Design of Admissible Heuristics for Kinodynamic Motion Planning via Sum of Squares Programming

Brian Paden, Valerio Varricchio, and Emilio Frazzoli

***Abstract*—How does one obtain an admissible heuristic for a kinodynamic motion planning problem? This paper 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 the problem data. This condition is also used to formulate an infinite dimensional linear program to optimize an admissible heuristic to be a close underestimate of the optimal cost to reach the goal. This optimization is then approximated and solved in polynomial time using sum of squares programming techniques. A number of examples are provided to demonstrate these concepts.**

***Index Terms*—Kinodynamic motion planning, Heuristic search, Sum-of-squares programming, Convex optimization.**

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 paper 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. Figure I demonstrates the benefit of using this heuristic on a typical shortest path problem where informing the search reduces the number iterations required to find a solution by 67%.

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 Section II and III respectively.

A good heuristic is one which closely underestimates the optimal cost to go from every vertex. This enables a larger number of provably suboptimal paths to be identified and discarded from the search. This gives rise to two challenges that are addressed in this paper. First, without a priori knowledge of the optimal cost to reach the goal, the definition of admissibility cannot be directly verified. This is particularly true for kinodynamic motion planning problems that can have complicated discontinuous value functions characterizing the cost to reach the goal. Second, it may be difficult to come up with a candidate heuristic in the first place.

The first contribution of this paper, presented in Section III, is a sufficient condition for the admissibility of a candidate heuristic. This condition takes the form of a linear partial differential inequality involving the heuristic and given problem data. The result provides a general analytical tool for validating a heuristic generated by intuition about the problem.

In Section IV, the admissibility condition is used to formulate a linear program over a Banach space of candidate heuristics. The objective of the linear program is constructed so that a solution to the Hamilton-Jacobi-Bellman (HJB) equation is a globally optimal solution to the optimization.

Section V outlines a computational procedure for approaching the optimization. An finite dimensional subspace of polynomials is used to approximates the space of heuristics. Sum of squares (SOS) programming [7] techniques are then used to obtain an approximate solution in polynomial time. In doing so we provide the first general procedure to compute admissible heuristics to kinodynamic motion planning problems.

Examples demonstrating how to use the admissibility condition to verify that a heuristic is admissible as well as numerical examples of the SOS programming approach are provided in Section VI. The YALMIP [8] scripts used to compute the example heuristics can be found in [].

II. Kinodynamic Motion Planning

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

In a kinodynamic motion planning problem, trajectories must satisfy several point-wise constraints. The first is an initial state constraint, $x(0) = x_0$ for a specified state $x_0 \in X_{free}$. Secondly, there is a terminal constraint; $x(T) \in X_{goal}$ for a specified subset $X_{goal} \subset \mathbb{R}^n$. Lastly, a subset $X_{free} \subset \mathbb{R}^n$ of the state space encodes the set of allowable states over the entire domain of the trajectory; $x(t) \in X_{free}$ for all $t \in [0,T]$. A trajectory x is *kinematically feasible* if it satisfies these point-wise constraints.

In addition to the kinematic constraints, the trajectory must satisfy differential constraints. At each time t the system is

The authors are with the Laboratory for Information and Decision Systems at MIT (e-mail: bapaden@mit.edu, varricchio@mit.edu, frazzoli@mit.edu).

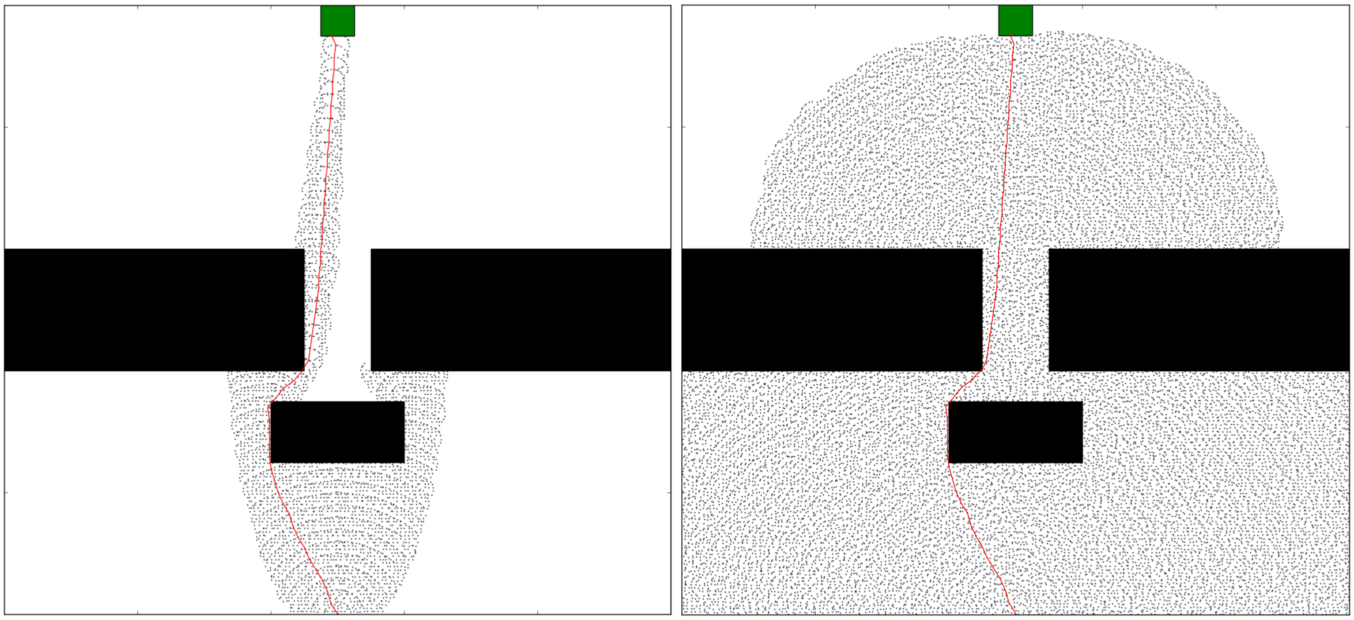

Fig. 1. A classic example where an admissible heuristic speeds up a search. Approximate shortest kinematic paths in a 2D environment computed with the generalized label correcting (GLC) method [4] are shown with dots representing vertices evaluated during the search. The algorithm was executed with (left) and without (right) an admissible heuristic. While the underlying graph is identical, the informed GLC method obtains a solution in 5203 iterations while the standard GLC method obtains a solution in 19030 iterations.

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* and unlike a trajectory it need not be continuous. However, 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 $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is globally Lipschitz. A trajectory x with domain $[0, T]$ is *dynamically feasible* if there exists a control signal u satisfying (1) for almost all $t \in [0, T]$. A *feasible* trajectory is one that is both dynamically and kinematically feasible.

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

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

It is assumed $g(z, w) \geq 0$ so that a nonnegative infinitesimal cost is associated to each state-action pair. A solution to an optimal kinodynamic motion planning problem is a feasible trajectory and control signal which minimizes (2).

A. The Value Function

The cost-to-go or value function $V : X_{free} \to \mathbb{R}$ describes is the greatest lower bound on the cost to reach the goal set from the initial state $z \in X_{free}$. The following properties of V follow immediately from the assumption $g(z, w) \geq 0$ in (2).

$$\begin{aligned} V(z) &\geq 0, \qquad \forall z \in X_{free}, \\ V(z) &= 0, \qquad \forall z \in X_{goal}. \end{aligned} \tag{3}$$

Additionally, if V is a classical[1] solution to the Hamilton-Jacobi-Bellman (HJB) equation:

$$\min_{w\in\Omega} \{\langle \nabla V(z), f(z, w)\rangle + g(z, w)\} = 0,\ \forall z \in X_{free} \backslash X_{goal}, \tag{4}$$

and $V(z) = 0$ for all z in the closure of X_{goal} (denoted $\bar{X}_{goal}$), then V is equal to the value function on X_{free}.

III. Graph-Search Oriented Approximations

Many methods for computing an approximate solution to the problem approximate the problem by a finite directed graph $(\mathcal{V}, E)$ whose vertices are states in the state space, and whose edges correspond to trajectories satisfying (1) connecting two vertices. Conceptually, the optimal feasible trajectories restricted to the graph are in some sense faithful approximations of optimal feasible trajectories for the original problem.

The non-negativity of the cost function (2) enables a non-negative 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 x_0 in the graph, $\hat{V}(x_0)$ is the cost of a shortest path to one of the goal vertices: $\mathcal{V} \cap X_{goal}$. Since the feasible trajectories represented by the graph are a subset of the feasible trajectories of the problem we have the inequality

$$V(z) \leq \hat{V}(z), \qquad \forall z \in \mathcal{V}. \tag{5}$$

[1]The gradient of V is well defined and the equation is satisfied for all $x \in X_{free} \backslash \bar{X}_{goal}$. The extension to viscosity solutions is beyond the scope of this paper.

A. Admissible Heuristics

An informed search making use of a heuristic can dramatically speed up the computation time of the approximated problem. To carry out an informed search and ensure the optimality of the result, many algorithms require an admissible heuristic $H : X_{free} \to \mathbb{R}$. A heuristic H for a problem with value function V is *admissible* if,

$$H(z) \leq V(z), \qquad \forall z \in X_{free}. \tag{6}$$

A useful property of the set of admissible heuristics is that they are closed under the $\max$ operation. More generally, if $\{H_k\}_{k\in\mathcal{K}}$ is a collection of admissible heuristics then the heuristic

$$H^\sharp(x) = \sup_{k\in\mathcal{K}}\{H_k(x)\} \tag{7}$$

is also admissible. This is an immediate consequence of the definition. Note also that the heuristic $H(x) = 0$ is admissible as a consequence of (3).

For the remainder, the set of candidate heuristics will be restricted to the Sobelov space $W^{1,1}(X_{free})$ of weakly differentiable functions on X_{free} into $\mathbb{R}$. This set is a Banach space with the norm

$$\|H\|_{W^{1,1}} = \int_{X_{free}} |H(z)| + \|\nabla H(z)\|\, d\mu(z). \tag{8}$$

The notation ∇H denotes the weak derivative of H which is the usual gradient when H is differentiable on X_{free}.

Since the value function is unknown it is difficult check that (6) is satisfied for a particular heuristic H. This motivates the first contribution of this paper; a sufficient condition for admissibility that can be checked using the problem data.

Lemma 1 (Admissibility)**.** *A heuristic $H \in W^{1,1}(X_{free})$ is an admissible heuristic if:*

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

and

$$\langle \nabla_z H(z), f(z,w)\rangle + g(z,w) \geq 0, \tag{AH2}$$

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

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

$$\begin{aligned} H(x(0)) &\leq H(x(0)) - H(x(T)) \\ &= -\textstyle\int_{[0,T]} \frac{d}{dt} H(x(t))\, d\mu(t) \\ &= -\textstyle\int_{[0,T]} \langle \nabla H(x(t)), f(x(t), u(t))\rangle\, d\mu(t) \\ &\leq \textstyle\int_{[0,T]} g(x(t), u(t)\, d\mu(t) \\ &= J(x,u). \end{aligned} \tag{9}$$

Note that the fourth step of the derivation follows from assuming (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,

$$H(z) \leq V(z), \qquad \forall z \in X_{free}. \tag{10}$$

□

While a heuristic is admissible if it satisfies (AH1) and (AH2), the converse is not necessarily true. There may be admissible heuristics not satisfying these conditions. Conveniently, heuristics satisfying the admissibility conditions form a convex set containing solutions to the HJB equation.

Observe that Lemma 1 does not require V or g to be continuous, nor does f have to be differentiable, making the it quite general. An immediate application of this result is to verify the admissibility of a candidate heuristic by checking the conditions of Lemma 1. Section VI provides three examples demonstrating this technique.

IV. Optimization of Admissible Heuristics

The second contribution of this paper is a general procedure for computing and optimizing an admissible heuristic.

Consider the following (infinite dimensional) linear program

$$\begin{gathered} \max_{H\in W^{1,1}(X_{free})} F(H) := \textstyle\int_{X_{free}} H(z)\, d\mu(z) \\ \text{subject to} : (AH1), \text{ and } (AH2) \end{gathered} \tag{LP}$$

where μ is a finite measure on X_{free}.

The constraint enforces admissibility of a feasible solution, while the objective favors heuristics with larger values. Since this is a linear program it is a concave maximization over a convex set; any locally optimal solution is also globally optimal.

The next result justifies the formulation of (LP) beyond the qualitative construction given above. In particular, a classical solution to the HJB equation is a feasible solution to the optimization.

Lemma 2. *If the value function V is a classical solution to the HJB equation then it is the unique (modulo a set of measure zero), globally optimal solution to (LP).*

Proof. (Feasibility) By definition, $V(z) = 0$ for all $z \in \bar{X}_{goal}$. Thus, the constraint (AH1) is satisfied and (AH2) is satisfied on $\bar{X}_{goal}$. Next,

$$\begin{gathered} \min_{w\in\Omega} \{\langle \nabla V(z), f(z,w)\rangle + g(z,w)\} = 0, \\ \forall z \in X_{free} \setminus \bar{X}_{goal}. \end{gathered} \tag{11}$$

This implies that

$$\begin{gathered} \langle \nabla V(z), f(z,w)\rangle + g(z,w) \geq 0, \\ \forall z \in X_{free} \setminus \bar{X}_{goal}, \text{ and } u \in \Omega. \end{gathered} \tag{12}$$

Therefore, solutions to the HJB equation satisfy (AH2) on all of X_{free}.

(Global optimality) By Lemma 1, any feasible solution H to the optimization is an admissible heuristic satisfying $H(x) \leq V(x)$ for all $x \in X_{free}$. Then,

$$\int_{X_{free}} H(x)\, d\mu(x) \leq \int_{X_{free}} V(x)\, d\mu(x). \tag{13}$$

That is, the integral of the value function upper bounds the optimal value to the optimization. However, this bound is attained since V is a feasible solution to the optimization. Thus, V is a global maximizer.

(Uniqueness) Suppose H' is a feasible solution attaining the optimal value, but there exists a subset of X_{free} with positive measure such that $H'(z) \neq V(z)$. By Lemma 1, $H'(z) \leq V(z)$ for all $z \in X_{free}$, but $H'(z) < V(z)$ on a set with positive measure. Therefore

$$\int_{X_{free}} H'(x)\, d\mu(x) < \int_{X_{free}} V(x)\, d\mu(x), \tag{14}$$

which is a contradiction. Thus, V is the unique globally optimal solution modulo solutions differing on a set of measure zero.

□

V. Sum of Squares (SOS) Relaxation to (LP)

To tackle (LP) with standard mathematical programming techniques, we must approximate $W^{1,1}(X_{free})$ by a finite dimensional subspace. The proposed basis for this subspace is a finite collection of polynomials. The relaxation can then be addressed efficiently using SOS programming.

SOS programming [7] is a method of optimizing a functional of a polynomial subject to semi-algebraic constraints. The technique involves relaxing the semi-algebraic constraints to a sum of squares constraint which results is equivalent to a semi-definite program (SDP). The advantages of this approach are that the approximate solution is guaranteed to be an admissible heuristic, and the relaxation is a convex program which can be solved in polynomial time using interior point methods.

A. Sum of Squares Polynomials

A polynomial $p \in \mathbb{R}[x]$ of degree $2d$ in n variables is said to be a sum of squares if it can be written as

$$p(x) = \sum_{k=1}^{d} q_k(x)^2, \tag{15}$$

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

$$p(x) = m(x)^T Q m(x), \tag{16}$$

for a positive semidefinite matrix Q and the vector of $\binom{n+d}{n}$ monomials $m(x)$ up to degree d. For a polynomial p admitting a decomposition of the form (16) we write $p \in SOS$.

Equation (16) is a collection of linear equality constraints between the entries of Q and the coefficients of $p(x)$. Finding entries of Q such that $Q \succeq 0$ and the equality constraints are satisfied is then a SDP. The complexity of finding a solution to this problem using interior point methods is generally 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 optimized solvers [9], [10] and modeling tools [11], [8] available.

B. Optimization of a Heuristic

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

$$\begin{aligned} X_{free} &= \{x \in \mathbb{R}^n : \, h_x(x) \geq 0\}, \\ \Omega &= \{u \in \mathbb{R}^m : \, h_u(u) \geq 0\}, \end{aligned} \tag{17}$$

for polynomials h_x and h_u. Assume also that f, g and the the candidate heuristic H are polynomials. Then the admissibility condition

$$\langle \nabla_x H(x), f(x,u)\rangle + g(x,u) \geq 0, \tag{18}$$

is a polynomial inequality. To restrict our attention to X_{free} and Ω, add the auxiliary sum of squares polynomials $\lambda_x(x) \geq 0$ and $\lambda_u(u) \geq 0$ to the equation as

$$\begin{aligned} &\langle \nabla_x H(x), f(x,u)\rangle + g(x,u) \\ &-\lambda_x(x)^T h_x(x) - \lambda_u(u)^T h_u(u) \geq 0 \end{aligned} \tag{19}$$

which trivially implies the positivity of (18) over X_{free} and Ω.

Note that the objective in (LP) is also linear in the coefficients of H when H is a polynomial. Thus, it is an appropriate objective for an SOS program. Note that the problem remains an SOS program if the measure μ has support on a proper subset of X_{free}, or if a discrete measure is used.

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

$$\begin{aligned} &\max_{H,\lambda_x,\lambda_u} \{F(H)\} \\ &\text{subject to} \\ &H(x) = 0 \;\forall\; x \in \bar{X}_{goal}, \\ &\langle \nabla_x H(x), f(x,u)\rangle + g(x,u) \\ &-\lambda_x(x) h_x(x) - \lambda_u(u) h_u(u) \in SOS, \\ &\lambda_x(x), \lambda_u(u) \in SOS. \end{aligned} \tag{20}$$

VI. Examples

The remainder of the paper is devoted to examples demonstrating the utility of Lemma 1 and the SOS relaxation of (LP).

A. Verifying Candidate Heuristics

It is often the case that intuition about a problem leads to a candidate heuristic. However, it can be difficult to verify the admissibility of that heuristic.

The next three examples demonstrate some techniques utilizing Lemma 1 to verify the admissibility of a heuristic. In particular, the Cauchy-Schwarz inequality and the inequality

$$|2ab| \leq a^2 + b^2, \tag{21}$$

are often useful.

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 reformulation of the shortest path problem,*

$$\dot{x} = u, \tag{22}$$

where $x \in \mathbb{R}^n$, and $u \in \{w \in \mathbb{R}^n : \|w\| = 1\}$. The cost which reflects a shortest path objective is

$$J(x,u) = \int_0^T 1 \, d\mu(t). \tag{23}$$

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

$$H(x) = \|x\|. \tag{24}$$

Applying the admissibility Lemma we obtain

$$\begin{aligned} \langle \nabla H(x), f(x,u)\rangle + g(x,u) &= \frac{\langle x, u\rangle}{\|x\|} + 1 \\ &\geq \frac{-\|x\|\|u\|}{\|x\|} + 1 \\ &\geq -1 + 1 \\ &= 0, \end{aligned} \tag{25}$$

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 in the first step.

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

Example 2. *Consider a simple wheeled robot with states $(x, y, \theta)^T \in \mathbb{R}^3$ and whose mobility is described by*

$$\begin{aligned} \dot{x} &= \cos(\theta), \\ \dot{y} &= \sin(\theta), \\ \dot{\theta} &= u. \end{aligned} \tag{26}$$

Let $X_{free} = \mathbb{R}^3$, $X_{goal} = \{(0,0,0)^T\}$, and $\Omega = \mathbb{R}$. The cost functional measures the duration of the trajectory.

$$J(x,u) = \int_{[0,T]} 1 \, d\mu(t). \tag{27}$$

Equivalently, this is the path length in the x-y plane.

As a candidate heuristic, take the length of the line segment connecting the x-y coordinate to the origin.

$$H_1(x,y,\theta) = \|(x,y)^T\|. \tag{28}$$

The intuition being that the shortest path in the absence of the differential constraint will be shorter than the shortest path for the constrained system.

The admissibility condition is verified for this heuristic using the Cauchy-Schwarz inequality. Inserting the expression for the heuristic into (AH2) yields

$$\begin{aligned} &\langle \nabla H(x,y,\theta), f(x,y,\theta)\rangle + g(x,y,\theta,u) \\ &= \frac{\left\langle (x,y,0)^T, (\cos(\theta), \sin(\theta), u)^T \right\rangle}{\|(x,y)^T\|_2} + 1 \\ &\geq -\frac{\left\|(x,y)^T\right\|_2 \left\|(\cos(\theta), \sin(\theta))^T\right\|}{\|(x,y)^T\|_2} + 1 \\ &\geq -1 + 1 \\ &= 0 \end{aligned} \tag{29}$$

Thus, the heuristic is admissible.

Next, consider a restriction of the control actions to $\Omega = [-1, 1]$. The original heuristic remains valid since the old problem is a relaxation of the new problem. Additionally, we can consider a second heuristic to augment the first,

$$H_2(x) = |\theta|, \tag{30}$$

With the added constraint, this heuristic satisfies Lemma 1,

$$\begin{aligned} \langle \nabla H_2(x), f(x,u)\rangle + g(x,u) &= \frac{\theta u}{|\theta|} + 1 \\ &\geq -\frac{|\theta||u|}{|\theta|} + 1 \\ &\geq -1 + 1 \\ &= 0, \end{aligned} \tag{31}$$

We can then combine these heuristics in the input constrained problem with the input constraint

$$H(x,y,\theta) = \max\{\|(x,y)\|, |\theta|\} \tag{32}$$

The heuristic in (32) was used to plan a feasible path in a 2D environment illustrated in Figure VI-A.

For the demonstration the singleton goal set was approximated by a small cube centered at $(0,0,0)^T$ as required by the motion planning algorithm. The use of the heuristic reduces the number of iterations of the algorithm by 91%.

The last example considers a problem with a quadratic regulator objective instead of a minimum time objective.

Example 3. *Consider a simple pendulum with dynamics*

$$\begin{aligned} \dot{\theta} &= \omega \\ \dot{\omega} &= \sin(\theta) + u \end{aligned} \tag{33}$$

Let $X_{free} = \mathbb{R}^2$, $\Omega = [-1, 1]$, and $X_{goal} = \{(0,0)^T\}$. The cost function will be a typical quadratic regulator cost.

$$J = \int_0^T \rho(\theta^2 + \omega^2 + u^2) \, d\mu(t). \tag{34}$$

Select a heuristic of the form

$$H(\theta, \omega) = \frac{\alpha}{2}\left(\theta^2 + \omega^2\right). \tag{35}$$

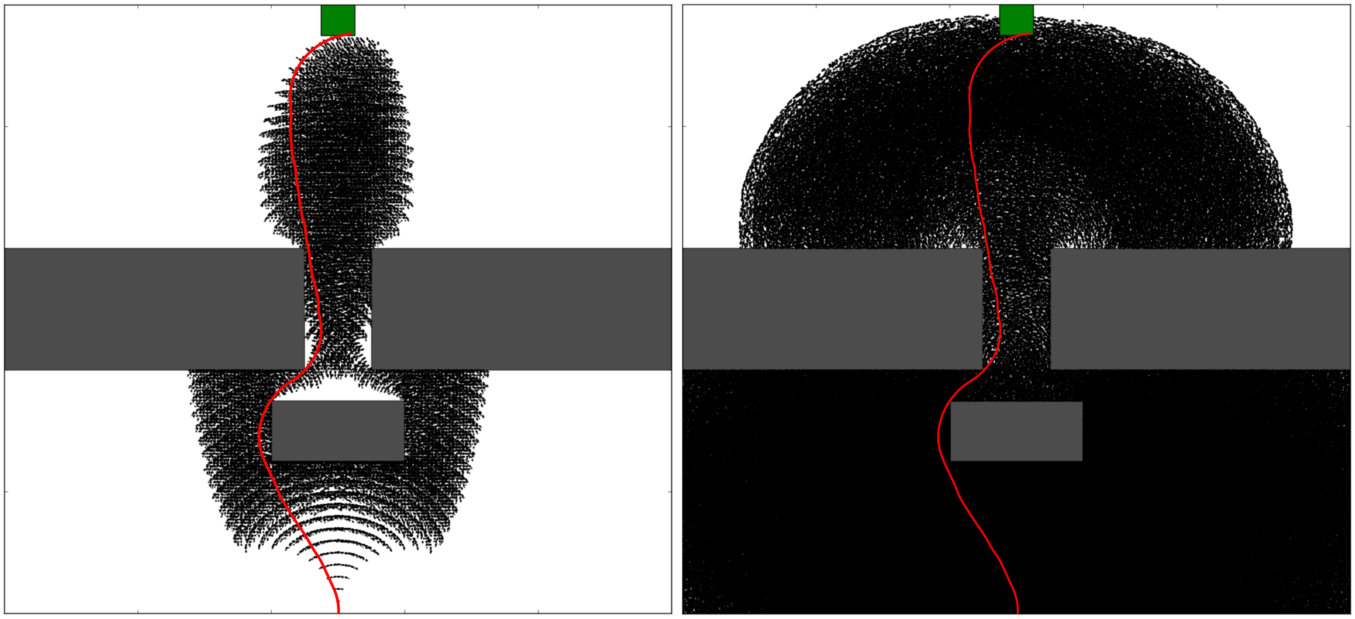

Fig. 2. Approximate shortest path in a 2D environment for a simple wheeled robot. Paths were computed by the GLC method with dots representing the projection of vertices evaluated during the search onto the x-y plane. The algorithm was executed with (left) and without (right) the admissible heuristic described in equation (32) of Example 2. The informed GLC method obtains the illustrated solution in 209341 iterations while the standard GLC method obtains the solution in 2380952 iterations.

Checking the admissibility condition,

$$\begin{aligned}&\langle \nabla H(x), f(x,u)\rangle + g(x,u)\\ &= \alpha\theta\omega + \alpha\omega\sin(\theta) + \alpha\omega u + \rho(\theta^2+\omega^2+u^2)\\ &\geq -|2\alpha\theta\omega| - |\alpha\omega u| + \rho(\theta^2+\omega^2+u^2)\\ &\geq -\alpha(\theta^2+\omega^2) - \tfrac{1}{2}\alpha(\omega^2+u^2) + \rho(\theta^2+\omega^2+u^2)\\ &= (\rho - \tfrac{3}{2}\alpha)\theta^2 + (\rho-\tfrac{3}{2}\alpha)\omega^2 + (\rho - \tfrac{1}{2}\alpha)u^2.\end{aligned} \tag{36}$$

The inequality in (21) was used in the third step of this derivation. The above quantity is nonnegative and therefore H is an admissible heuristic for $\alpha \leq \frac{2}{3}\rho$.

B. SOS Heuristic Optimization Examples

The next two examples demonstrate the SOS programming formulation described in Section V. In both examples, a closed form solution for the value function $V(x)$ is known for $X_{free} = \mathbb{R}$ and $X_{free} = \mathbb{R}^2$ respectively. This solution provides a useful point of comparison for the computed heuristics. These examples also illustrate how the flexibility in selecting a measure on X_{free}.

These example problems were implemented using the SOS module in YALMIP [8] and solved using SDPT3 for the underlying semidefinite program [10]. To further illustrate the approach, YALMIP scripts for these examples can be found in [].

Example 4 (Single Integrator (1D))**.** *To illustrate the procedure, take the trivial example $f(x,u) = u$ where $x, u \in \mathbb{R}$, $X_{free} = [-1,1]$, $\Omega = [-1,1]$, and $X_{goal} = \{0\}$. Again we use the minimum time objective*

$$J(x,u) = \int_0^T 1\, d\mu(t). \tag{37}$$

The value function $V(x) = |x|$ is obtained by inspection.

The heuristic is parameterized by the coefficients of a univariate polynomial of degree $2d$

$$H(x) = \sum_{i=0}^{2d} c_i x^i. \tag{38}$$

Using a discrete measure on $[-1,1]$ concentrated at the boundary the SOS program is,

$$\begin{aligned}&\max_{H,\lambda_x,\lambda_u} \{H(1) + H(-1)\}\\ &\text{subject to,}\\ &H(0) = 0,\\ &\left(\tfrac{d}{dx}H(x)\right)u + 1\\ &-\lambda_x(x)(1-x^2) - \lambda_u(u)(1-u^2) \in SOS,\\ &\lambda_x(x), \lambda_u(u) \in SOS.\end{aligned} \tag{39}$$

The numerical solution for polynomial heuristics with increasing degree is shown in Figure 3.

Example 5 (Double Integrator (1D))**.** *As an example with a more complex value function take the vector field*

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} = \begin{pmatrix}x_2\\ u\end{pmatrix}, \tag{40}$$

and the minimum-time cost functional

$$J(x,u) = \int_0^T 1\, d\mu(t). \tag{41}$$

The remaining problem data for this example are $X_{free} = [-3,3]^2$, $\Omega = [-1,1]$, and $X_{goal} = \{(0,0)^T\}$. The exact

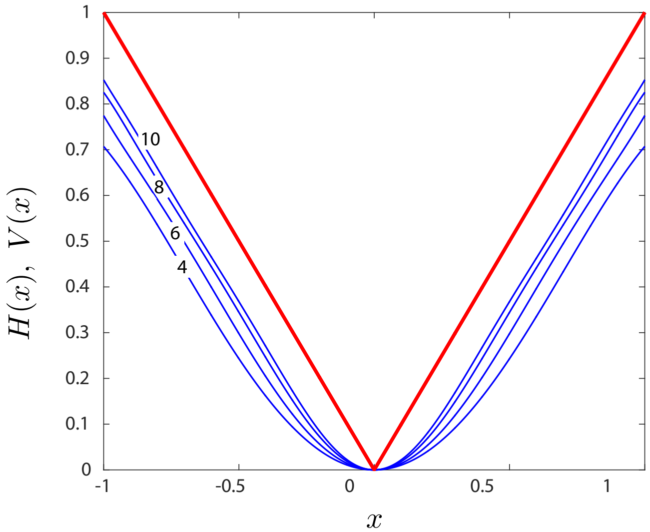

Fig. 3. Univariate polynomial heuristics of degrees 4, 6, 8, and 10 for the 1D single integrator shown in blue. The value function is shown in red. Polynomial heuristics with higher degree provide better underestimates of the value function.

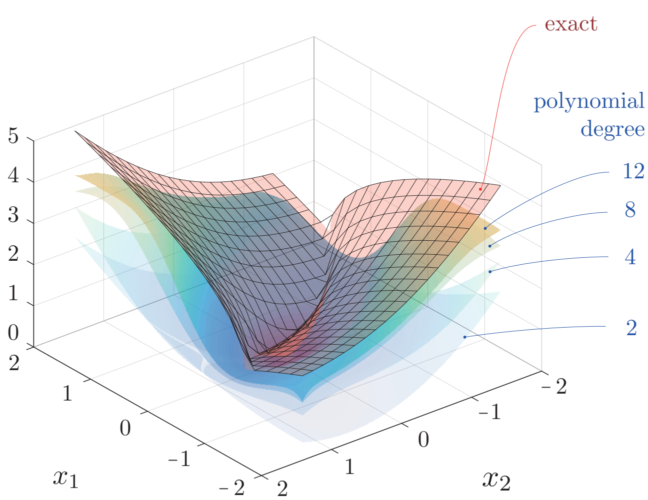

Fig. 4. Polynomial heuristics of degree 2, 4, 8, and 12 for the 1D double integrator in comparison with the known value function shown in red. Polynomial heuristics with higher degree provide better underestimates of the value function.

value function for this problem can be computed by hand and provides a point of comparison for the computed heuristics.

Polynomial heuristics of degree $2d$ *of the form*

$$H(x_1, x_2) = \sum_{p+q \leq 2d} c_{p,q}\; x_1^p x_2^q, \tag{42}$$

are computed for $X_{free} = [-3, 3]^2$ *and* $\Omega = [-1, 1]$. *In this example the support for the measure* μ *is* $[-2, 2]^2$.

The SOS program is formulated as follows

$$\begin{aligned} \max_{H,\lambda_x,\lambda_u} \int_{[-2,2]^2} H(x_1, x_2)\, d\mu(x) & \\ \text{subject to,} & \\ H(0,0) = 0, & \\ (\nabla H(x_1, x_2)) \begin{pmatrix} x_2 \\ u \end{pmatrix} + 1 & \\ -\lambda_{x_1}(x_1)\left(9 - x_1^2\right) & \\ -\lambda_{x_2}(x_2)\left(9 - x_2^2\right) & \\ -\lambda_u(u)(1-u^2) \in SOS & \\ \lambda_{x_1}(x_1), \lambda_{x_2}(x_2), \lambda_u(u) \in SOS. & \end{aligned} \tag{43}$$

The optimized heuristics of increasing degree are shown in Figure VI-B together with the value function for $X_{free} = \mathbb{R}^n$.

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 through several examples how to utilize the condition. The admissibility condition was then used to formulate a linear program over a Banach space whose optimal solution coincides with classical solutions to the HJB equation. Using sum of squares programming we were able to provide approximate solutions to this optimization in polynomial time. This provides the first 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. We will also explore using the DSOS and SDSOS [12] programming techniques which would enable using polynomial heuristics with higher degree.

References

[1] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, 1968.

[2] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *49th IEEE conference on decision and control (CDC)*, pp. 7681–7687, IEEE, 2010.

[3] K. Hauser and Y. Zhou, "Asymptotically optimal planning by feasible kinodynamic planning in state-cost space," *arXiv preprint arXiv:1505.04098*, 2015.

[4] B. Paden and E. Frazzoli, "A generalized label correcting method for optimal kinodynamic motion planning," *arXiv preprint arXiv:1607.06966*, 2016.

[5] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal incremental path planning focused through an admissible ellipsoidal heuristic," in *International Conference on Intelligent Robots and Systems*, 2014.

[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 *International Conference on Robotics and Automation*, pp. 3067–3074, IEEE, 2015.

[7] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, Citeseer, 2000.

[8] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pp. 284–289, IEEE, 2004.

[9] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

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

[11] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, "Introducing SOSTOOLS: A general purpose sum of squares programming solver," in *41st Conference on Decision and Control*, vol. 1, pp. 741–746, IEEE, 2002.

[12] A. A. Ahmadi and A. Majumdar, "DSOS and SDSOS optimization: Lp and socp-based alternatives to sum of squares optimization," in *48th Annual Conference on Information Sciences and Systems*, pp. 1–5, IEEE, 2014.