Verification and Synthesis of Admissible Heuristics for Kinodynamic Motion Planning

Brian Paden, Valerio Varricchio, and Emilio Frazzoli

***Abstract*—How does one obtain an admissible heuristic for a kinodynamic motion planning problem? This letter develops the analytical tools and techniques to answer this question. A sufficient condition for the admissibility of a heuristic is presented, which can be checked directly from problem data. This condition is also used to formulate an infinite-dimensional linear program to optimize an admissible heuristic. We then investigate the use of sum-of-squares programming techniques to obtain an approximate solution to this linear program. A number of examples are provided to demonstrate these new concepts.**

***Index Terms*—Admissible heuristics, convex optimization, kinodynamic motion planning.**

I. Introduction

Many graph search problems arising in robotics and artificial intelligence that would otherwise be intractable can be solved efficiently with an effective heuristic informing the search. However, efficiently obtaining a shortest path on a graph requires the heuristic to be admissible as described in the seminal letter introducing the A^* algorithm [1]. In short, an admissible heuristic provides an estimate of the optimal cost to reach the goal from every vertex, but never overestimates the optimal cost.

A major application for admissible heuristics is in searching graphs approximating robotic motion planning problems. The workhorse heuristic in kinematic shortest path problems is the Euclidean distance from a given state to the goal. This heuristic is admissible irrespective of the obstacles in the environment. Fig. 1 demonstrates the use of this heuristic on a typical shortest path problem where informing the search reduces the number of iterations required to find the goal by 84% in comparison to a uniform cost search.

More recently, methods have been developed for generating graphs approximating optimal trajectories in kinodynamic motion planning problems. Notable examples include the kinodynamic variant of the RRT^* algorithm [2], the state-augmentation technique proposed in [3], and the GLC algorithm [4]. While this is not a comprehensive literature review on optimal kinodynamic motion planning, the use of admissible heuristics has been proposed for each of these methods (the use of heuristics for RRT^* was proposed recently in [5], [6]). The kinodynamic motion planning problem and the use of admissible heuristics are reviewed in Sections II and III respectively.

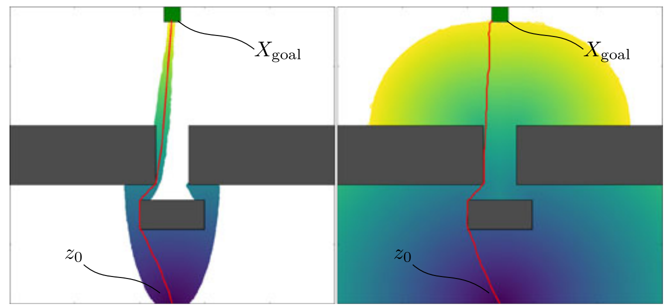

Fig. 1. Comparison of a shortest path search in a graph approximating continuous paths in a 2D environment. The heuristically guided search (left) obtains the solution in 216,602 iterations while the uniform cost search (right) requires 1,313,374 iterations. Each vertex evaluated is illustrated by a marker colored according to the cost to reach that vertex.

A good heuristic is one which closely underestimates the optimal cost-to-go from every vertex to the goal. This enables a larger number of provably suboptimal paths to be identified and discarded from the search. While admissibility of a heuristic is an important concept it gives rise to two challenging questions: (i) Without a priori knowledge of the optimal cost-to-go, how do we verify the admissibility of a candidate heuristic? (ii) How do we systematically construct good heuristics for kinodynamic motion planning problems?

The first question is addressed in Section III where a sufficient condition for the admissibility of a candidate heuristic is presented. This condition takes the form of an affine inequality involving the heuristic and given problem data. The result provides a general analytical tool for validating a heuristic constructed by intuition about the problem.

The second question is addressed in Section IV where the admissibility condition is used to formulate a linear program over the space of candidate heuristics. The objective of the optimization is constructed so that the optimal cost-to-go is a globally optimal solution to the optimization. The approach to analyzing and constructing admissible heuristics is inspired from the dual formulation to the trajectory optimization problem [7], [8].

Manuscript received September 7, 2016; accepted December 29, 2016. Date of publication January 10, 2017; date of current version January 26, 2017. This letter was recommended for publication by Associate Editor D. Halperin and Editor N. Amato upon evaluation of the reviewers' comments. This work was supported in part by ARO MURI under Grant W911NF1110046 and in part by the Swiss Federal Institute of Technology.

B. Paden and E. Frazzoli are with the Institute for Dynamic Systems and Control, Swiss Federal Institute of Technology, Zurich 8006, Switzerland (e-mail: padenb@ethz.ch; emilio.frazzoli@ethz.ch).

V. Varricchio is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: valerio@mit.edu).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2017.2651157

Section V introduces an approach to solving the convex optimization of Section IV. Sum-of-squares (SOS) programming techniques [9] are used to obtain an approximate solution on a finite-dimensional subspace of polynomials. In doing so we take the first steps towards a general procedure for computing admissible heuristics to kinodynamic motion planning problems.

The results of this letter are illustrated in Section VI where the admissibility of a number of heuristics for practical robot models are verified using the admissibility condition of Section III. The SOS approximation to the heuristic optimization is also illustrated in Section VI with a numerical example. The YALMIP [10] scripts used to compute the example heuristics can be found in [11].

II. Kinodynamic Motion Planning

Consider a system whose state at time $t \in \mathbb{R}$ is described by a vector in $\mathbb{R}^n$. A *trajectory* x, representing a time evolution of the system state, is a continuous map from a closed time domain $[0,T]$ to the state space; $x : [0,T] \to \mathbb{R}^n$ for a $T > 0$.

A trajectory in a kinodynamic motion planning problem must satisfy several point-wise constraints. First, a subset $X_{\text{free}} \subset \mathbb{R}^n$ of the state space encodes the set of allowable states over the entire domain of the trajectory; $x(t) \in X_{\text{free}}$ for all $t \in [0,T]$. Secondly, there is an initial state constraint, $x(0) = z_0$ for a state $z_0 \in X_{\text{free}}$. Lastly, there is a terminal constraint; $x(T) \in X_{\text{goal}}$ for a subset $X_{\text{goal}} \subset X_{\text{free}}$.

In addition to the point-wise constraints, the trajectory must satisfy differential constraints. At each time t the system is affected by a control action $u(t)$. The set of available control actions is a subset Ω of $\mathbb{R}^m$. The time history of control actions is referred to as a *control signal*. The control signal is assumed to be Lebesgue integrable and essentially bounded. The control action affects the trajectory through the differential equation,

$$\dot{x}(t) = f(x(t), u(t)), \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. A trajectory x with domain $[0,T]$ must satisfy (1) for some control signal u almost everywhere (a.e.) on $t \in [0,T]$. A *feasible* trajectory is one that satisfies these point-wise and differential constraints.

Notation: To distinguish between states and trajectories, the symbol z will denote a state while x will denote a trajectory; i.e. z and $x(t)$ are elements of $\mathbb{R}^n$, but x is a map into $\mathbb{R}^n$. Similarly, w will denote a control action while u will denote a control signal. The symbol μ is reserved for the Lebesgue measure over the reals while m denotes any finite, positive measure with support on a subset of X_{free}.

Next, a cost functional J provides a way to quantify the merit of a candidate trajectory and control signal,

$$J(x,u) = \int_{[0,T]} g(x(t), u(t))\ \mu(dt). \tag{2}$$

It is assumed $g(z,w) \geq 0$ for all $z \in X_{\text{free}}$ and $w \in \Omega$ so that a nonnegative running cost is associated to each state-action pair. For brevity, *problem data* refers to z_0, X_{free}, X_{goal}, Ω, f, and g defining an instance of a kinodynamic motion planning problem as described above.

A solution to an optimal kinodynamic motion planning problem is a feasible trajectory and control signal which minimizes (2). While the minimum of (2) may not be attained, the greatest lower bound of the cost functional from the initial state z_0 is always well defined.

A. The Value Function

The optimal cost-to-go or value function $V : X_{\text{free}} \to \mathbb{R}$ describes the greatest lower bound on the cost to reach the goal set from the initial state $z_0 \in X_{\text{free}}$ That is

$$V(z_0) = \inf_{x,u} \{J(x,u)\}, \tag{3}$$

where x is a feasible trajectory with $x(0) = z_0$. The following properties of V follow immediately from the assumption $g(z,w) \geq 0$ in (2):

$$\begin{aligned} V(z) &\geq 0 \qquad \forall z \in X_{\text{free}}, \\ V(z) &= 0 \qquad \forall z \in X_{\text{goal}}. \end{aligned} \tag{4}$$

If the value function V is differentiable, it is a classical[1] solution to the Hamilton-Jacobi-Bellman (HJB) equation,

$$\begin{gathered} \inf_{w \in \Omega} \{\langle \nabla V(z), f(z,w) \rangle + g(z,w)\} = 0, \\ \forall z \in X_{\text{free}} \setminus \bar{X}_{\text{goal}}, \end{gathered} \tag{5}$$

with the boundary condition $V(z) = 0$ for all z in the closure of X_{goal} (denoted $\bar{X}_{\text{goal}}$). Likewise, if the HJB equation admits a classical solution, then it is equal to the value function.

III. Graph-Search Oriented Approximations

Many computational methods for solving the kinodynamic motion problem approximate the set of all possible trajectories by a finite directed graph $(\mathcal{V}, E)$, whose vertices are states in the state space, and whose edges correspond to trajectories between two vertices satisfying (1). Conceptually, shortest paths on the graph are in some sense faithful approximations of optimal feasible trajectories for the problem.

The non-negativity of the cost function (2) enables a nonnegative edge-weight to be assigned to each edge corresponding to the cost of the trajectory in relation with that edge. The approximated problem can then be addressed using shortest path algorithms for graphs.

The value function $\hat{V} : \mathcal{V} \to \mathbb{R}$ on the weighted graph is analogous to the value function V in the original problem. For a vertex z in the graph, $\hat{V}(z)$ is the cost of a shortest path to one of the goal vertices: $\mathcal{V} \cap X_{\text{goal}}$. Since the feasible trajectories represented by the graph are a subset of the feasible trajectories of the problem we have the inequality

$$V(z) \leq \hat{V}(z) \qquad \forall z \in \mathcal{V}. \tag{6}$$

A. Admissible Heuristics

To carry out an informed search and ensure the optimality of the result, many algorithms require an admissible heuristic

[1] In some cases the value function is not differentiable in which case a generalized solution concept is used [12].

$H : X_{\text{free}} \to \mathbb{R}$. A heuristic H for a problem with value function V is *admissible* if

$$H(z) \leq V(z) \qquad \forall z \in X_{\text{free}}. \tag{7}$$

In light of (6), an admissible heuristic for the kinodynamic motion planning problem will also be admissible for any graph-based approximation to the problem. For the remainder, the set of candidate heuristics will be restricted to differentiable scalar valued functions on X_{free}.

Since the value function is unknown it is difficult to check that (7) is satisfied for a particular heuristic H. This motivates the first contribution of this letter, a sufficient condition for admissibility that can be checked directly from the problem data.

Lemma 1 (Admissibility): A heuristic H is an admissible heuristic if the following two conditions are satisfied:

$$H(z) \leq 0 \qquad \forall z \in X_{\text{goal}}, \tag{AH1}$$

$$\langle \nabla H(z), f(z,w)\rangle + g(z,w) \geq 0, \quad \forall z \in X_{\text{free}}, \text{ and } \forall w \in \Omega. \tag{AH2}$$

Proof: Choose a feasible trajectory x and associated control signal u. By construction $x(T) \in X_{\text{goal}}$ so $H(x(T)) \leq 0$ by (AH1). Then

$$\begin{aligned} H(x(0)) \leq\ & H(x(0)) - H(x(T)) \\ =\ & -\int_0^T \frac{d}{dt} H(x(t))\, \mu(dt) \\ =\ & -\int_0^T \langle \nabla H(x(t)), f(x(t),u(t))\rangle\, \mu(dt) \\ \leq\ & \int_0^T g(x(t), u(t)\, \mu(dt) \\ =\ & J(x,u). \end{aligned} \tag{8}$$

The third step combines the chain-rule of differential calculus with (1). The fourth step of the derivation follows from (AH2). Since $H(x(0)) \leq J(x,u)$ for any feasible trajectory and related control we conclude that H provides a lower bound on the cost-to-go from any initial condition. By definition, the value function V is the greatest lower bound. Thus, we have (7) follows from (AH1) and (AH2). ■

Inequalities of the form (AH1) and (AH2) appear frequently in the optimal control literature where H is considered a smooth sub-solution to the HJB equation.

Observe that Lemma 1 only requires that f and g be integrable. It does not requre V or g to be continuous, nor does f have to be differentiable, and hence is quite general.

An immediate application of this result is as a sufficient condition for the admissibility of a candidate heuristic. Section VI provides three examples demonstrating this technique.

Another concept related to admissibility is *consistency* [1]. This is a stronger property which is also verified by Lemma 1 if (AH1) is replaced by the condition that $H(z) = 0$ for all $z \in X_{\text{goal}}$. The proof can be found in the Appendix.

IV. Optimization of Admissible Heuristics

The second contribution of this letter is a general procedure for computing and optimizing an admissible heuristic. To motivate the proposed optimization we review duality results developed by Fleming [7] and Vinter [8]. Stated informally,[2] the result applied to our problem is as follows:

Theorem (2.1-[8]): Consider the kinodynamic planning problem

$$\begin{aligned} \min_{x,u}\ & \textstyle\int_0^T g(x(t),u(t))\, \mu(dt) \\ \text{subject to: }\ & x(0) = z_0, \\ & x(T) \in X_{\text{goal}}, \\ & x(t) \in X_{\text{free}} \qquad \forall t \in [0,T], \\ & \dot{x}(t) = f(x(t),u(t)) \qquad \text{a.e. } t \in [0,T], \\ & u(t) \in \Omega \qquad \text{a.e. } t \in [0,T]. \end{aligned} \tag{P}$$

The dual problem is

$$\begin{aligned} \max_{H}\ & H(z_0) \\ \text{subject to: }\ & H(z) \leq 0 \qquad \forall z \in X_{\text{goal}}, \\ & \langle \nabla H(z), f(z,w)\rangle + g(z,w) \geq 0 \\ & \forall z \in X_{\text{free}}, \text{ and } \forall w \in \Omega, \end{aligned} \tag{D}$$

and strong duality holds. That is, the optimal values of the two problems coincide.

Since there is no duality gap the optimal value at the initial condition $V(z_0)$ can be obtained by solving the dual problem. Observe that the objective of the dual problem is linear and the constraints are affine making it a linear program.

Problem (D) will not yield a particularly good heuristic since it optimizes the heuristic at a single point. However, it does suggest a related optimization. Instead of optimizing H at a single point, we can optimize the integral of H with respect to any finite measure m on X_{free}. This objective will produce a heuristic which is admissible and takes on the largest possible values on the support of m subject to the admissibility condition.

The integral objective is still linear and the problem remains a (infinite-dimensional) linear program,

$$\boxed{\begin{aligned} \max_{H}\ & \textstyle\int_{X_{\text{free}}} H(z)\ m(dz) \\ \text{subject to: }\ & H(z) \leq 0 \qquad \forall z \in X_{\text{goal}}, \\ & \langle \nabla H(z), f(z,w)\rangle + g(z,w) \geq 0 \\ & \forall z \in X_{\text{free}}, \text{ and } \forall w \in \Omega, \end{aligned}} \tag{LP}$$

Note that (LP) reduces to (D) if a discrete measure concentrated at z_0 is used. Alternatively, a discrete measure results in a discrete sum rather than an integral.

To further justify using the objective in (LP) to optimize our heuristic we show that the value function is the solution when it is differentiable.

Lemma 2: If the value function V is differentiable on $X_{\text{free}} \setminus \bar{X}_{\text{goal}}$, then it solves (LP).

[2] This result requires a relaxed notion of a trajectory and some mild technical assumptions on the problem data; cf [8] for details.

Proof: (Feasibility) From (4), $V(z) = 0$ for all $z \in \bar{X}_{\text{goal}}$ so the constraints (AH1) and (AH2) are satisfied on $\bar{X}_{\text{goal}}$. Since V is differentiable, it solves the HJB (5). Thus,

$$\inf_{w\in\Omega} \{\langle \nabla V(z), f(z,w)\rangle + g(z,w)\} = 0, \quad \forall z \in X_{\text{free}} \setminus \bar{X}_{\text{goal}}. \tag{9}$$

This implies

$$\langle \nabla V(z), f(z,w)\rangle + g(z,w) \geq 0, \quad \forall z \in X_{\text{free}} \setminus \bar{X}_{\text{goal}}, \text{ and } w \in \Omega. \tag{10}$$

Therefore, (AH2) is satisfied.

(Optimality) By definition, an admissible heuristic H satisfies $H(z) \leq V(z)$ for all $z \in X_{\text{free}}$. By the non-negativity of m,

$$\int_{X_{\text{free}}} H(z)\ m(dz) \leq \int_{X_{\text{free}}} V(z)\ m(dz). \tag{11}$$

That is, the value function provides an upper bound on the objective in (LP). Since V is a feasible solution, this upper bound is attained and V is therefore an optimal solution. ■

Thus, the HJB equation and (LP) can both be solved to obtain the exact optimal cost-to-go for a problem. While the HJB equation is a nonlinear partial differential equation, the problem (LP) is a linear program lending itself to the methods of convex analysis.

A. Problem Relaxations

In many applications the set X_{free} is not entirely known a priori. This is particularly true when the heuristic is computed off-line and used in a real-time application where a perception system constructs or modifies X_{free} for the current task (e.g. detecting obstacles in a robot workspace).

This consideration motivates the following observation: suppose X_{free}, X_{goal}, Ω, f, and g is problem data for problem P; and $\tilde{X}_{\text{free}}$, $\tilde{X}_{\text{goal}}$, $\tilde{\Omega}$, $\tilde{g}$, and $\tilde{f}$ is problem data for problem $\tilde{P}$. Let V_P and $V_{\tilde{P}}$ denote the optimal cost-to-go for each of these problems.

If the two problems are related by

$$cX_{\text{free}} \subset \tilde{X}_{\text{free}}, \quad X_{\text{goal}} \subset \tilde{X}_{\text{goal}}, \quad \Omega \subset \tilde{\Omega}, \quad f = \tilde{f}, \\ g(z,w) \geq \tilde{g}(z,w) \quad \forall z \in X_{\text{free}}, \text{ and } w \in \Omega, \tag{12}$$

then any feasible trajectory of problem P must also be feasible for problem $\tilde{P}$. Additionally, this trajectory will have the same or lesser cost for $\tilde{P}$. Therefore, $V_{\tilde{P}}(z) \leq V_P(z)$ for any $z \in X_{\text{free}}$. We can conclude that an admissible heuristic $H_{\tilde{P}}$ for problem $\tilde{P}$ must also be admissible for problem P since $H_{\tilde{P}}(z) \leq V_{\tilde{P}}(z)$ implies $H_{\tilde{P}}(z) \leq V_P(z)$. Problem $\tilde{P}$ is referred to as a *relaxation* of problem P.

In the case of an unknown environment, one can derive an admissible heuristic for a relaxed problem which considers only constraints known in advance. This heuristic remains admissible if X_{free} is updated to a smaller set due to perceived obstacles. Alternatively, it may be easier to verify the admissibility of a candidate heuristic with (AH1) and (AH2), or evaluate (LP) on a relaxation of a particular problem. This comes at the expense of increasing the gap between the heuristic and the optimal cost-to-go for the actual problem which can make the heuristic less effective for a search-based algorithm.

V. Sum-of-Squares (SOS) Approximation to (LP)

One way to tackle (LP) in the case of problem data consisting of semi-algebraic sets and polynomials is a SOS programming approximation.

SOS programming [9] is a method of optimizing a functional of a polynomial subject to semi-algebraic constraints. The technique involves replacing semi-algebraic constraints with sum-of-squares constraints which can then be solved as a semi-definite program (SDP). The advantage of this particular approach to approximating (LP) is that the result of the SOS program is guaranteed to satisfy the admissibility constraint.

A. Sum-of-Squares Polynomials

A polynomial p in the ring $\mathbb{R}[z]$ in n variables is said to be a sum-of-squares if it can be written as

$$p(z) = \sum_{k=1}^{d} q_k(z)^2, \tag{13}$$

for polynomials $q_k(z)$. Clearly, $p(z) \geq 0$ for all $z \in \mathbb{R}^n$. Note also that $p(z)$ is a sum-of-squares if and only if it can be written as

$$p(z) = \mathfrak{m}(z)^T Q \mathfrak{m}(z), \tag{14}$$

for a positive semi-definite matrix Q and the vector of monomials $\mathfrak{m}(z)$ up to degree d. For a polynomial p admitting a decomposition of the form (14) we write $p \in SOS$.

Equation (14) is a collection of linear equality constraints between the entries of Q and the coefficients of $p(z)$. Finding entries of a positive semi-definite Q such that the equality constraints are satisfied is then a semi-definite program (SDP). The complexity of finding a solution to this problem using interior-point methods is polynomial in the size of Q.

This method of analyzing polynomial inequalities has had a profound impact in many fields. As a result there are a number of efficient solvers [13], [14] and modeling tools [10], [15] available.

B. Optimizing the Heuristic

To proceed with computing a heuristic using the SOS programming framework the problem data must consist of polynomials and intersections of semi-algebraic sets. Let

$$X_{\text{free}} = \{z \in \mathbb{R}^n : h_z(z) \geq 0\}, \\ \Omega = \{w \in \mathbb{R}^m : h_w(w) \geq 0\}, \tag{15}$$

for vectors of polynomials h_z and h_w with "$\geq$" denoting element-wise inequalities. Assume also that f, g and the candidate heuristic H are polynomials in the state and control variables. Then the admissibility condition (AH1) is a polynomial inequality. To restrict the non-negativity constraint of the heuristic to X_{free} and Ω, we add auxiliary vectors of SOS polynomials

$\lambda_z(z) \geq 0$ and $\lambda_w(w) \geq 0$ to the equation as

$$\begin{aligned} &\langle \nabla H(z), f(z,w)\rangle + g(z,w) \\ &- \langle \lambda_z(z), h_z(z)\rangle - \langle \lambda_w(w), h_w(w)\rangle \geq 0, \\ &\forall w \in \mathbb{R}^m, \text{ and } z \in \mathbb{R}^n, \end{aligned} \tag{16}$$

which trivially implies the positivity of (AH1) over X_{free} and Ω.

When H is a polynomial, the objective in (LP) is linear in the coefficients of H. Thus, it is an appropriate objective for an SOS program.

The SOS program which is solved to obtain an admissible heuristic is then

$$\begin{aligned} \max_{H,\lambda_z,\lambda_w} \; & \int_{X_{\text{free}}} H(z)\, m(dz) \\ \text{subject to}: \; & H(z) \leq 0, \quad \forall z \in \bar{X}_{\text{goal}}, \\ & \langle \nabla H(z), f(z,w)\rangle + g(z,w) \\ & - \langle \lambda_z(z), h_z(z)\rangle - \langle \lambda_w(w), h_w(w)\rangle \in SOS, \\ & \lambda_z(z), \lambda_w(w) \in SOS. \end{aligned} \tag{17}$$

VI. Examples

The remainder of the letter is devoted to examples demonstrating how to apply Lemma 1 to verify admissibility, and the SOS approximation of (LP). For consistency, the GLC method [4] is used in numerical examples. The technique runs an approximate shortest path search on a tree of control primitives. With increasing search resolution the algorithm output converges to an optimal solution. However, the subtree searched by the algorithm may vary with different heuristics which explains why the solution in several of the following examples differ with differing heuristics.

A. Verifying Candidate Heuristics

The next three examples demonstrate some techniques utilizing Lemma 1 to verify the admissibility of a heuristic.

In the first example we show how to use Lemma 1 to verify a classic heuristic used in kinematic shortest path problems.

Example 1: Consider a holonomic shortest path problem,

$$\dot{x}(t) = u(t), \tag{18}$$

where $x(t) \in \mathbb{R}^n$, and $u(t) \in \{w \in \mathbb{R}^n : \|w\| = 1\}$. The cost is the path length,

$$\begin{aligned} J(x,u) = & \int_0^T \|\dot{x}(t)\|\, \mu(dt) \\ = & \int_0^T \|u(t)\|\, \mu(dt) \\ = & \int_0^T 1\, \mu(dt). \end{aligned} \tag{19}$$

Let the goal set be $\{0\}$. We would like to verify the classic heuristic

$$H(z) = \|z\|. \tag{20}$$

Applying the admissibility Lemma we obtain

$$\begin{aligned} \langle \nabla H(z), f(z,w)\rangle + g(z,w) = & \frac{\langle z, w\rangle}{\|z\|} + 1 \\ \geq & \frac{-\|z\|\|w\|}{\|z\|} + 1 \\ \geq & -1 + 1 \\ = & \; 0, \end{aligned} \tag{21}$$

which reverifies the fact that the Euclidean distance is an admissible heuristic for the shortest path problem. The crux of this derivation is simply applying the Cauchy-Schwarz inequality. Fig. 1 illustrates the use of this heuristic.

In the next example, we derive heuristics for a classic wheeled robot model.

Example 2: Consider a simple wheeled robot with state coordinates in $\mathbb{R}^3$ and whose mobility is described by

$$\begin{aligned} \dot{p}_1(t) = & \cos(\theta(t)), \\ \dot{p}_2(t) = & \sin(\theta(t)), \\ \dot{\theta}(t) = & u(t), \end{aligned} \tag{22}$$

where $(p_1(t), p_2(t), \theta(t))$ are individual coordinates of the state. Let $X_{\text{free}} = \mathbb{R}^3$, $X_{\text{goal}} = \{0\}$, and $\Omega = [-1, 1]$. The cost functional measures the duration of the trajectory,

$$J(p_1, p_2, \theta, u) = \int_0^T 1\, \mu(dt). \tag{23}$$

We will consider two candidate heuristics: the line segment connecting the p_1-p_2 coordinate to the origin (discussed in [16]), and the magnitude of the heading error,

$$\begin{aligned} H_1(p_1(t), p_2(t), \theta(t)) = & \sqrt{p_1(t)^2 + p_2(t)^2}, \\ H_2(p_1(t), p_2(t), \theta(t)) = & |\theta(t)|. \end{aligned} \tag{24}$$

Clearly, (AH1) is satisfied for both heuristics. Then (AH2) is once again verified with the Cauchy-Schwarz inequality.

For brevity, but with some abuse of notation, the time argument is dropped from the trajectory in the following derivations. Inserting the expression for the heuristics into (AH2) yields

$$\begin{aligned} &\langle \nabla H_1(p_1, p_2, \theta), f(p_1, p_2, \theta, u)\rangle + g(p_1, p_2, \theta, u) \\ &= \frac{\langle (p_1, p_2, 0), (\cos(\theta), \sin(\theta), u)\rangle}{\sqrt{p_1^2 + p_2^2}} + 1 \\ &\geq -\frac{\sqrt{p_1^2 + p_2^2}\sqrt{\cos^2(\theta) + \sin^2(\theta)}}{\sqrt{p_1^2 + p_2^2}} + 1 \\ &= 0. \end{aligned} \tag{25}$$

Similarly, for H_2,

$$\begin{aligned} &\langle \nabla H_2(p_1, p_2, \theta), f(p_1, p_2, \theta, u)\rangle + g(p_1, p_2, \theta, u) \\ &= \frac{\langle (0, 0, \theta), (\cos(\theta), \sin(\theta), u)\rangle}{|\theta|} + 1 \\ &\geq -\frac{|\theta||u|}{|\theta|} + 1 \\ &= 0. \end{aligned} \tag{26}$$

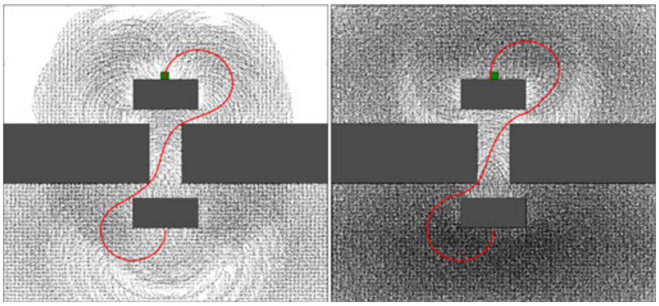

Fig. 2. Comparison of the vertices examined by the heuristically guided search (left) and the uniform cost search (right) for a simple wheeled robot motion planning problem. Equation 2) was used as the heuristic.

Therefore, both heuristics are admissible. Note that the maximum of the two heuristics is also an admissible heuristic,

$$H(p_1, p_2, \theta) = \max\{H_1(p_1, p_2, \theta), H_2(p_1, p_2, \theta)\}. \qquad (27)$$

Fig. 2 illustrates a shortest path query in a simple environment with and without the use of the heuristic in (2). The heuristically guided search obtains a solution in 81,686 iterations while the uniform cost search requires 403,197 iterations. In contrast, using just H_1 in (24) as a heuristic requires 104,492 iterations.

Example 3: Consider an autonomous underwater vehicle (AUV) navigating a strong current relative to the vehicle's top speed as discussed in [17]. This scenario is common for long range underwater gliders which travel at roughly 0.5 $^{\mathrm{m}}/_{\mathrm{s}}$ relative to currents traveling at 1.0–1.5 $^{\mathrm{m}}/_{\mathrm{s}}$.

Let the state space be $\mathbb{R}^n$ ($n = 2$ or 3) representing position in a local Cartesian coordinate system. The AUV's motion is modeled by

$$\dot{x}(t) = c(x(t)) + u(t), \qquad (28)$$

where $c : \mathbb{R}^n \to \mathbb{R}^n$ describes the current velocity at each point in the state space. The input $u(t)$ is the controlled velocity relative to the current. The thrust constraint is represented by

$$\Omega = \{w \in \mathbb{R}^n : \|w\| \leq w_{\max}\}, \qquad (29)$$

where $w_{\max}$ is the maximum achievable speed relative to the current. The goal set is $X_{\text{goal}} = \{0\}$. The objective

$$J(x, u) = \int_0^T 1 + \|u(t)\| \, \mu(dt), \qquad (30)$$

reflects a penalty on the duration of the path as well as the total work done by AUV's motors.

The admissibility of the heuristic proposed in [18] can be evaluated using Lemma 1. Let $v_{\max} := \max_z\{\|c(z)\|\}$ be the maximum speed of the current, and consider the heuristic

$$H(z) = \frac{\|z\|}{(v_{\max} + w_{\max})}. \qquad (31)$$

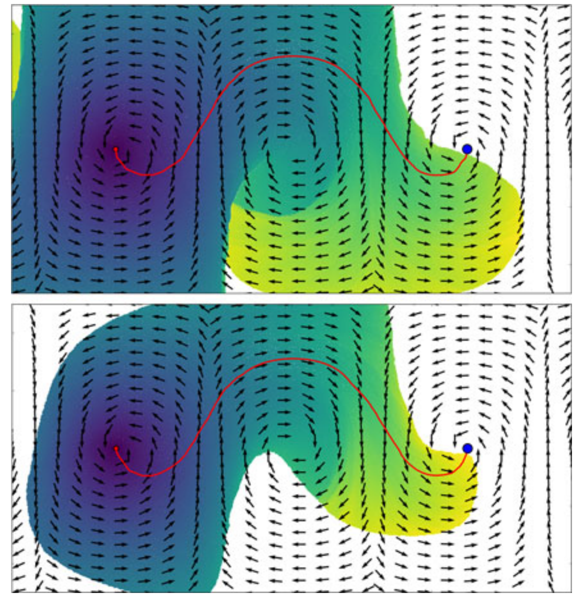

Fig. 3. Comparison of a minimum cost path search in a graph approximating feasible paths for an AUV in a strong current. The heuristically guided search using (31) of Example 3 obtains the solution in 271,642 iterations (left) while the uniform cost search requires 537,168 iterations (right). Initial and final states are illustrated with red and blue markers respectively. Colored regions represent states explored by the algorithm and the color indicates the relative cost to reach that state.

Clearly (AH1) is satisfied. Now evaluate (AH2),

$$\begin{aligned}\langle \nabla H(z), f(z,w)\rangle + g(z,w) &= \frac{\langle z, c(z) + w\rangle}{\|z\|(v_{\max} + w_{\max})} \\ &\quad + 1 + \|w\| \\ &\geq \frac{-\|z\| \cdot \|c(z)\| - \|z\| \cdot \|w\|}{\|z\|(v_{\max} + w_{\max})} \\ &\quad + 1 + \|w\| \\ &\geq \frac{-\|c(z)\| - \|w\|}{(v_{\max} + w_{\max})} \\ &\quad + 1 + \|w\|. \end{aligned} \qquad (32)$$

Notice that $v_{\max}$ and $w_{\max}$ were defined so that $\|w\| + \|c(z)\| \leq v_{\max} + w_{\max}$. Thus,

$$\begin{aligned}\langle \nabla H(z), f(z,w)\rangle + g(z,w) &\geq -1 + 1 + \|w\| \\ &\geq 0, \end{aligned} \qquad (33)$$

so the heuristic is admissible. Fig. 3 illustrates a numerical example in a 2D environment where the maximum current speed is 2.6 times the AUV's maximum relative speed. The heuristically guided search expands 49% fewer vertices than the uniform cost search.

B. SOS Heuristic Optimization Example

The last example demonstrates the SOS programming formulation described in Section V. A closed form solution for the optimal cost-to-go is known for $X_{\text{free}} = \mathbb{R}^2$ which provides a useful point of comparison for the computed heuristic. This examples also illustrates the flexibility in selecting a measure on X_{free}.

The example problem was implemented using the SOS module in YALMIP [10] and solved using SDPT3 for the underlying semi-definite program [14]. To further illustrate the approach, YALMIP scripts for this example can be found in [11].

Example 4 (Double Integrator): Consider a double integrator model,

$$\dot{x}_1(t) = x_2(t), \quad \dot{x}_2(t) = u(t), \tag{34}$$

with the minimum-time cost functional

$$J(x, u) = \int_0^T 1 \, dt, \tag{35}$$

and remaining problem data $X_{\text{free}} = [-3, 3]^2$, $\Omega = [-1, 1]$, and $X_{\text{goal}} = \{0\}$.

Polynomial heuristics of degree $2d$ of the form

$$H(x_1, x_2) = \sum_{p+q \le 2d} c_{p,q} \, x_1^p x_2^q, \tag{36}$$

are computed for $X_{\text{free}} = [-3, 3]^2$ and $\Omega = [-1, 1]$. In this example the support for the measure m is $S = [-2, 2] \times [-\sqrt{2}, \sqrt{2}]$. This focuses the optimization in a region around the goal while maintaining admissibility of the heuristic over all of X_{free}.

The SOS program is formulated as follows

$$\begin{aligned}
&\max_{H, \lambda_x, \lambda_u} \int_{X_{\text{free}}} H(x_1, x_2) \; m(dx) \\
&\text{subject to} : H(0, 0) \le 0, \\
&\langle \nabla H(x_1, x_2), (x_2, u) \rangle + 1 \\
&\quad - \lambda_{x_1}(x_1) \left(9 - x_1^2\right) \\
&\quad - \lambda_{x_2}(x_2) \left(9 - x_2^2\right) \\
&\quad - \lambda_u(u)(1 - u^2) \in SOS, \\
&\lambda_{x_1}(x_1), \lambda_{x_2}(x_2), \lambda_u(u) \in SOS.
\end{aligned} \tag{37}$$

Note that H can be integrated over the rectangular region S in closed form with standard integration rules or with the integration functionality in YALMIP.

The optimized heuristics of increasing degree are shown in Fig. 4 together with the value function for $X_{\text{free}} = \mathbb{R}^n$. The percent error in Fig. 4 is defined as

$$\%\,\text{error} = 100 \cdot \int_{X_{\text{free}}} \frac{V(z) - H(z)}{V(z)} m(dz). \tag{38}$$

C. Observations

The optimization of Example 4 focused on the region $[-2, 2] \times [-\sqrt{2}, \sqrt{2}]$ instead of $[-3, 3]^2$. The reason is that some states in $[-3, 3]^2$ cannot be reached and have infinite cost. A remarkable observation is that the resulting SOS program does not admit a maximum when the integral includes a subset of X_{free} where the value function is unbounded. This is entirely consistent with the theoretical results since the heuristic is free to go unbounded over this set as well. Therefore, it is important to optimize the heuristic over a region where the optimal cost-to-go is known to be bounded.

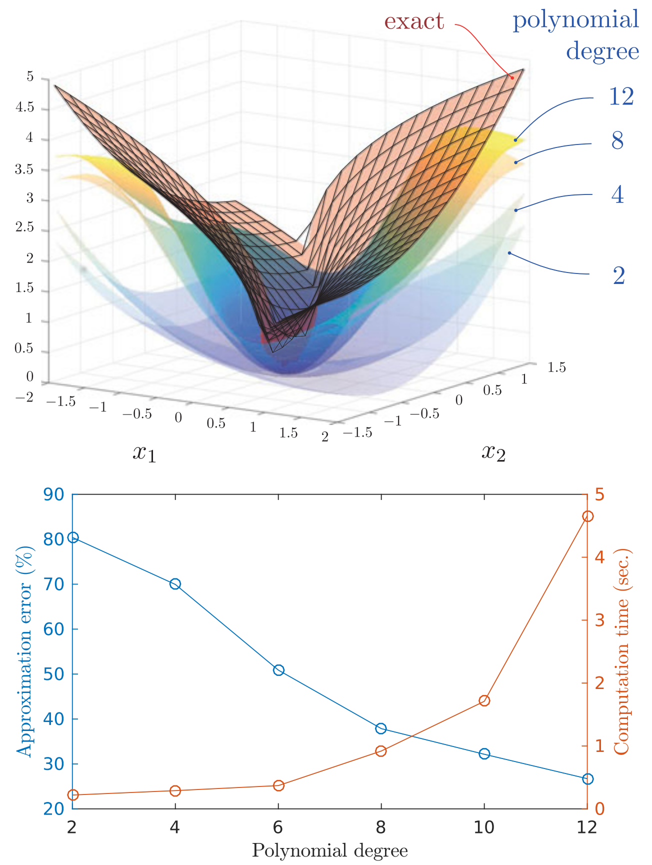

Fig. 4. (top) Polynomial heuristics of degree 2, 4, 8, and 12 for the double integrator model in comparison with the known value function shown in red. (bottom) Running time and approximation accuracy to compute heuristics of degrees from 2 to 12.

While semi-definite programs run in polynomial time in problem size, there is combinatorial growth in the number of monomials with increasing degree resulting in large semi-definite programs. For an n-dimensional state space, the number of monomials of degree d or less is

$$\binom{n+d}{n}. \tag{39}$$

This rapid growth in problem size is one of the current limitations of this approach to (LP). However, the technique produces a reasonable polynomial approximation in Example 4 despite a non-differentiable optimal value function.

There are a number possibilities for improving this technique. The DSOS and SDSOS [19] programming techniques which are being developed will enable the derivation of heuristics with higher degree polynomials. Additionally, a straight forward solution is to apply finite difference methods to directly approximate (LP) as a finite-dimensional linear program.

VII. Conclusions and Future Work

We have provided a sufficient condition for verifying the admissibility of a candidate heuristic in general kinodynamic motion planning problems and demonstrated the utility of this condition through several practical examples. The admissibility condition was then used to formulate a linear program over the space of candidate heuristics whose optimal solution coincides with classical solutions to the HJB equation. We also discussed a method for obtaining an approximate solution to this linear program using sum-of-squares programming techniques. This provides the first steps towards a general synthesis procedure for admissible heuristics to kinodynamic motion planning problems.

Automatic synthesis of admissible heuristics in kinodynamic motion planning will be a useful asset to many of the recently developed planning algorithms. Efforts to further develop this technique are being pursued. In the sequel, symmetry reduction techniques from optimal control theory will be applied to reduce the size of the resulting sum-of-squares program.

Appendix

Consistency of a heuristic is a type of triangle inequality. To define consistency, the value function and heuristic for the kinodynamic motion planning problem must be parametrized by the goal set. This is denoted $V(z; X_{\text{goal}})$ and $H(z; X_{\text{goal}})$. A heuristic $H(\,\cdot\,; X_{\text{goal}})$ is *consistent* if,

$$\begin{aligned} &lH(z; X_{\text{goal}}) = 0, && \forall z \in X_{\text{goal}}, \\ &H(z; X_{\text{goal}}) \leq V(z; \{y\}) + H(y; X_{\text{goal}}), && \forall y, z \in X_{\text{free}}. \end{aligned} \tag{40}$$

Note that the inequality above involves the optimal cost-to-go from z to y.

Lemma 3 (Consistency): A heuristic $H(\,\cdot\,; X_{\text{goal}})$ is consistent if:

$$H(z; X_{\text{goal}}) = 0, \qquad \forall z \in X_{\text{goal}}, \tag{CH1}$$

and

$$\langle \nabla H(z; X_{\text{goal}}), f(z,w) \rangle + g(z,w) \geq 0, \tag{CH2}$$

for all $u \in \Omega$ and all $z \in X_{\text{free}}$.

The proof is nearly identical to that of Lemma 1.

Proof: Choose a trajectory x and associated control signal u such that $x(0) = z$ and $x(T) = y$. Then

$$\begin{aligned} &H(x(0); X_{\text{goal}}) - H(x(T); X_{\text{goal}}) \\ &= -\int_0^T \frac{d}{dt} H(x(t); X_{\text{goal}})\, \mu(dt) \\ &= -\int_0^T \langle \nabla H(x(t); X_{\text{goal}}), f(x(t), u(t)) \rangle \, \mu(dt) \\ &\leq \int_0^T g(x(t), u(t)\, \mu(dt) \\ &= J(x,u). \end{aligned} \tag{41}$$

Thus, $H(z; X_{\text{goal}}) - H(y; X_{\text{goal}})$ lower bounds $J(x,u)$ for any trajectory starting at z and terminating at y. Since $V(\,\cdot\,; y)$ is the greatest lower bound to the cost of such trajectories we have

$$H(z; X_{\text{goal}}) - H(y; X_{\text{goal}}) \leq V(z; y), \qquad \forall y, z \in X_{\text{free}}. \tag{42}$$

Rearranging the expression above yields the definition of consistency for $H(\,\cdot\,; X_{\text{goal}})$. ■

References

[1] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[2] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *Proc. IEEE Conf. Decis. Control*, 2010, pp. 7681–7687.

[3] K. Hauser and Y. Zhou, "Asymptotically optimal planning by feasible kinodynamic planning in state-cost space," arXiv:1505.04098, 2015. [Online]. Available at: http://arxiv.org/abs/1505.04098

[4] B. Paden and E. Frazzoli, "A generalized label correcting method for optimal kinodynamic motion planning," in *Proc. Algorithmic Found. Robot. XII (WAFR)*, 2016. [Online]. Available at: http://www.wafr.org/program.html\#detailedprogram

[5] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal incremental path planning focused through an admissible ellipsoidal heuristic," in *Proc. Int. Conf. Intell. Robots Syst.*, 2014, pp. 3067–3074.

[6] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 3067–3074.

[7] W. H. Fleming and D. Vermes, "Generalized solutions in the optimal control of diffusions," in *Proc. Stochastic Differ. Syst. Stochastic Control Theory Appl.*, Springer, 1988, pp. 119–127.

[8] R. Vinter, "Convex duality and nonlinear optimal control," *J. Control Optim.*, vol. 31, no. 2, pp. 518–538, 1993.

[9] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," PhD dissertation, California Inst. Tech., 2000.

[10] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Symp. Comput. Aided Control Syst. Des.*, 2004, pp. 284–289.

[11] B. Paden, V. Varricchio, and E. Frazzoli, "Sum-of-squares heuristic synthesis for kinodynamic motion planning." 2016. [Online]. Available at: https://github.com/bapaden/Sum_of_Squares_Admissible_Heuristics/releases

[12] M. G. Crandall and P.-L. Lions, "Viscosity solutions of hamilton-jacobi equations," *Trans. Amer. Math. Soc.*, vol. 277, no. 1, pp. 1–42, 1983.

[13] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.*, vol. 11, no. 1–4, pp. 625–653, 1999.

[14] K.-C. Toh, M. J. Todd, and R. H. Tutuncu, "SDPT3-a MATLAB software package for semidefinite programming, version 1.3," *Optim. Methods Softw.*, vol. 11, no. 1–4, pp. 545–581, 1999.

[15] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, "Introducing SOSTOOLS: A general purpose sum-of-squares programming solver," in *Proc. IEEE Conf. Decis. Control*, vol. 1, 2002, pp. 741–746.

[16] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 485–501, 2010.

[17] B. Garau, M. Bonet, A. Alvarez, S. Ruiz, and A. Pascual, "Path planning for autonomous underwater vehicles in realistic oceanic current fields: Application to gliders in the western mediterranean sea," *J. Maritime Res.*, vol. 6, no. 2, pp. 5–22, 2014.

[18] B. Garau, A. Alvarez, and G. Oliver, "Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach," in *Proc. Int. Conf. Robot. Autom.*, 2005, pp. 194–198.

[19] A. A. Ahmadi and A. Majumdar, "DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum-of-squares optimization," in *Proc. IEEE Conf. Inf. Sci. Syst.*, 2014, pp. 1–5.