mathbf{u}\|}+\boldsymbol{\lambda}^T}_{=\mathbf{0}^T})\delta\mathbf{u}\right]d\tau\\
&-\underbrace{\boldsymbol{\lambda}^T d\mathbf{x}|_T}_{=0}+(\underbrace{g(\mathbf{x}(T),T)\|\mathbf{u}(T)\|+\boldsymbol{\lambda}^T(T)\dot{\mathbf{x}}(T)}_{=0})\delta T
\end{aligned}\tag{14}$$

Note that the terminal constraint is $\mathbf{x}(T)=\boldsymbol{\xi}$. Thus, we infer that

$$d\mathbf{x}|_T=[0,0]^T.\tag{15}$$

Therefore, from (14) we obtain a set of Euler-Lagrange equations:

$$\frac{\partial g(\mathbf{x}(t,t)}{\partial\mathbf{x}(t)}\|\mathbf{u}(t)\|+\dot{\boldsymbol{\lambda}}^T(t)=\mathbf{0}^T,\tag{16}$$

$$g(\mathbf{x}(t),t)\frac{\mathbf{u}^T(t)}{\|\mathbf{u}(t)\|}+\boldsymbol{\lambda}^T(t)=\mathbf{0}^T,\tag{17}$$

for all $t\in[0,T]$ and boundary conditions

$$g(\mathbf{x}(T),T)\|\mathbf{u}(T)\|+\boldsymbol{\lambda}^T(T)\dot{\mathbf{x}}(T)=0.\tag{18}$$

To show that the solution of $\boldsymbol{\lambda}(t)$ satisfies

$$\boldsymbol{\lambda}(t)=-\nabla Q(\mathbf{x}(t)),\tag{19}$$

we show the equations (16), (17), and (18) hold assuming (7). Substituting (7), (6), (8), and (19) into the left-hand side of (17), we obtain

$$\begin{aligned}
&g(\mathbf{x}(t),t)\frac{\mathbf{u}(t)}{\|\mathbf{u}(t)\|}+\boldsymbol{\lambda}(t)\\
=&g(\mathbf{x}(t),t)\frac{\nabla Q(\mathbf{x})}{\|\nabla Q(\mathbf{x})\|}-\nabla Q(\mathbf{x}(t))\\
=&[0,0]^T,
\end{aligned}\tag{20}$$

which verifies that (17) holds. Substituting (7), (6), (8), and (19) into the left-hand side of (18), we derive

$$\begin{aligned}
&g(\mathbf{x}(T),T)\|\mathbf{u}(T)\|+\boldsymbol{\lambda}^T(T)\dot{\mathbf{x}}(T)\\
=&g(\mathbf{x}(T),T)\frac{\gamma G_1 v_{\max}}{g\left(\mathbf{x}(T),\frac{Q(\mathbf{x}(T))}{\gamma G_1 v_{\max}}\right)}\\
&-\nabla Q(\mathbf{x}(T))^T\frac{\gamma G_1 v_{\max}}{g\left(\mathbf{x}(T),\frac{Q(\mathbf{x}(T))}{\gamma G_1 v_{\max}}\right)}\frac{\nabla Q(\mathbf{x}(T))}{\|\nabla Q(\mathbf{x}(T))\|}\\
=&\gamma G_1 v_{\max}-\gamma G_1 v_{\max}\\
=&0
\end{aligned}\tag{21}$$

which indicates that (18) holds. Equation (16) remains to be proven. Given (6) and (8), we have

$$\frac{\partial g(\mathbf{x}(t),t)^T}{\partial \mathbf{x}} = \frac{\partial \|\nabla Q(\mathbf{x}(t))\|^T}{\partial \mathbf{x}} = \frac{1}{g}\frac{\partial^2 Q}{\partial \mathbf{x}^2}\frac{\partial Q^T}{\partial \mathbf{x}} \tag{22}$$

From (7) and (19), $\dot{\boldsymbol{\lambda}}$ satisfies

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial^2 Q}{\partial \mathbf{x}^2}\dot{\mathbf{x}} = -\frac{\partial^2 Q}{\partial \mathbf{x}^2}\frac{\gamma G_1 v_{\max}}{g\left(\mathbf{x}, \frac{Q(\mathbf{x})}{\gamma G_1 v_{\max}}\right)}\frac{\nabla Q(\mathbf{x})}{\|\nabla Q(\mathbf{x})\|} \tag{23}$$

Substituting (7), (6), (22), and (23) into the left-hand side of (16), we obtain

$$\begin{aligned}
&\frac{\partial g(\mathbf{x}(t),t)^T}{\partial \mathbf{x}(t)}\|\mathbf{u}(t)\| + \dot{\boldsymbol{\lambda}}(t)\\
=&\frac{1}{g\left(\mathbf{x}(t), \frac{Q(\mathbf{x}(t))}{\gamma G_1 v_{\max}}\right)}\frac{\partial^2 Q}{\partial \mathbf{x}^2}\frac{\partial Q^T}{\partial \mathbf{x}}\frac{\gamma G_1 v_{\max}}{g\left(\mathbf{x}(t), \frac{Q(\mathbf{x}(t))}{\gamma G_1 v_{\max}}\right)}\\
&-\frac{\partial^2 Q}{\partial \mathbf{x}^2}\frac{\gamma G_1 v_{\max}}{g\left(\mathbf{x}, \frac{Q(\mathbf{x})}{\gamma G_1 v_{\max}}\right)}\frac{\nabla Q(\mathbf{x})}{\|\nabla Q(\mathbf{x})\|}\\
=&[0,0]^T
\end{aligned} \tag{24}$$

which shows that (16) holds. Since all of the three Euler-Lagrange equations (16), (17), and (18) hold, we prove that $\boldsymbol{\lambda}(t)$ satisfies (19) and that (7) satisfies the necessary condition to minimize (4).

In the end, to show (9), we substitute (7) and (8) into (4) and obtain

$$\begin{aligned}
J(\mathbf{z}) =& \int_0^T g(\mathbf{x}(\tau),\tau)\|\dot{\mathbf{x}}(\tau)\|d\tau\\
=& \int_0^T g(\mathbf{x}(\tau),\tau)\frac{\gamma G_1 v_{\max}}{g\left(\mathbf{x}(\tau), \frac{Q(\mathbf{x}(\tau))}{\gamma G_1 v_{\max}}\right)}d\tau\\
=&\gamma G_1 v_{\max} T
\end{aligned} \tag{25}$$

From (11), we conclude

$$J(\mathbf{z}) = \gamma G_1 v_{\max}\frac{Q(\mathbf{z})}{\gamma G_1 v_{\max}} = Q(\mathbf{z}) \tag{26}$$

which completes our proof. ■

IV. Receding Horizon Control Formulation when the State of the Moving Obstacles Detected in Mission

In a more realistic scenario, the vehicle often detects the moving obstacles in real-time using an on-board sensor with limited range. In this section we introduce a receding horizon control formulation (see, e.g. [6]) that addresses the case that moving and static obstacles are detected in real-time only within the vehicle's limited sensor range. We first give a necessary condition to optimize the cost of the proposed receding horizon controller. We show that by computing a PDE over a local domain centered at the vehicle's current position, we can find a suboptimal controller minimizing the proposed cost functional over the planning horizon. Thus, we can forgo the computation of the new PDE defined in (6) at each iteration over the entire domain even if we detect new moving obstacles. Then, we propose a sufficient condition that guarantees that the vehicle converges to the goal using a sequence of local plans. We show that we can apply the methodology proposed in [15] and [9] to justify this sufficient condition.

Before we introduce the receding horizon control formulation, we construct a Eikonal equation ([17]) satisfying

$$\|\nabla Q^*(\boldsymbol{\xi})\| = g^*(\boldsymbol{\xi}),\ Q^*(\mathbf{z}) = 0. \tag{27}$$

where $g^* \in C^1(\overline{\Omega};\mathbb{R}^+)$ corresponding to the risk for traversal when there are only static obstacles in the environment. In our approach, the solution Q^* is computed over the entire domain $\overline{\Omega}$. Note that $g(\cdot,\cdot)$ and the level set Q defined in Section II account for static and moving obstacles locally around the current position of the vehicle. Since for the same location, the risk to traverse will increase if a moving obstacle is passing through, we can assume

$$0 < G_1 \le g^*(\boldsymbol{\xi}) \le g(\boldsymbol{\xi},t) \le G_2,\ \forall\ (\boldsymbol{\xi},t) \in \overline{\Omega}\times[t_0,\infty).$$

It is well known (e.g. [12] and [17]) that the solution $Q^*(\boldsymbol{\xi})$, in contrast to $Q(\cdot)$ defined in (6), encodes the minimal risk path from $\boldsymbol{\xi}$ to the goal $\mathbf{z}$ in a static environment.

In the receding horizon control formulation that is proposed herein, the values of Q^* serve as the terminal cost in the local cost functional. Indeed, in Proposition 2, we compute a path in a local region around the vehicle using the results in Section III. This local path will account for moving obstacles. We also compute a global path as in (27) that accounts for only static obstacles. When computing local paths, it is not immediately obvious how to choose the end-point of the path since a local region may not include the desired goal location. It is also important to show that a sequence of local paths does eventually lead to the desired goal location. The level set Q^* that is computed for the global path is used to address both of these challenges.

If the perception sensor's sampling period is h, we often let the implementation horizon to be h as well. We choose H as the planning horizon where $H \ge h$. Note that the receding horizon control formulation shall make provisions for the newly detected moving obstacles. Thus, we employ $g(\mathbf{x}(\tau),\tau)$ for the process cost during the horizon H and aim to minimize the risk for traversal over the horizon as well as the expected minimal risk for traversal from the terminal state $\mathbf{x}(t_k+H)$ toward the goal. The receding horizon formulation is then to find the local optimal control and state pair on $[t_k, t_k+H]$ that solves the minimization problem

$$J(\mathbf{x}(\cdot),t_k) = \min_{\mathbf{u}(\cdot)\in\mathbb{U}}\int_{t_k}^{t_k+H} g(\mathbf{x}(\tau),\tau)\|\dot{\mathbf{x}}(\tau)\|d\tau + Q^*(\mathbf{x}(t_k+H)) \tag{28}$$

subject to

$$\begin{cases}
\dot{\mathbf{x}}(t) = \mathbf{u}(t),\\
\mathbf{x}(t_0) = \mathbf{x}_0,\\
\mathbf{x}(t_k) = \mathbf{x}(t_{k-1}+h),\\
\mathbf{x}(t) \in \overline{\Omega}.
\end{cases} \tag{29}$$

for $k=0,1,2,\dots,\infty$. The overall planned trajectory and control on $[t_0,\infty)$ are spliced together from the locally optimal trajectory in the usual way (see, e.g. [6]). The following proposition shows the necessary condition to minimize (28).

Proposition 2: Consider the optimization problem (28) that is subject to the dynamics (29), and a function $Q_k \in C^2(\overline{\Omega};\mathbb{R}^1)$ satisfying

$$\begin{aligned} &\|\nabla Q_k(\boldsymbol{\xi})\| = g\left(\boldsymbol{\xi}, \frac{Q_k(\boldsymbol{\xi})}{\gamma G_1 v_{\max}} + t_k\right), \\ &Q_k(\mathbf{x}(t_k)) = 0. \end{aligned} \tag{30}$$

where $\gamma \in (0,1]$ is a constant scaler. If the matching condition

$$\nabla Q_k(\mathbf{x}(t_k+H)) = -\nabla Q^*(\mathbf{x}(t_k+H)), \quad \text{if } \mathbf{x}(t_k+H) \neq \mathbf{z}, \tag{31}$$

is satisfied, then optimal controller that necessarily minimizes (28) is characterized by

$$\mathbf{u}(t) = \begin{cases} \gamma \frac{G_1 v_{\max}}{g\left(\mathbf{x}(t), \frac{Q_k(\mathbf{x}(t))}{\gamma G_1 v_{\max}} + t_k\right)} \frac{\nabla Q_k(\mathbf{x}(t))}{\|\nabla Q_k(\mathbf{x}(t))\|}, & \text{if } \mathbf{x}(t) \neq \mathbf{z}, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{32}$$

Remark 4: The condition (31) indeed shows how to chose the end-point of the path over each planning horizon H.

Remark 5: Note that from (11), the value of the Q_k indeed corresponds to the relative time that ASV travels from its current position to any point in the environment. Since the planning horizon is H, using the ordered upwind method ([18]), we can terminate the computation of Q_k as soon as the solution is larger than the constant $\gamma G_1 v_{\max} H$. Thus, Q_k will be computed over a small, local domain.

Proof: [Proof of Proposition 2] The proof is very much similar to that of Proposition 1, given the fact that the matching condition (31) holds. For reasons of limited space, we thus omit it herein. ■

We now present the sufficient condition such that the vehicle will converge to the goal under the receding horizon control formulation (28).

Proposition 3: Assume that the viscosity solution of $Q^* \in C^1(\overline{\Omega};\mathbb{R})$. Define the value function

$$V(\mathbf{x}(t_k),t_k) = J(\mathbf{x}(\cdot),t_k). \tag{33}$$

We assume that $V(\mathbf{x},t)$ is a continuously differentiable function such that

$$W_1(\mathbf{x}-\mathbf{z}) \leq V(\mathbf{x},t) \leq W_2(\mathbf{x}-\mathbf{z}),$$

where $W_1(\cdot)$ and $W_2(\cdot)$ are continuous positive definite functions on $\overline{\Omega}$. If the matching condition (31) holds for any time t_k, $V(\mathbf{x}(t_k),t_k)$ is a Lyapunov function. The vehicle state $\mathbf{x}(t) \to \mathbf{z}$ as $t \to \infty$.

Remark 6: In the general case, $Q^*(\boldsymbol{\xi})$ is not differentiable everywhere. Interested readers are referred to [1] for the stability analysis when a Lyapunov function is not in C^1.

Remark 7: The proof of Proposition 3 follows very similar procedure of the asymptotical stability proof of the receding horizon controller proposed in [15] and [9]. The difference is that since the risk for traversal $g(\mathbf{x}(t),t)$ depends on time, the optimal cost $J(\mathbf{x}(\cdot),t_k)$ depends on time as well. Thus, the corresponding Lyapunov function $V(\mathbf{x},t)$ that appears in the proof of Proposition 3 is also time-varying.

Proof: [Proof of Proposition 3] Following the procedure in [15] and [9], we denote by $\mathbf{u}_f(t)$ the admissible feedback controller that ensures the vehicle converge to the goal $\mathbf{z}$. Given the level set values Q^*, let the controller $\mathbf{u}_f$ satisfy

$$\mathbf{u}_f(t) = -\alpha v_{\max} \frac{\nabla Q^*(\mathbf{x}(t))}{\|\nabla Q^*(\mathbf{x}(t))\|}, \text{ if } \mathbf{x}(t) \neq \mathbf{z}, \tag{34}$$

for all $t \in [t_0,\infty)$, where

$$0 < \alpha \leq \gamma \frac{G_1}{G_2} \leq 1. \tag{35}$$

Thus, from (27) we infer that

$$\dot{Q}^*(\mathbf{x}(t)) = \nabla Q^*(\mathbf{x}(t)) \cdot \mathbf{u}_f(t) = -g^*(\mathbf{x}(t)) \alpha v_{\max}. \tag{36}$$

Therefore, $Q^*(\mathbf{x}(t))$ is a control Lyapunov function. Denote the optimal controller that minimizes $J(\mathbf{x}(\cdot),t_k)$ by $\mathbf{u}^*(t)$ for $t \in [t_k, t_k+H]$. Since $t_{k+1} = t_k + h$ and since $h \leq H$, we can define the admissible controller $\mathbf{u}^+(t)$ satisfying

$$\mathbf{u}^+(t) = \begin{cases} \mathbf{u}^*(t), \ \forall t \in [t_{k+1}, t_{k+1}+H-h] \\ \mathbf{u}_f(t), \ \forall t \in [t_{k+1}+H-h, t_{k+1}+H] \end{cases} \tag{37}$$

Given the initial state $\mathbf{x}(t_{k+1})$ at t_{k+1}, we define $\mathscr{J}(\mathbf{x}(t_{k+1}),\mathbf{u}^+(\cdot))$ the cost function that is evaluated by $\mathbf{u}^+(\cdot)$ satisfying

$$\begin{aligned} &\mathscr{J}(\mathbf{x}(t_{k+1}),\mathbf{u}^+(\cdot)) \\ &:= \int_{t_{k+1}}^{t_{k+1}+H} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}^+(\tau)\| d\tau + Q^*(\mathbf{x}(t_{k+1}+H)) \end{aligned} \tag{38}$$

Since $J(\mathbf{x}(\cdot),t_{k+1})$ is the minimal cost, we obtain the following inequality

$$\begin{aligned} &J(\mathbf{x}(\cdot),t_{k+1}) \\ &\leq \mathscr{J}(\mathbf{x}(t_{k+1}),\mathbf{u}^+(\cdot)) \\ &= \int_{t_{k+1}}^{t_{k+1}+H-h} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}^+(\tau)\| d\tau \\ &\quad + \int_{t_{k+1}+H-h}^{t_{k+1}+H} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}^+(\tau)\| d\tau + Q^*(\mathbf{x}(t_{k+1}+H)) \\ &= \int_{t_k+h}^{t_k+H} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}^*(\tau)\| d\tau \\ &\quad + \int_{t_k+H}^{t_k+h+H} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}_f(\tau)\| d\tau + Q^*(\mathbf{x}(t_{k+1}+H)) \end{aligned}$$

$$
\begin{aligned}
&= \underbrace{\int_{t_k}^{t_k+H} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}^*(\tau)\|d\tau + Q^*(\mathbf{x}(t_k+H))}_{J(\mathbf{x}(\cdot),t_k)} \\
&\quad - \int_{t_k}^{t_k+h} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}^*(\tau)\|d\tau - Q^*(\mathbf{x}(t_k+H)) \\
&\quad + \int_{t_k+H}^{t_k+h+H} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}_f(\tau)\|d\tau + Q^*(\mathbf{x}(t_{k+1}+H)) \\
&= J(\mathbf{x}(\cdot),t_k) - \int_{t_k}^{t_k+h} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}^*(\tau)\|d\tau - Q^*(\mathbf{x}(t_k+H)) \\
&\quad + \int_{t_k+H}^{t_k+h+H} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}_f(\tau)\|d\tau + Q^*(\mathbf{x}(t_k+H+h))
\end{aligned} \tag{39}
$$

From (34), (36) and (37), together with the inequality (39), we infer that

$$
\begin{aligned}
&J(\mathbf{x}(\cdot),t_{k+1}) - J(\mathbf{x}(\cdot),t_k) \\
&\leq - \int_{t_k}^{t_k+h} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}^*(\tau)\|d\tau \\
&\quad + \int_{t_k+H}^{t_k+h+H} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}_f(\tau)\|d\tau + \int_{t_k+H}^{t_k+h+H} \frac{\partial Q^*}{\partial \mathbf{x}}(\tau)\mathbf{u}_f(\tau)d\tau \\
&= - \int_{t_k}^{t_k+h} g(\mathbf{x}(\tau),\tau)\|\mathbf{u}^*(\tau)\|d\tau + \int_{t_k+H}^{t_k+h+H} g(\mathbf{x}(\tau),\tau)\alpha v_{\max} d\tau \\
&\quad - \int_{t_k+H}^{t_k+h+H} g^*(\mathbf{x}(\tau))\alpha v_{\max} d\tau
\end{aligned} \tag{40}
$$

From the fact that the matching condition (31) holds, we infer that $\mathbf{u}^*$ satisfies (32). Thus (40) further becomes

$$
\begin{aligned}
&J(\mathbf{x}(\cdot),t_{k+1}) - J(\mathbf{x}(\cdot),t_k) \\
&\leq - \int_{t_k}^{t_k+h} \gamma v_{\max} G_1 d\tau + \int_{t_k+H}^{t_k+h+H} g(\mathbf{x}(\tau),\tau)\alpha v_{\max} d\tau \\
&\quad - \int_{t_k+H}^{t_k+h+H} g^*(\mathbf{x}(\tau))\alpha v_{\max} d\tau \\
&= \int_{t_k+H}^{t_k+h+H} (g(\mathbf{x}(\tau),\tau)\alpha - \gamma G_1) v_{\max} d\tau \\
&\quad - \int_{t_k+H}^{t_k+h+H} g^*(\mathbf{x}(\tau))\alpha v_{\max} d\tau
\end{aligned} \tag{41}
$$

From (35), the inequality (41) becomes

$$
\begin{aligned}
&J(\mathbf{x}(\cdot),t_{k+1}) - J(\mathbf{x}(\cdot),t_k) \\
&\leq \int_{t_k+H}^{t_k+h+H} \underbrace{\left(g(\mathbf{x}(\tau),\tau)\gamma \frac{G_1}{G2} - \gamma G_1 \right)}_{\leq 0} v_{\max} d\tau \\
&\quad - \int_{t_k+H}^{t_k+h+H} g^*(\mathbf{x}(\tau))\alpha v_{\max} d\tau \\
&\leq - \int_{t_k+H}^{t_k+h+H} g^*(\mathbf{x}(\tau))\alpha v_{\max} d\tau
\end{aligned} \tag{42}
$$

According to (33), we have for all $\mathbf{x}(t_k) \neq \mathbf{z}$,

$$
\begin{aligned}
&\dot{V}(\mathbf{x}(t_k),t_k) \\
&= \lim_{h\to 0} \frac{V(\mathbf{x}(t_{k+1}),t_{k+1}) - V(\mathbf{x}(t_k),t_k)}{h} \\
&\leq - g^*(\mathbf{x}(t_k+H))\alpha v_{\max} \\
&\leq - G_1 \alpha v_{\max}.
\end{aligned} \tag{43}
$$

Since the environmental geometry $\overline{\Omega}$ is a closed and bounded set, there exists a continuous positive definite function $W_3 : \overline{\Omega} \to \mathbb{R}$ such that

$$
W_3(\mathbf{x}-\mathbf{z}) \leq G_1 \alpha v_{\max}.
$$

Thus, the inequality (43) becomes

$$
\dot{V}(\mathbf{x}(t_k),t_k) \leq -W_3(\mathbf{x}(t_k)-\mathbf{z}).
$$

By the hypothesis, $V(\mathbf{x},t)$ is a Lyapunov function for a non-autonomous system. Thus, we conclude that the vehicle converge to the goal $\mathbf{z}$ (see e.g. Theorem 4.9 of [10]). ■

V. Simulation Results

To validate the proposed receding horizon controller, we show an example of an autonomous surface vehicle (ASV) navigating in a riverine environment. We employ the ordered upwind method proposed in [18] and [20] for computing the solutions of Eikonal equation and the PDE (30). This method has a computational complexity of $O(N \ln N)$ where N is the total number of the cells in the domain $\overline{\Omega}$. Figure 1 represents the river map which spans an area of $800m \times 600m$. There are three moving obstacles traveling in the river environment as seen in Figure 1. The trajectories of these obstacles are chosen so that a collision will occur if the vehicle does not account for the trajectory of the obstacles. The on-board sensor can detect them within 60 meters range. We assume that as soon as the moving obstacles enter the detection range, the ASV immediately knows their trajectories. At any time t, if a cell is within 9 meters around the moving obstacles, the cost function $g(\boldsymbol{\xi},t) = 7$. Otherwise, $g(\boldsymbol{\xi},t) = 0.2$. Thus, we set the lower bound $G_1 = 0.2$. We choose the planning horizon $H = 50$ *secs*, the maximum speed $v_{\max} = 5$ m/s and $\gamma = 1$. Since the condition (31) is satisfied at each horizon, as shown in Figure 2, the ASV eventually reaches the target.

In order to illustrate how to choose the end point at each horizon as well as how the ASV manages to avoid the moving obstacles, we discuss the receding horizon planning process corresponding to the ASV's motion shown in Rectangle 1 in Figure 2. Note that the solution of the Eikonal equation (27) over the global domain serves as the terminal cost in the receding horizon control formulation (28). When we replan the path over each horizon, we compute the solution of the PDE (30) over a small, local domain as shown in Figure 3. Due to the discussion in Remark 5, we terminate the computation of the local PDE as soon as its solution is higher than $\gamma G_1 v_{\max} H$. Thus, as shown in Figure 3, the PDE's solutions to the areas corresponding to the moving obstacles have not been computed before the termination of

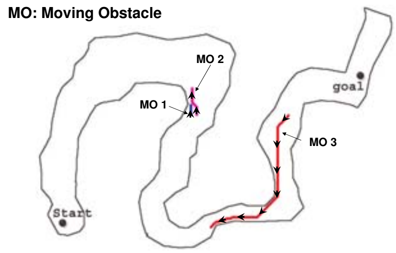

Fig. 1. The riverine map and moving obstacles.

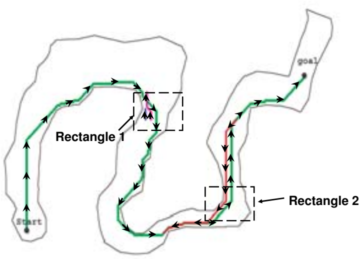

Fig. 2. The planned trajectory of the ASV plotted onto the map.

computing the local PDE because they will be higher than $\gamma G_1 v_{\max} H$. When the ASV searches for the new path, it automatically avoids going through these areas. To choose the end point for the planning horizon, we check whether the condition (31) holds at the end of the planning horizon as shown in Figure 3. To accommodate numerical errors, we relax the condition (31) so that if the difference between the two hand sides of (31) is smaller than a given threshold we say that the condition (31) holds. In this example, the global level set Q^* is only computed once at the beginning the mission. In practice, it would be computed as needed to ensure that the global level-set adequately represents the environment. This idea is clarified and discussed in detail in [22].

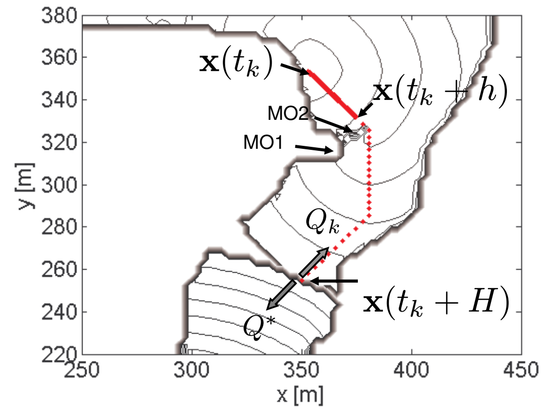

Fig. 3. The solution of the local PDE, the partial solution of the global Eikonal equation and the planned trajectory. Two gray arrows correspond to $\nabla Q_k(\mathbf{x}(t_k+H))$ and $\nabla Q^*(\mathbf{x}(t_k+H))$ respectively.

REFERENCES

[1] J.-P. Aubin and A. Cellina, *Differential Inclusions*, Speinger-Verlag, 1984.

[2] A.E. Bryson, Jr. and Y.C. Ho, *Applied Optimal Control*, Taylor & Francis, 1975.

[3] P. Fiorini and Z. Shiller, "Time optimal trajectory planning in dynamic environments," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1553–1558, 1994.

[4] T. Fraichard, "Dyanamic trajectory planning with dynamic constraints: a 'state-time space' approach," *Proc. of IEEE/RSJ conference on Intelligent Robts and Systems*, pp.1393–1399, 1993.

[5] K. Fujimura and H. Samet, "A hierarchical strategy for path planning among moving obstacles," *Trans on Robotics and Automation*, vol. 5, no. 1, pp. 61–69, 1989.

[6] G.C. Goodwin, M.M. Seron and J.A. De Doná, *Contrstrained Control and Estimation: An Optimisation Approach*, Springer-Verlag, 2005.

[7] Y. Guo and T. Tang, "Optimal trajectory generation for nonholonomic robots in dynamic environments," *Proc. of International Conference on Robotics and Automation*, pp. 2552–2557, 2008.

[8] K.S. Hwang and M.Y. Ju, "Speed planning for maneuvering motion," *Journal of Intelligent and Robotic Systems*, no. 33, pp. 25–44, 2002.

[9] A. Jadbabaie, J. Yu, and J. Hauser, "Stabilizing receding horizon control of nonlinear systems: a control Lyapunov function approach," *Proc. of the American Control Conference*, pp. 1535–1539, 1999.

[10] H.K. Khalil, *Nonlinear Systems, Third Edition*, Prentice Hall, 2002.

[11] R. Kimmel, N. Kiryati, and A.M. Bruckstein, "Multivalued distance maps for motion planning on surfaces with moving obstacles," *IEEE Trans on Robotics and Automation*, vol. 14, no. 3, 1998.

[12] R. Kimmel and J.A. Sethian, "Optimal Algorithm for Shape from Shading and Path Planning," *Journal of Mathematical Imaging and Vision*, vol. 14, pp.237-244, 2001.

[13] Y. Kuwata and J.P. How, "Stable Trajectory Design for Highly Constrained Environments using Receding Horizon Control," *Proc. of American Control Conference*, pp. 902-907, 2004.

[14] K.J. Kyriakopoulos and G.N. Saridis, "An integrated collision prediction and avoidance scheme for mobile robots in non-stationary environments," *Automatica*, vol. 29, no. 2, pp.309–322, 1993.

[15] D.Q. Mayne, J.B. Rawlings, C.V. Rao and P.O.M. Scokaert, "Survey Paper: Constrained model predictive control: Stability and optimality", *Automatica*, no.36, pp. 789-814, 2000.

[16] Z. Qu, J. Wang, and C.E. Plaisted, "A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles," *IEEE Trans. on Robotics*, vol. 20, no. 6, 2004.

[17] J.A. Sethian, "Fast Marching Method," *SIAM Review*, vol.41, No.2, pp.199-235, 1999.

[18] J.A. Sethian and A. Vladimirsky "Ordered upwind methods for static Hamilton-Jacobi equations," *Proc. Nat. Acad. Sci. USA*, vol.98,pp. 11069-11074, 2001.

[19] A. Tsoularis and C. Kambhampati, "Avoiding moving obstacles by deviation from a mobile robot's nominal path", *The International Journal of Robtics Research*, vol. 18, no.5, pp. 454–465, 1999.

[20] A. Vladimirsky, "Static PDE for Time-Dependent Control Problems ", *Interfaces and Free Boundaries*, no.8, pp. 281-300, 2006.

[21] B. Xu, A.J. Kurdila, and D.J. Stilwell, "A hybrid receding horizon control method for path planning in uncertain environments", *Proc. of IEEE/RSJ conference on Intelligent Robts and Systems*, 2009.

[22] B. Xu, D.J. Stilwell, and A.J. Kurdila "Efficient computation of level sets for path planning", *Proc. of IEEE/RSJ conference on Intelligent Robts and Systems*, 2009.