# Extended Performance Guarantees for Receding Horizon Search with Terminal Cost

Benjamin Biggs Virginia Tech babiggs@vt.edu Daniel J. Stilwell Virginia Tech stilwell@vt.edu James McMahon US Naval Research Laboratory Code 7130 james.mcmahon@nrl.navy.mil

Abstract—The computational difficulty of planning search paths that seek to maximize a general deterministic value function increases dramatically as desired path lengths increase. Mobile search agents with limited computational resources often utilize receding horizon methods to address the path planning problem. Unfortunately, receding horizon planners may perform poorly due to myopic planning horizons. We provide methods of incorporating terminal costs in the construction of receding horizon paths that provide a theoretical lower bound on the performance of the search paths produced. The results presented in this paper are of particular value in subsea search applications. We present results from simulated subsea search missions that use real-world data acquired by an autonomous underwater vehicle during a subsea survey of Boston Harbor.

#### I. INTRODUCTION

We address the problem of planning paths for mobile search agents that seek to maximize search effectiveness in finite time. We are specifically interested in situations where the search area is large enough that exhaustive search is likely not possible and the length of search paths precludes the possibility of solving for optimal paths in real time. Our work is inspired by subsea search applications where the scale of the environment is orders of magnitude larger than the search agent while the features being measured are equivalent to the size of the agent. We provide numerical simulations of our approach using real-world data acquired by an autonomous underwater vehicle. Throughout this paper, optimal search paths maximize a general deterministic value function.

We provide formal guarantees on the performance of receding horizon approaches to informative path planning. In a receding horizon approach to path planning, an agent computes an optimal (e.g., maximizes information gain) path that is sufficiently short that it can be computed in real-time. An agent follows the first few steps of a short optimal path, and then computes another short-horizon optimal path. The agent does this iteratively until it has traversed a path that has a desired length or the mission otherwise terminates.

Washington D.C., USA Benjamin Biggs and Daniel Stilwell are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA Although each short-horizon path is optimal, the receding horizon path constructed from a sequence of short-horizon paths does not necessarily inherit the desirable properties of the short-horizon paths. In essence, searching a large area using receding horizon path planning methods, while generally effective, can result in poor performance due to myopic planning horizons that lead a search agent to become trapped in local maxima/minima.

In prior work, we showed that a specific class of terminal cost could be appended to the optimization problem that would lead to useful lower bounds for informative path planning, and that for applications such as subsea mine-hunting, which is the motivation for our work, a suitable terminal cost is always available. Our approach in [1] addresses the case that the short-horizon path is the optimal, but could not be directly extended to the case that the short-horizon path is sub-optimal, as would happen using many realtime planning approaches, such sample-based planning (e.g., Monte-Carlo Tree Search). Herein, we specifically address the case of sample-based planning that generates sub-optimal short-horizon paths. We propose a specific terminal cost and corresponding sufficient conditions for which the receding horizon path inherits a sub-optimal property from the suboptimality of each short-horizon path.

Receding horizon path planning methods have their roots in receding horizon control [2]–[4]. The methods proposed in this work and in [1] are inspired by works in receding horizon control such as [5]–[7] and [8]. Approaches to path planning that incorporate terminal costs include [9], [10] and [11]. It is shown in [11] that in order to converge to a goal location the cost-to-go at each step should represent a bound on the cost of the receding horizon trajectory plus the terminal cost at the next iteration. In prior work [1], the authors perform a similar analysis and prove that the existence of a feasible next-step, cost-to-go combination with a reward greater than the costto-go from the final location along the last path produced at each planning step is sufficient to guarantee a lower bound on the expected reward of the overall receding horizon path. Neither [1] nor [11] addresses sub-optimal paths within the planning horizon.

Approaches to informative path planning related to this paper include [12]–[15], where the concept of algebraic re-

<sup>\*</sup>This work was supported by the Office of Naval Research via grants N00014-18-1-2627, and N00014-19-1-2194. The work of J. McMahon is supported by the Office of Naval Research through the NRL Base Program. James McMahon is with the US Naval Research Laboratory, Code 7130,

dundancy is used to prune a search tree. These methods allow for longer planning horizons, but do not utilize terminal costs. In [16], the planning horizon is extended until the reward of a path surpasses a threshold value, but no terminal cost is appended. The methods in [17], [18] employ sampling based methods to plan non-myopic search paths with guaranteed convergence to the optimal search path in the case of [18]. In contrast we use sampling based methods (Monte Carlo Tree Search) within a short planning horizon and append terminal costs in order to guarantee a lower bound on the performance of search paths. Novel roll-out policies for Monte Carlo Tree Search are presented in [19] in the context of subsea search path planning. We use a slightly modified version of Monte Carlo Tree Search within a short planning horizon and incorporate terminal costs.

The organization of this paper is as follows. The problem description is presented in Section II. The *cost-to-go* function is defined and a path value function that includes the terminal cost is presented in Section III. Sufficient conditions for sub-optimality guarantees on the expected value of receding horizon paths constructed with terminal costs are presented in Section IV. Numerical experiments are described and results are presented in Section V. Proofs are provided in the appendices.

#### **II. PROBLEM DESCRIPTION**

We consider a search region  $\mathcal{G} \subset \mathbb{R}^2$ . An *n*-length path through the search region beginning at some initial location  $p_0 \in \mathcal{G}$  is denoted  $\gamma_n(p_0)$  and is composed of a sequence of locations  $\{p_0, p_1, \ldots, p_n\}$ . A sequence is feasible if it can be traversed by a mobile search agent. To be explicit, we consider that the state of a mobile search agent at time *t* may be represented by the tuple  $s_t = \{p_t, x_t\}$  where  $x_t$  gives all states of the mobile search agent excluding the position states  $p_t$ . We say that a state  $s_t$  is feasible from the state  $s_{t-1}$  if there exists an action *a* available to the search agent such that applying *a* to the state  $s_{t-1}$  results in the new state  $s_t$ . An *n*-length path may then be represented as a sequence of states  $\gamma_n(s_0) = \{s_0, s_1, \ldots, s_n\}$ . The set of all feasible *n*-length paths beginning at state  $s_0$  is denoted  $\Gamma_n(s_0)$ .

The value of a path is given by

$$J(\gamma_n(s_0)) = \sum_{i=1}^n g(s_i, s_{i-1}, \dots, s_0)$$
(1)

where g is a non-negative function that returns the anticipated value of the information gained while transitioning from one state to the next. We assume that the value of information gained while transitioning states is dependent on previously executed state transitions. That is,  $g(s_i, s_{i-1}, \ldots, s_0)$  returns the value of transitioning from state  $s_{i-1}$  to  $s_i$  given all previous state transitions. In essence, if a location has been searched before then the reward for searching the location again will likely be discounted. In the context of search, the goal is to plan a path that maximizes the amount of information gained while traversing the path. The optimal path with respect to (1) is

$$\gamma_n^*(s_0) = \underset{\gamma_n(s_0) \in \Gamma_n(s_0)}{\arg \max} J_n(s_0)$$
(2)

with value

$$J(\gamma_n^*(s_0)) = \max_{\gamma_n(s_0) \in \Gamma_n(s_0)} J_n(s_0)$$
(3)

For notational simplicity we define

$$g(s_i) \triangleq g(s_i, s_{i-1}, \dots, s_0) \tag{4}$$

and

$$J_n(s_0) \triangleq J(\gamma_n(s_0)) \tag{5}$$

## III. RECEDING HORIZON PATH PLANNING AND THE Cost-to-Go FUNCTION

The process of planning an *l*-length receding horizon search path is the following. The search agent solves for an optimal path within an *n*-length planning horizon with  $n \ll l$ . The agent then traverses the optimal *n*-length path out to an *m*-length execution horizon with  $m \le n$ . The agent repeats this process until it has traversed a path of length *l*. Receding horizon path planning is generally effective, but it is well known that receding horizon paths do not retain the optimality property of the short paths used to construct them. That is, only considering paths of length *n* that are myopic with respect to the overall mission length *l* or the size of the search region  $\mathcal{G}$  can lead to undesirable behavior.

To address the problem of poor performance caused by myopic planning horizons, we consider a path of length  $l \gg n$ . The reward associated with the *l*-length path is

$$J_l(s_0) = \sum_{i=1}^{l} g(s_i)$$
(6)

where  $g(s_i)$  is the simplified notation for the same nonnegative function g as in (1). We note that the summation on the right-hand side of (6) may be separated into 2 parts and written as

$$J_l(s_0) = \sum_{i=1}^n g(s_i) + \sum_{j=n+1}^l g(s_j)$$
(7)

We call the right-most summation in (7) the *cost-to-go*. Notice that the first summation on the right-hand side of (7) represents the value of the portion of a path that can be computed optimally and the *cost-to-go* represents the value of the remainder of the path that is normally ignored during a planning step when planning a receding horizon path. Therefore, the value of the optimal *l*-length path may be represented as

$$J_l^*(s_0) = \max_{\gamma_n(s_0) \in \Gamma_n(s_0)} \left( J_n(s_0) + J_{l-n}^*(s_n) \right)$$
(8)

Because we assume that the search agent has only sufficient computational resources to compute n-length optimal

paths, we assume that there exists a lower bound on the *cost-to-go*.

Definition 3.1: A lower bound on the cost-to-go in 7 is the function  $B_l(s_n)$  that satisfies

$$B_l(s_n) \le J_{l-n}^*(s_n) \tag{9}$$

for all  $n \leq l$  with  $B_l(s_l) = 0$ , and for the infinite case where  $l = \infty$ ,  $B_l(s_k) \to \infty$  as  $k \to \infty$ .

We assume that the lower bound on the *cost-to-go* may be efficiently computed in real-time. Using this lower bound on the *cost-to-go*, we define the value of an *n*-length path as

$$\mathcal{V}(\gamma_n(s_0)) = \sum_{i=1}^n g(s_i) + B_l(s_n)$$
(10)

For notational simplicity we define

$$\mathcal{V}_n(s_0) \triangleq \mathcal{V}(\gamma_n(s_0)) \tag{11}$$

The optimal path with respect to (10) is now is now defined as

$$\gamma_n^*(s_0) = \underset{\gamma_n(s_0) \in \Gamma_n(s_0)}{\arg \max} \mathcal{V}_n(s_0) \tag{12}$$

with value

$$\mathcal{V}_n^*(s_0) = \max_{\gamma_n(s_0) \in \Gamma_n(s_0)} \mathcal{V}_n(s_0)$$
(13)

Notice the similarity between (10) and (8). The intuition behind (10) is the following. If the lower bound on the *costto-go* may be found efficiently and approximates the value of the optimal *cost-to-go*, then the value of an *n*-length path with an appended terminal cost approximates the value of the optimal *l*-length path. In practice though, it may be difficult to efficiently approximate the value of the optimal *cost-togo*. As such, we do not seek to plan optimal paths, but instead we seek to provide a lower bound on the value of receding horizon search paths that approximates the value of the optimal *l*-length path as nearly as is feasible.

## IV. SUB-OPTIMALITY GUARANTEES FOR RECEDING HORIZON PATH PLANNING WITH TERMINAL COSTS

In prior work [1] it was shown that appropriate use of a terminal cost when planning receding horizon search paths can guarantee a lower bound on the value of the receding horizon search paths given that the path produced at each planning step is optimal. We extend the main result from [1] to the case that at each planning step the path produced is *nearly optimal*. We also consider a multi-step execution horizon *m* instead of the single step execution horizon considered in [1]. One motivation for using an *m*step execution horizon is that some optimization methods such as Monte Carlo Tree Search can generate better paths when the planning time is longer. In other words, allowing the search agent to complete multiple state transitions while planning may allow for extended planning horizons that may in turn benefit search performance. In Proposition 4.1, we state a condition under which the value of a receding horizon path constructed from nearoptimal paths is always bounded below by the lower bound on the *cost-to-go*. We say that a path  $\gamma'_n(s_k)$  is *nearly optimal* if the value of the path satisfies

$$\mathcal{V}_n'(s_k) \ge \mathcal{V}_n^*(s_k) - \epsilon \tag{14}$$

for some  $\epsilon \geq 0$ .

Proposition 4.1: Suppose the *l*-length receding horizon path  $P_l$  is composed of *m*-step sequences along a sequence of *n*-length near-optimal paths that seek to maximize (10) satisfying (14). For every *n*-length near-optimal path

$$\gamma'_n(s_k) = \{s_k, s'_{k+1}, \dots, s'_{k+n}\}$$
(15)

suppose that there exists a set of states  $\{\tilde{s}_{k+n+1}, \ldots, \tilde{s}_{k+n+m}\}$  that are feasible from  $s'_{k+n}$  such that

$$B_{l}(s'_{k+n}) + \epsilon \le \sum_{i=k+n+1}^{k+m} g(\tilde{s}_{i}) + B_{l}(\tilde{s}_{k+n+m})$$
(16)

then the RH path satisfies

$$J(\boldsymbol{P}_l) \ge B_l(s_0) \tag{17}$$

As  $\epsilon$  increases, the existence of a set of feasible locations satisfying (16) may be a difficult requirement to satisfy. Therefore, in Proposition 4.2 we state a condition under which the value of a receding horizon path is always no less than the lower bound on the *cost-to-go* from the initial state along the *l*-length receding horizon path. In Corollary 4.3 we show that the hypothesis of Proposition 4.2 can always be satisfied given a terminal cost defined by the best path in a set of readily available naive paths, such as mowing-the-lawn paths that are used often for subsea applications. Corollary 4.3 also satisfies (16) given  $\epsilon = 0$ .

Proposition 4.2: Suppose that at each planning step the path  $\gamma'_n(s_k)$  satisfies

$$\mathcal{V}_n'(s_k) \ge \mathcal{V}_{n-m}'(s_k) \tag{18}$$

where  $\mathcal{V}'_{n-m}(s_k)$  is the value of the n-m steps not included in the execution horizon of  $\gamma'_n(s_{k-1})$  and that

$$\mathcal{V}_n'(s_0) \ge B_l(s_0) \tag{19}$$

then the RH path satisfies

$$J(\mathbf{P}_l) \ge B_l(s_0) \tag{20}$$

If the best path in a set of readily available, naive paths is used to determine the lower bound on the *cost-to-go* then there always exists an *n*-length path at the next planning step satisfying (18). Specifically (18) is satisfied because the first m steps along the best naive path in the set of naive paths used to determine the lower bound on the *cost-to-go* may be appended to the remainder of the optimal *n*-length path from the previous planning step guaranteeing the existence of an *n*-length path satisfying (18).

Let  $\Gamma_{l-k-n}^{K}(s_{k+n})$  be a set of K feasible naive paths such as a set of mowing-the-lawn paths that are commonly used in subsea applications.

Corollary 4.3: Given  $\Gamma_{l-k-n}^{K}(s_{k+n})$ , let the lower bound on the cost-to-go be defined

$$B_{l}(s_{k+n}) = \max_{\tilde{\gamma}_{l-k-n}(s_{k+n}) \in \Gamma_{l-k-n}^{K}(s_{k+n})} J(\tilde{\gamma}_{l-k-n}(s_{k+n}))$$
(21)

Then the inequality

$$B_{l}(\tilde{s}_{k+n}) \leq \sum_{i=k+n+1}^{k+n+m} g(\tilde{s}_{i}) + B_{l}(s_{k+n+m})$$
(22)

is always satisfied.

The main benefit of Corollary 4.3 is the ability to leverage strengths of distinct naive paths. For example, a greedy receding horizon planner is computationally efficient. Therefore, an unbiased path such as a lawnmower path may be used alongside a greedy planner that incorporates a terminal cost to determine a lower bound on the *cost-to-go* from a given state. Leveraging the strengths of several naive paths then allows the receding horizon planner to more closely approximate the value of (8) than would normally be possible with a single naive path. This does not imply optimality of the receding horizon path. Instead an improved lower bound on the costto-go guarantees an improved lower bound on the value of the receding horizon path produced.

#### V. NUMERICAL EXPERIMENTS

In order to verify the sub-optimality guarantees presented in propositions 4.1 and 4.2, we employ a probabilistic implementation of receding horizon control using Monte Carlo Tree Search (MCTS). MCTS has been used to address a wide variety of problems [20]. Theoretical analysis of the classical upper confidence bounds applied to trees (UCT) algorithm is provided in [21], [22]. The UCT algorithm is designed to balance exploitation of best actions found so far and exploration of other actions in an effort to ensure that the algorithm has a small probability of choosing a sub-optimal action if stopped prematurely and, given enough time, will determine the best action. To estimate the value of a state associated with a new leaf node the MCTS algorithm utilizes a finite set of Monte Carlo simulations to provide an expected value.

In this work, we utilize a basic implementation of MCTS with minor modifications to highlight the effect of appending a terminal cost to a receding horizon path. First, the UCT is computed by normalizing the estimated reward to a factor of a naively computed upper bound,

$$\frac{\mathbb{E}}{B_u(s_o)} + C\sqrt{\frac{\ln(N)}{n}} \tag{23}$$

where in this case we compute  $B_u(s_o)$  by summing the lhighest value cells at the start of the mission. Second, instead of back-propagating the average reward of a Monte Carlo

0.6 0.5 0.4 0.3 0.2 0.1 Fig. 1: 150 x 150 cell sensor performance map of an envi-

ronment measured in the approaches to Boston Harbor using side scan sonar on-board an unmanned underwater vehicle. The data corresponds to an environmental characterization metric which helps define how well the sonar will perform during search and characterization missions [24]

simulation, we back-propagate the maximum value. Lastly, in the experiments the search agent has three available actions: move forward, turn left, turn right. While there is no incurred penalty while performing a turn, the search agent does not sample the cell when turning. This is to more closely mirror the real-world application of subsea search where a search sensor is less effective when not traveling in straight line paths. To account for this in the MCTS planner, the roll-out policy performs a biased sampling of random actions where forward is sampled at a rate of 50% and the other two actions are sampled both at 25%. We then compare the results of a receding horizon MCTS planner with terminal costs to a brute-force implementation which finds the optimal solution to (12) using the same mission length, horizon, and execution values: l, n, and m, respectively. Furthermore, we compare the results to both MCTS and brute force implementations where a terminal cost is not appended.

Throughout these numerical experiments, we leverage strengths of simple lawnmower paths in the context of subsea search [23] to compute a lower bound on the cost-to-go from a given state. In contrast to prior work [1] where a horizontal lawnmower path was used to determine a lower bound on the cost-to-go, we use the higher reward between a horizontal lawnmower path and a vertical lawnmower path starting from a given state.

Figure 1 is the environment the search agent is tasked with exploring. The map shows the reward for sampling a cell given a characterization metric for a subsea search sensor,





Fig. 2: Comparison of the resulting total accumulated reward for each of the planning approaches described in this paper where the mission length l is varied.

the data is real data collected from a 21" bluefin UUV in the approaches to Boston Harbor (see [24] for a detailed explanation on the performance estimation metric). Using this environment we compare the planning results from receeding horizon planners that utilize a n-step receeding horizon with n = 6, and an m-step execution horizon of m = 3. The MCTS performed 20 simulations per leaf node evaluation and was given 4 seconds per step to plan (we planned for a total of 12 seconds per cycle since 3 steps were performed each cycle). For each experiment the starting position of the search agent was randomly selected from one of the four corners of the map. The mission length was set to values ranging from 500 steps to 5000 steps. MCTS (MCTS-RH-TC) experiments were conducted 40 times while the bruteforce (BF-RH-TC) method was conducted 4 (one for each corner, the solution is deterministic so no additional trials are required). We compare the results of a receeding horizon planner with the terminal cost appended to the n-step path to a planner not utilizing a terminal cost (BF-RH and MCTS-RH, brute force and MCTS respectively). We also compare the results to the initial lower-bound computation (LB, the better of two naive lawnmower patterns starting from the initial position) as well as a MCTS planner that searches for the full-length path (otherwise using the same parameters as the MCTS-RH and MTCS-RH-TC planners). Figure 2 shows the results of the experiments, figure 3 show the results for l at 5000, namely to highlight how close the MCTS results are able to mirror the optimal BF results. Figure 4a and 4b show two representative trials of the MCTS planners with and without the appended terminal cost respectively.

### VI. CONCLUSIONS

We present theoretical extensions to prior work [1]. We also present an easily satisfied sufficient condition to guarantee lower bounds on the value of receding horizon paths



Fig. 3: Results for the presented planners where the length l is set to 5000.

constructed with terminal costs. We demonstrate the efficacy of the proposed methods in the context of subsea search by performing numerical experiments with real-world data. Specifically, we demonstrate that the best solution from a set of readily-available, naive solutions may be used to set a desirable lower bound on the value of receding horizon search paths. We also demonstrate that the receding horizon search planner is capable of producing paths with rewards significantly higher than the lower bound. Future work will address extending the results presented in this paper to multivehicle search.

A limitation of the contributions presented in this work is a dependence on the existence of a naive paths that ensure good search performance. This will likely be of particular significance in extending this work to multi-vehicle search. In the case of subsea search with a single agent, lawnmower paths provide a desirable lower bound.

## APPENDIX A PROOF OF PROPOSITION 4.1

*Proof:* Consider the (n + m)-length path  $\tilde{\gamma}_n(s_k) = \{\gamma'_n(s_k), \tilde{s}_{k+n+1}, \ldots, \tilde{s}_{k+n+m}\}$  composed of an *n*-length near-optimal path  $\gamma'(s_k) = \{s_k, s'_{k+1}, \ldots, s'_{k+n}\}$  and a feasible set of states  $\{\tilde{s}_{k+n+1}, \ldots, \tilde{s}_{k+n+m}\}$  satisfying (16). Then from equation (10) the value of  $\tilde{\gamma}_n(s_k)$  is

$$\tilde{\mathcal{V}}_{n+m}(s_k) = \sum_{i=k+1}^{k+n} g(s'_i) + \sum_{i=k+n+1}^{k+n+m} g(\tilde{s}_i) + B_l(\tilde{s}_{k+n+m})$$
(24)

or equivalently

$$\tilde{\mathcal{V}}_{n+m}(s_k) = \mathcal{V}'_n(s_k) - B_l(s'_{k+n})$$

$$+ \sum_{i=k+n+1}^{k+n+m} g(\tilde{s}_i) + B_l(\tilde{s}_{k+n+m}).$$
(25)

By hypothesis

$$B_{l}(s_{k+n}') + \epsilon \le \sum_{i=k+n+1}^{k+m} g(\tilde{s}_{i}) + B_{l}(\tilde{s}_{k+n+m})$$
(26)

such that

$$\tilde{\mathcal{V}}_{n+m}(s_k) \ge \mathcal{V}'_n(s_k) + \epsilon,$$
 (27)



Fig. 4: Representative Search Path Using MCTS-RH-TC (a) and MCTS-RH (b). The grayscale values indicate the reward for searching an area. The colored line represents the search path, the start at step 0 (dark blue) and the end at step 2500 (yellow).

and  $\mathcal{V}_n'(s_k)$  is bounded below by  $\mathcal{V}_n^*(s_k) - \epsilon$  yielding

$$\tilde{\mathcal{V}}_{n+m}(s_k) \ge \mathcal{V}_n^*(s_k). \tag{28}$$

We note that (24) may be equivalently written as

$$\tilde{\mathcal{V}}_{n+m}(s_k) = \sum_{\substack{i=k+1\\k+n+m}}^{k+m} g(s'_i) + \sum_{\substack{i=k+m+1\\k+n+m}}^{k+n+m} g(\tilde{s}_i) + B_l(\tilde{s}_{k+n+m}).$$
(29)

That is, there exists some feasible *n*-length path  $\hat{\gamma}_n(s'_{k+m})$  such that

$$\hat{\mathcal{V}}_{n}(s_{k+m}') = \sum_{i=k+m+1}^{k+n} g(s_{i}') + \sum_{i=k+n+1}^{k+n+m} g(\tilde{s}_{i}) + B_{l}(\tilde{s}_{k+n+m})$$
(30)

and

$$\tilde{\mathcal{V}}_{n+m}(s_k) = \sum_{i=1}^m g(s'_i) + \hat{\mathcal{V}}_n(s'_{k+m}).$$
(31)

Replacing left-hand side of (28) with the right-hand side of (31) gives

$$\sum_{i=k+1}^{k+m} g(s_i') + \hat{\mathcal{V}}_n(s_{k+m}') \ge \mathcal{V}_n^*(s_k)$$
(32)

where  $\hat{\gamma}_n(s'_{k+m})$  is not optimal and therefore

$$\sum_{i=k+1}^{k+m} g(s'_i) + \mathcal{V}^*_n(s'_{k+m}) \ge \mathcal{V}^*_n(s_k)$$
(33)

where subtracting  $\mathcal{V}_n^*(s_{k+m}')$  from both sides gives

$$\sum_{i=k+1}^{k+m} g(s'_i) \ge \mathcal{V}_n^*(s_k) - \mathcal{V}_n^*(s'_{k+m}).$$
(34)

Evaluating (34) at different values of k from k = 1 to k = l - m in increments of m gives

$$\sum_{i=1}^{m} g(s'_i) \ge \mathcal{V}_n^*(s_0) - \mathcal{V}_n^*(s'_m)$$
(35)

$$\sum_{i=m+1}^{2m} g(s_i') \ge \mathcal{V}_n^*(s_m) - \mathcal{V}_n^*(s_{2m}')$$
(36)

$$\sum_{i=l-m+1}^{l} g(s'_i) \ge \mathcal{V}_n^*(s_{l-m}) - \mathcal{V}_n^*(s'_l).$$
(37)

And summing the inequalities in (35) gives

$$\sum_{i=1}^{l} g(s'_{i}) \geq \mathcal{V}_{n}^{*}(s_{0}) - \mathcal{V}_{n}^{*}(s'_{m}) + \mathcal{V}_{n}^{*}(s_{m}) - \qquad (38)$$
$$\mathcal{V}_{n}^{*}(s'_{2m}) + \ldots + \mathcal{V}_{n}^{*}(s_{l-m}) - \mathcal{V}_{n}^{*}(s'_{l}).$$

Which, because in receding horizon path planning  $s_m=s_m^\prime,s_{2m}=s_{2m}^\prime$  and so on, reduces to

$$\sum_{i=1}^{l} g(s_i') \ge \mathcal{V}_n^*(s_0) - \mathcal{V}_n^*(s_l').$$
(39)

We note that the summation on the left-hand side of (39) represents the value of the *l*-length receding horizon path  $P_l$  and that  $\mathcal{V}_n^*(s_l') = 0$ . Using these observations, we rewrite (39) as

$$J(\mathbf{P}_l) \ge \mathcal{V}_n^*(s_0). \tag{40}$$

Finally, because  $\mathcal{V}_n^*(s_0) \geq B_l(s_0)$ , we conclude that

$$J(\mathbf{P}_l) \ge B_l(s_0). \tag{41}$$

## APPENDIX B PROOF OF PROPOSITION 4.2

*Proof:* From (10), the reward attained while traversing the first m-steps along a path may be represented by

$$\sum_{i=k+1}^{k+m} g(s'_i) = \mathcal{V}'_n(s_k) - \sum_{i=k+m+1}^{k+n} g(s'_i) - B_l(s'_{k+n}).$$
(42)

Where  $\sum_{i=k+m+1}^{k+n} g(s'_i) + B_l(s'_{k+n})$  is the value of an (n-m)-length path remaining of  $\gamma'_n(s_k)$  after traversing the first m steps. We use the notation

$$\mathcal{V}'_{n-m}(s'_{k+m}) = \sum_{i=k+m+1}^{k+n} g(s'_i) + B_l(s'_{k+n})$$
(43)

and write

$$\sum_{i=k+1}^{k+m} g(s_i') = \mathcal{V}'_n(s_k) - \mathcal{V}'_{n-m}(s_{k+m}').$$
(44)

As in the proof for 4.1, we evaluate (44) at different values of k from k = 1 to k = l - m in increments of m to get

$$\sum_{i=1}^{m} g(s'_i) = \mathcal{V}'_n(s_0) - \mathcal{V}'_{n-m}(s'_m)$$
(45)

:

$$\sum_{i=m+1}^{2m} g(s_i') = \mathcal{V}'_n(s_m) - \mathcal{V}'_{n-m}(s_{2m}')$$
(46)

$$\sum_{i=l-m+1}^{l} g(s_i') = \mathcal{V}'_n(s_{l-m}) - \mathcal{V}'_{n-m}(s_l')$$
(47)

where summing the inequalities in (45) gives

$$\sum_{i=1}^{l} g(s'_{i}) = \mathcal{V}'_{n}(s_{0}) - \mathcal{V}'_{n-m}(s'_{m}) + \mathcal{V}'_{n}(s_{m}) - (48)$$
$$\mathcal{V}'_{n-m}(s'_{2m}) + \ldots + \mathcal{V}'_{n}(s_{l-m}) - \mathcal{V}'_{n-m}(s'_{l}).$$

By hypothesis 
$$\mathcal{V}'_n(s_k) \geq \mathcal{V}'_{n-m}(s_k)$$
. Therefore, because in receding horizon path planning  $s_m = s'_m, s_{2m} = s'_{2m}$  and so on, (48) reduces to

$$\sum_{i=1}^{l} g(s'_i) \ge \mathcal{V}'_n(s_0) - \mathcal{V}'_{n-m}(s'_l).$$
(49)

We note that the summation on the left-hand side of (49) represents the value of the *l*-length receding horizon path  $P_l$  and that  $\mathcal{V}'_{n-m}(s'_l)$ . Using these observations, we rewrite (49) as

$$J(\mathbf{P}_l) \ge \mathcal{V}'_n(s_0),\tag{50}$$

and from (19) we conclude that

$$J(\mathbf{P}_l) \ge B_l(s_0). \tag{51}$$

#### APPENDIX C Proof of Corollary 4.3

Proof: Let  $\tilde{\gamma}^*_{l-k-n}(s_{k+n})$  be a path in  $\Gamma^K_{l-k-n}(s_{k+n})$  satisfying

$$\tilde{\gamma}_{l-k-n}^*(s_{k+n}) = \underset{\tilde{\gamma}_{l-k-n}(s_{k+n}) \in \Gamma_{l-k-n}^K(s_{k+n})}{\arg\max} J(\tilde{\gamma}_{l-k-n}(s_{k+n})$$
(52)

such that

$$B_l(s_{k+n}) = J(\tilde{\gamma}_{l-k-n}^*(s_{k+n})).$$
(53)

We recall that

$$J(\tilde{\gamma}_{l-k-n}^{*}(s_{k+n})) = \sum_{j=k+n+1}^{l} g(\tilde{s}_{j})$$
(54)

and note that

$$\sum_{j=k+n+1}^{l} g(\tilde{s}_j) = \sum_{i=k+n+1}^{k+n+m} g(\tilde{s}_i) + \sum_{j=k+n+m+1}^{l} g(\tilde{s}_j).$$
(55)

Given that  $B_l(\tilde{s}_{k+n})$  is defined by the right-most summation in (54), we write

$$B_l(\tilde{s}_{k+n}) = \sum_{i=k+n+1}^{k+n+m} g(\tilde{s}_i) + \sum_{j=k+n+m+1}^l g(\tilde{s}_j).$$
 (56)

We note that  $\sum_{j=k+n+m+1}^{l} g(\tilde{s}_j)$  represents the value of a single path in the set of available paths used to define the lower bound on the *cost-to-go* and as such may not represent the highest valued naive path such that

$$\sum_{i=k+n+2}^{l} g(\tilde{s}_j) \le B_l(s_{k+n+m}).$$
(57)

Using (57) and (56) we write

$$B_{l}(\tilde{s}_{k+n}) \le g(\tilde{s}_{k+n+1}) + \ldots + g(\tilde{s}_{k+n+m}) + B_{l}(s_{k+n+m})$$
(58)

which concludes the proof.

#### REFERENCES

- B. Biggs, D. J. Stilwell, H. Yetkin, and J. McMahon, "Performance guarantees for receding horizon search with terminal cost," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (iROS), IEEE, 2019.
- [2] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [3] J. H. Lee, "Model predictive control: Review of the three decades of development," *International Journal of Control, Automation and Systems*, vol. 9, no. 3, p. 415, 2011.
- [4] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
- [5] G. de Nicolao, L. Magni, and R. Scattolini, "On the robustness of receding-horizon control with terminal constraints," *IEEE Transactions* on Automatic Control, vol. 41, no. 3, pp. 451–453, 1996.
- [6] J. A. Primbs and V. Nevistić, "Feasibility and stability of constrained finite receding horizon control," *Automatica*, vol. 36, no. 7, pp. 965 – 971, 2000.
- [7] A. Jadbabaie and J. Hauser, "On the stability of receding horizon control with a general terminal cost," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 674–678, 2005.

- [8] S. Liu and J. Liu, "Economic model predictive control with extended horizon," *Automatica*, vol. 73, pp. 180–192, 2016.
- [9] J. Bellingham, A. Richards, and J. P. How, "Receding horizon control of autonomous aerial vehicles," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 5, pp. 3741–3746, IEEE, 2002.
- [10] Y. Kuwata and J. How, "Three dimensional receding horizon control for uavs," in AIAA Guidance, Navigation, and Control Conference and Exhibit, p. 5144, 2004.
- [11] B. Mettler and O. Toupet, "Receding horizon trajectory planning with an environment-based cost-to-go function," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 4071–4076, IEEE, 2005.
- [12] M. P. Vitus, W. Zhang, A. Abate, J. Hu, and C. J. Tomlin, "On efficient sensor scheduling for linear dynamical systems," *Automatica*, vol. 48, no. 10, pp. 2482–2493, 2012.
- [13] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 6447–6454, IEEE, 2014.
- [14] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot slam," in 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 4775–4782, IEEE, 2015.
- [15] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1025–1032, 2018.
- [16] S.-H. Yoo, A. Stuntz, Y. Zhang, R. Rothschild, G. A. Hollinger, and R. N. Smith, "Experimental analysis of receding horizon planning algorithms for marine monitoring," in *Proc. Tenth International Conference* on Field and Service Robotics (FSR), vol. 113, pp. 31 – 44, 2016.

- [17] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.
- [18] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically optimal planning for non-myopic multi-robot information gathering," *Proceedings of Robotics: Science and Systems (RSS)*, *Freiburg, Germany*, pp. 22–26, 2019.
- [19] H. Yetkiny, J. McMahon, N. Topin, A. Wolek, Z. Waters, and D. J. Stilwell, "Online planning for autonomous underwater vehicles performing information gathering tasks in large subsea environments," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6354–6361, IEEE, 2019.
- [20] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.
- [21] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *European conference on machine learning*, pp. 282–293, Springer, 2006.
- [22] L. Kocsis, C. Szepesvári, and J. Willemson, "Improved monte-carlo search," Univ. Tartu, Estonia, Tech. Rep, vol. 1, 2006.
- [23] S. Frolov, B. Garau, and J. Bellingham, "Can we do better than the grid survey: Optimal synoptic surveys in presence of variable uncertainty and decorrelation scales," *Journal of Geophysical Research: Oceans*, vol. 119, no. 8, pp. 5071–5090, 2014.
- [24] J. McMahon, H. Yetkin, A. Wolek, Z. J. Waters, and D. J. Stilwell, "Towards real-time search planning in subsea environments," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 87–94, IEEE, 2017.