Efficient Guidance in Finite Time Flow Fields

Blane Rhoads, Igor Mezić, and Andrew Poje

Abstract—We study path planning for small vehicles in strong, spatially complex, time-varying flow fields. Of particular interest is how optimal trajectories relate to flow structures and might be approximated heuristically. Toward this end, we focus on cases where the only concern is the position at some fixed final time, and the control effort. This allows a natural coordinate transformation for the optimal control problem in terms of the so-called flow map. In the transformed coordinates the flow is zero, but the control input (the velocity of the vehicle relative to the flow) is multiplied (and, in more than 1 dimension, rotated) by a time-varying matrix—the Jacobian of the flow map. The definition of what we call the pulled back end cost function provides additional insight and leads to a simple but effective “Lagrangian heuristic control” law, which, in 1d at least, reduces to the optimal control for the case of linear time-invariant flows and quadratic end costs. We demonstrate this control and compare it to the optimal control by solving the associated Hamiltonian Jacobi Bellman (HJB) equation backwards in time with an adaptive 1d grid.

I. INTRODUCTION

The motivating application for this work is high-level control of small AUVs (autonomous underwater vehicles) based on flow field forecasts obtained from data-assimilating ocean models. The availability and accuracy of such forecasts, as well as the number of such vehicles in the ocean, have increased dramatically over the years. For oceanographic sampling, the buoyancy-driven vehicles known as gliders are especially popular thanks to their range, duration, and efficiency. What gliders lack is speed, and this means they have much to gain from path planning. The vulnerability of AUVs such as gliders to ocean currents has been a recurring theme in experiments. As stated in [1], “a methodology to exploit available estimates or predictions of the flow field is of significant interest”. Optimal control provides a natural framework for achieving this.

The problem of interest is stated as follows. Given an n-dimensional time-varying flow field, a fixed final time, a (preferably) smooth end cost function, the initial position-time of a vehicle, and its maximum speed relative to the flow, find the vehicle velocity vector as a function of time that minimizes a given combination of the end cost and the control effort (the integral of the magnitude of the vehicle velocity squared).

There is relatively little work on path planning for AUVs in realistically complex flow fields. A couple of examples that use iterative methods and consider time-varying flow fields are [2] and [3]. Another popular approach, used in [4] for time-invariant, real ocean flow fields is known as RRTs (rapidly exploring random trees). Alternatively, one can take a dynamic programming approach, as in [5] and [6], which consider spatially-complex and, respectively, time-invariant, and time-varying flow fields, but constrain to discrete graphs. The inherent limitation of this has been well established and is remedied by a handful of methods for HJB equations developed in the context of general robot path planning.

Ordered upwind methods (OUMs) [7] generalize the Fast Marching Method of [10] to allow the full continuum of directions of motion in the presence of flows. An heuristically-guided version of OUMs called the FM* algorithm (inspired by A*), was presented in [11] specifically for the application of AUVs. Finally, [12] applies an OUM to solve a static HJB equation for the time-to-go function and globally optimal feedback trajectories of UAVs (unmanned aerial vehicles) in a 2D, time-invariant, but realistic model of a strong tornadic storm. For flow fields stronger that the vehicle, one can use the more general Level Set methods, as demonstrated by [13] for the case of minimum time control in a complex ocean flow field.

Another approach is to solve the HJB equation by a method of characteristics known as the extremal field method. For example, [14] and [15] use related approaches to compute minimum time trajectories in time-varying flow fields from a real ocean model. Other examples of this approach, limited to time-invariant systems, include [16], [17], and [18], although [18] computes the manifold using the more general and more sophisticated framework of continuation methods.

Finally, [19] solves the time-varying HJB equation directly backwards in time, using a semi-Lagrangian, “Godunov” scheme, which effectively transforms the optimal trajectory problem into a point wise optimization problem. To our knowledge, this is the only example of a method related to the one we employ presently. Compared to [19], our method makes use of the exact analytical solution of the linearly constrained quadratic point wise optimization problem, and uses an adaptive grid to more precisely capture the so-called shocks or cusps in the (viscosity) solution of the HJB equation, which we believe to be essential to any shock-capturing scheme.

In addition to this novel HJB scheme, this paper presents a simple one-to-one transformation of coordinates to establish a direct link between a class of AUV motion planning problems and Lagrangian quantities commonly used for visualization of the fluid flows in which they are set. Then we propose a scheme called Lagrangian heuristic control based...
on the structure observed in the transformed problem. This connection has been pursued, for instance, in [20], and presents a control law based on LCS (Lagrangian coherent structures).

The rest of this paper is organized as follows. Section II states the optimal control problem of interest. Section III defines the flow map, proving its key properties. Section IV defines the pulled-back end cost function and proves a bound between it and the optimal cost-to-go function for the case where control effort is not penalized. Section V presents the proposed set of control schema, termed Lagrangian heuristic control. And Section VI presents an algorithm for computing both the optimal and the heuristic controls in 1d, and demonstrates them for simple example.

II. FIXED FINAL TIME OPTIMAL CONTROL PROBLEM

The vehicle motion will be modeled by the state equation

\[ \dot{x} = v(x, t) + u \] (1)

where the state \( x = x(t) \in \mathbb{D} \subset \mathbb{R}^n \) is the vehicle position at time \( t \in [t_{\min}, t_{\max}] \), the control input \( u = u(t) \in \Omega := \{ \dot{u} \in \mathbb{R}^n : \dot{u} < s \} \) is the velocity of the vehicle relative to the flow \( v \), and the parameter \( s \) is the maximum vehicle speed.

For each initial state \( x_0 \) and time \( t_0 \), let the admissible set \( A = A(x_0, t_0) \) be defined as the set of state-input trajectories \( (x, u) \) satisfying (1), for all times \( t \in [t_0, t_{\max}] \), and the initial condition \( x(t_0) = x_0 \). Note that \( x \) and \( u \) may refer to either the instantaneous vectors \( x(t) \) and \( u(t) \) or their trajectories, depending on the context.

Finally, consider the cost functional defined by

\[ J(t_0, x, u) := \int_{t_0}^{t_{\max}} W|u(t)|^2 dt + h(x(t_{\max})), \] (2)

where \( W \) is a weighting parameter and \( h \) is the end cost function.

The optimal trajectory problem is to minimize \( J \) over \( A \), for a specific initial state-time \( (x_0, t_0) \), i.e. to find a state-input trajectory \( (x, u) \) satisfying

\[ V(x_0, t_0) := \min_{(x, u) \in A(x_0, t_0)} J(t_0, x, u). \] (3)

The function \( V : \mathbb{D} \times [t_{\min}, t_{\max}] \rightarrow \mathbb{R} \), if it exists, is the so-called value function. If the optimal trajectory problem satisfies the so-called Principle of Optimality—i.e. the tail of a given optimal trajectory is an optimal trajectory itself (as is the case here)—then a sufficient and necessary condition for the existence of such a function \( V \) is that it is the (viscosity) solution of the time-varying HJB equation

\[ V_t(x, t) = - \min_{u \in \Omega} H(x, u, t, \nabla V(x, t)), \] (4)

with the end condition

\[ V(x, t_{\max}) = h(x), \] (5)

where

\[ H(x, u, t, \nabla V) := (v(x, t) + u) \cdot \nabla V + W|u|^2 \] (6)
is called the Hamiltonian. Moreover, the pointwise argument \( u \) of the minimum (4) defines an optimal feedback control law, whose simulation directly yields solutions of the optimal trajectory problem for as many initial state-times \( (x_0, t_0) \) as desired. The problem of obtaining \( V \) and \( u \) on \( D \times [t_{\min}, t_{\max}] \) will be referred to as the optimal feedback control problem.

III. TRANSFORMED PROBLEM: ZERO FLOW

A recurring theme in the dynamical systems community has been the role of LCS (Lagrangian coherent structures) in transport of passive material as well as active agents such as living organisms and autonomous vehicles. These structures can be used to delineate natural boundaries in the flow and indicate where the effect of small actuations on the ultimate destination of a vehicle is the greatest. Thus using them for control is of interest. But simple and general control laws are lacking. Hence we present a simple transformation of the present optimal control problem that, at least for this particular finite time scenario, makes clear the exact relationship between its solution trajectories and these flow structures, and leads us to derive a preliminary class of such control laws. The transformation is based on the so-called flow map, which underlies the numerical computation of LCS via FTLEs (finite time Lyapunov exponents).

A. The flow map

Let the flow map \( X : D \times [t_{\min}, t_{\max}]^2 \rightarrow D \) be defined as the mapping of points \( (x, t) \) in space-time along the (passive particle) trajectories of the flow field \( v \) forward or backward to a given time \( T \), that is, the solution of

\[ \frac{d}{dT} X(x, t, T) = v(X(x, t, T), T). \] (7)

for which \( X(x, t, T) = x \). In addition, assuming the Jacobian \( v_x \) is defined, the gradient of the flow map \( X_x : D \times [t_{\min}, t_{\max}]^2 \rightarrow \mathbb{R}^{n \times n} \) is also defined.

B. Lemma: transitive property of the flow map

For all \( (x, t, T, \tau) \in \mathbb{R}^3 \times [t_{\min}, t_{\max}]^3 \),

\[ X(X(x, t, T), \tau, T) = X(x, t, T). \] (8)

The proof for this lemma is straightforward and thus omitted.

C. Lemma: invariance of the (fixed final time) flow map along passive particle trajectories

Suppose \( (x, T) \in \mathbb{D} \times [t_{\min}, t_{\max}] \) and \( y(T) := X(x, t, T) \) for all \( T \in [t_{\min}, t_{\max}] \). Then, for all \( T_1 \) and \( T_2 \in [t_{\min}, t_{\max}] \),

\[ X(y(T_1), T_1, t_{\max}) = X(y(T_2), T_2, t_{\max}). \] (9)

The proof is as follows.

\[ X(y(T_1), T_1, t_{\max}) := X(X(x, t, T_1), T_1, t_{\max}), \]

\[ = X(x, t, t_{\max}), \text{ by (8)}, \]

\[ = X(X(x, t, T_2), T_2, t_{\max}), \text{ by (8)}, \]

\[ = X(y(T_2), T_2, t_{\max}). \]
Note, the conclusion of the Lemma (that the flow map is constant along the nominal trajectory $y$) is true if and only if, for all $T \in [t_{\text{min}}, t_{\text{max}}]$,

$$
\frac{d}{dT} X(y(T), T, t_{\text{max}}) = 0
$$

(10)

$$
\iff \text{for all } T \in [t_{\text{min}}, t_{\text{max}}], \quad X_y(y(T), T, t_{\text{max}}) = 0
$$

(11)

\begin{align*}
\text{In particular, } & \quad X_y(y(t), t, t_{\text{max}}) \cdot y(y(t), T) + X_t(y(T), T, t_{\text{max}}) = 0. \\
\text{i.e. } & \quad X_y(x, t, t_{\text{max}}) \cdot v(y(t), t) + X_t(x, t, t_{\text{max}}) = 0.
\end{align*}

In summary, the Lemma also states that the flow map $X_y$ solves the advection PDE $X_y \cdot v + X_t = 0$. The boundary condition is of course $X_y(x, t, t_{\text{max}}) = x$.

D. Theorem: change of the (fixed final time) flow map along controlled trajectories

Suppose that the final time is fixed at $T = t_{\text{max}}$ and, moreover, the flow map and its gradient are evaluated along a particular $(x, u) \in A(x_0, t_0)$ so that, for all $t \in [t_0, t_{\text{max}}]$, $X(t) := X(x(t), t) := X(x(t), t, t_{\text{max}})$ and, similarly, $X_u(t) := X_u(x(t), t, t_{\text{max}})$. As such, we have the following theorem: $\frac{d}{dT} X(t) = X_u(t) \cdot u(t)$, or simply

$$
X = X_u \cdot u.
$$

(12)

This transformed state equation is analogous to (1), and the analog of (2) is

$$
J(t_0, X, u) := \int_{t_0}^{t_{\text{max}}} W(u(t))^2 dt + h(X(t_{\text{max}})).
$$

(13)

The proof is as follows.

$$
\frac{d}{dT} X(x(t), t, t_{\text{max}}) = X_u(x(t), t, t_{\text{max}}) \cdot u(t) + X_t(x(t), t, t_{\text{max}}),
$$

by the chain rule,

$$
= X_u(x(t), t, t_{\text{max}}) \cdot [v(x(t), t) + u(t)] + X_t(x(t), t, t_{\text{max}}),
$$

by the definition of $A(x(t), t_0)$,

$$
= X_u(x(t), t, t_{\text{max}}) \cdot u(t) + X_u(x(t), t, t_{\text{max}}) \cdot v(x(t), t) + X_t(x(t), t, t_{\text{max}}),
$$

by the distributive property,

$$
= X_u(x(t), t, t_{\text{max}}) \cdot u(t),
$$

by Lemma III-C (11).

Thus the optimal control problem is transformed from $(x, t)$ space to $(X, t)$ space. If, for instance, $h(x) = |x - x_f|$, then instead of seeking a trajectory from $(x_0, t_0)$ to (the vicinity of) $(x_f, t_{\text{max}})$, one seeks a trajectory from $(x_0, t_0) := (X(t_0), t_0)$ to $(X_f, t_{\text{max}}) := (x_f, t_{\text{max}})$—since $X(t_{\text{max}}) = X(x(t_{\text{max}}), t, t_{\text{max}}) = x(t_{\text{max}})$. Note that $x(t)$ is easily recovered by $x(t) = X(x(t), t_{\text{max}}, t)$, that is, $X(X(x(t), t, t_{\text{max}}), t_{\text{max}}, t)$ (by (8)). Given this one-to-one mapping between $x$ and $X$, the HJB equation is well defined in the new space; however, its numerical solution, though not yet attempted, is not thought to be fundamentally easier.

The advantage, rather, is primarily conceptual. Instead of being simultaneously advected by $v$ and driven by $u$ per (1), backwards in time, the optimal state trajectories and the surface of the value function $V$ are driven solely by $X_u \cdot u$; the maximum size of the effective control depends on the direction of $u$ but it is zero if (and only if) $u = 0$. Also, in addition to $x$ (or $X$), the $n$ by $n$ matrix $X_u$ depends on time $t$, but in a way that is easy to visualize in $X$ space.

IV. APPROXIMATION OF $V$ (FOR THE $W = 0$ CASE) BY THE PULLED-BACK END COST FUNCTION $h'$

We now present a theorem for an explicit bound on the difference between the value function $V$ of (4) and what we’ll call the pulled-back end cost function, defined in terms of the flow map $X$ as

$$
h'(x, t) := h(X(x, t, t_{\text{max}})).
$$

(14)

The bound is relatively conservative. Nevertheless, it highlights the connection between the optimal control and the hyperbolicity of the flow field, and it provides theoretical grounds for heuristic control schema based on the pulled-back end cost function, at least for small vehicle speed $s$ and small time horizons $t_{\text{max}} - t$. Note that for $W = 0$ the optimal control is non-unique but $V$ is still well-defined.

A. Assumptions

1) $W = 0$ in (2), and thus $V = \min_{(x, u) \in A(x_0, t_0)} h(x(t_{\text{max}})).$

2) There exists an upper bound $c_v$ on the magnitude of the Jacobian $X_u$ of the flow.

3) There exists an upper bound $c_h$ on the magnitude of the gradient $h_x$ of the end cost.

B. Lemma: bound on flow map Jacobian

Suppose $(x, t, T) \in \mathbb{D} \times [t_{\text{min}}, t_{\text{max}}]^2$. Then

$$
|X_u(x, t, T)| \leq \exp c_v(T - t).
$$

(15)

We omit the details of this proof and simply state that it follows from the fact that

$$
X_u(x(t, T), I + \int_T^t v(x(t, \tau), \tau) \cdot X_u(x(t, \tau)) d\tau).
$$

(16)

C. Lemma: bound on difference between (fixed final time) flow map and the final state

Suppose $(x, u, \tau) \in \mathbb{D} \times [t_{\text{min}}, t_{\text{max}}]$ and $(y, u) \in A(x, t)$. Then

$$
|X(x, t, t_{\text{max}}) - y(t_{\text{max}})| \leq s(t_{\text{max}} - t) \exp \{c_h(t_{\text{max}} - t)\}.
$$

(17)

The proof is as follows. Consider

$$
X'(T) := X(y(T), T, t_{\text{max}}),
$$

that is, the flow map evaluated along the controlled trajectory, for times $T \in [t_{\text{min}}, t_{\text{max}}]$. This "trajectory" can be written in integral form as

$$
X'(T) = X'(t) + \int_t^T \frac{d}{d\tau} X'(\tau) d\tau.
$$

6184
But also, by Theorem III-D, we have \( \frac{d}{dT} X_y(T) = \frac{d}{dT} X(y(T), T, t_{\text{max}}) \cdot u(T) \), and thus
\( X_y(T) = \int_T^{t_{\text{max}}} X_x(y(T), T, t_{\text{max}}) \cdot u(\tau) d\tau \), and thus
\( X_y(T) = \int_T^{t_{\text{max}}} X_x(y(T), T, t_{\text{max}}) \cdot u(\tau) d\tau \), and thus
\begin{align*}
X(y(t_{\text{max}}), t_{\text{max}}) = X(x, t, t_{\text{max}}) \\
+ \int_t^{t_{\text{max}}} X_x(y(\tau), \tau, t_{\text{max}}) \cdot u(\tau) d\tau
\end{align*}
and in particular
\begin{align*}
X(y(t_{\text{max}}), t_{\text{max}}) &= X(x, t, t_{\text{max}}) \\
+ \int_t^{t_{\text{max}}} X_x(y(\tau), \tau, t_{\text{max}}) \cdot u(\tau) d\tau
\end{align*}
and thus
\begin{align*}
X(x, t, 0) - y(0) &= -\int_{t_{\text{max}}}^{t_{\text{max}}} X_x(y(\tau), \tau, t_{\text{max}}) \cdot u(\tau) d\tau
\end{align*}
which yields the bound
\begin{align*}
|X(x, t, t_{\text{max}}) - y(t_{\text{max}})| &= \left| -\int_{t_{\text{max}}}^{t_{\text{max}}} X_x(y(\tau), \tau, t_{\text{max}}) \cdot u(\tau) \cdot d\tau \right| \\
&\leq \int_{t_{\text{max}}}^{t_{\text{max}}} |X_x(y(\tau), \tau, t_{\text{max}})| \cdot |u(\tau)| \cdot d\tau \\
&\leq \int_{t_{\text{max}}}^{t_{\text{max}}} \exp\{c_v(t_{\text{max}} - \tau)\} \cdot s \cdot d\tau,
\end{align*}
by Lemma IV-B and the definition of \( A(x_0, t_0) \),
\begin{align*}
&\leq \int_{t_{\text{max}}}^{t_{\text{max}}} \exp\{c_v(t_{\text{max}} - t)\} \cdot s \cdot d\tau,
&\text{since } (t_{\text{max}} - \tau) \leq (t_{\text{max}} - t),
&\leq \exp\{c_v(t_{\text{max}} - t)\} \cdot s \cdot \int_{t_{\text{max}}}^{t_{\text{max}}} d\tau,
&= \exp\{c_v(t_{\text{max}} - t)\} \cdot s \cdot \int_{t_{\text{max}}}^{t_{\text{max}}} d\tau.
\end{align*}

D. Theorem: bound on the difference between value function and pulled-back end cost function

If the assumptions of Section IV-A are satisfied, then
\begin{align*}
h'(x, t) - V(x, t) &\leq c_h s(t_{\text{max}} - t) \exp\{c_v(t_{\text{max}} - t)\}.
\end{align*}
This implies \( h' \rightarrow V \) exponentially as \( t \rightarrow t_{\text{max}} \) and linearly as \( s \rightarrow 0 \).

The proof is as follows. By the definition of \( V \), \( \exists y,u \in A(x, t) \) for which \( V(x, t) = h(y(t_{\text{max}})) \). And
\begin{align*}
h'(x, t) - V(x, t) &= h(X(x, t, t_{\text{max}})) - V(x, t) \\
&= [h(X(x, t, t_{\text{max}})) - V(x, t)] \\
&\leq c_h [X(x, t, t_{\text{max}}) - y(t_{\text{max}})] \\
&\leq c_h s(t_{\text{max}} - t) \exp\{c_v(t_{\text{max}} - t)\},
\end{align*}
by Lemma IV-C.

V. LAGRANGIAN HEURISTIC CONTROL

Unlike \( x, X \) can be driven in any direction, but how fast or slow depends on \( X_\star \). Often the flow \( v \) is area/volume preserving. If so, the effect of \( X_\star \) is to rotate the control vector \( u \) and/or multiply it—either magnifying it, if its rotated version is aligned with what we’ll call the “fast” direction, or shrinking it, in the “slow” direction. Roughly speaking, this effect grows exponentially backwards in time. At the final time \( t = t_{\text{max}} \), \( X_\star \) is the identity matrix, but, the larger \( t_{\text{max}} - t \), the more the progress towards \( X_f \) is either helped or hindered by \( X_\star \), depending whether the maximum eigenvalue is aligned with or transverse to the desired path.

The basic scheme revealed by these observations is first of all to choose \( \frac{d}{dt} h(X(t)) = h_x(X(t))Xu \), and, second, to choose \( u \leq s \). In the worst case \( |u| = s \), but Section VI derives a better alternative for \( W > 0 \). This control law is greedy in the sense that it guarantees monotonic descent of \( h(X) \). It needs only what’s happening in the immediate vicinity of the passive particle trajectory that it’s on. But it does look ahead in time. It’s also important that it considers the effect of the hyperbolicity of the flow field, in addition to the rotation; in a slightly more naïve strategy, one might choose \( u \) to make the path of \( X \) a straight line down the slope of \( h \), and not exploiting the large eigenvalues. This phenomenon of hyperbolicity combined with rotation, which has recently given rise to the theory of mesohyperbolicity [22], is interesting in its own right, and has a variety of applications besides control.

VI. 1D CASE STUDY

The above ideas are generally applicable in 2 or higher dimensions; however, for the sake of presentation, we now focus on the special case of a 1D flow. That said, instead of focusing on the limit case of small vehicle speeds \( s \) (and energy weight \( W = 0 \)), we focus on the case where \( s \) is large, but where the effective vehicle speed is softly constrained by larger values of \( W \). This case proved more interesting, since, overall, it involves less constraint on the vehicle trajectories, allowing them to more freely exploit the structure of the flow—in the 1D case stretching or shrinking. In this setting we present the following:

- the analytical solution of the HJB equation for the case of 1D linear time-invariant (LTI) flows and a quadratic end cost,
- the derivation of a heuristic control law for general 1D nonlinear time-varying flows that reduces to the optimal control for the LTI case,
- an algorithm for computing both the optimal control (solving the HJB equation) and the heuristic control (in nonlinear, time-varying 1D flows), and
- the results of the algorithm for one 1D nonlinear but time-invariant example flow.

The results illustrate how the algorithm works, the general effectiveness of both the optimal and the heuristic controls, and the limitations of the heuristic control. Extension of the LTI solution and general heuristic control law to 2 or higher dimensions is presumably straightforward but out of the scope of this paper. The present algorithm for computing the optimal and heuristic controls is simple but would not be practical for 2 or higher dimensions. Moreover, it does not take advantage of the local nature of the heuristic control law, as will be explained at the end of Section VI-C. Better algorithms are the subject of ongoing work.

A. Analytical solution for 1D LTI flows and a quadratic end cost

Consider the case where \( v(x, t) = v(x) = v(x) = ax \), for some scalar \( a \), and the end cost is \( h(x) = h(x) = x^2 \). For simplicity let \( t_{\text{max}} = 0 \) so that \( t \leq 0 \) in the domain of the solution, and \( V(x, 0) = h(x) = x^2 \). From Equation (6), the Hamiltonian is \( H = (ax + u)V_x + Wu^2 \) (where \( u = u \).
is the control and \( V_x = \nabla V \) is the gradient of the value function) and, assuming for now that \( s \gg 0 \), is minimized by \( u = -\frac{1}{4W} V_x \). Thus the HJB equation, (4), reduces to
\[
V_t = \frac{1}{4W} V_x^2 - axV_x,
\]
and the end condition (5) is \( V(x, 0) = h(x) = x^2 \). Rather than solve this problem directly, we first transform it using the flow map, as alluded to in Section III. In the present LTI case, this makes things only slightly easier. But we are ultimately interested in more general flows.

The flow map is given by \( X(x, t, T) = x e^{a(T-t)} \) and in particular we define the transformed coordinate variable \( x' \) by \( x' = x' (x, t) := X(x, t, t_{\max}) = X(x, t, 0) = xe^{-at} \) (formerly loosely referred to as \( X \) in Section V). The Jacobian \( X_x \) is the new scalar \( x'_x = e^{-at} \) and, from Equation (12), the dynamics of the new coordinate \( x' \) for a given controlled trajectory \( x(t) \) are \( \dot{x}' = x'_x u = e^{-at} u \). Thus we can define the new cost-to-go \( V'(x', t) = V'(x', t) = V(x, t) \), the new Hamiltonian \( H' = H'(x', u, t, V'_{x'}) = e^{-at} u V'_{x'} + W u^2 \), the new control \( u = -\frac{1}{4W} e^{-at} V'_{x'} \), and the new HJB equation
\[
V'_t = \frac{1}{4W} e^{-2at}(V'_{x'})^2.
\]
The boundary condition is \( V'(x', 0) = V(x, 0) = x^2 \), since \( x' = x \) at time \( t = 0 \).

To solve this new, transformed problem, we look for a solution of the form
\[
V'(x', t) = K(t)(x')^2,
\]
and separate the variables. (20) becomes \( K_x(x')^2 = \frac{1}{4W} e^{-2at}(2x'K)^2 \) (where \( K_t := \frac{dK}{dt} \)) and the end condition becomes \( K(0) = 1 \). The solution of this ODE is
\[
K = W \left( W + \frac{e^{-2at} - 1}{2a} \right)^{-1}
\]
if \( a \neq 0 \) and \( K = W(W - 1)^{-1} \) if \( a = 0 \). This yields
\[
u = -\frac{1}{2W} e^{-at} K(2x'), \quad \text{or (for } a \neq 0 \text{)},
\]
\[
u = -e^{-at} \left( W + \frac{e^{-2at} - 1}{2a} \right)^{-1} x'.
\]

B. Generalization of LTI Lagrangian heuristic control law to general 1d nonlinear time-varying flows

The control law (23) may also be written as
\[
u = \frac{1}{2} \left( W + \frac{(x')^2 - 1}{2v_x} \right)^{-1} h'_x,
\]
where \( h'_x := \frac{\partial}{\partial x} h(x, t) \), assuming \( h'(x, t) := h(x'(x, t)) \) denotes the pulled back end cost function, and \( v_x := \frac{\partial}{\partial x} v(x, t) \) is the Jacobian of the flow. In other words, the optimal control is to go down the gradient of the pulled back cost function, with a very particular speed. The speed depends inversely on the weight \( W \) (as in the minimum \( u = -\frac{1}{4W} V_x \) of the value function-dependent Hamiltonian) but also depends on \( v_x \) and \( x'_x \), which take into account the general sensitivity of the flow as well as time remaining.

This quantity is easily computable for any flow \( v \) and end cost \( h \). However, numerically, it is not very well-behaved. Specifically, it leads to chattering—bouncing back and forth of the trajectory—in the smooth but very narrow trenches that tend to appear in the surface of \( h' \). Hence we chose to substitute \( e^{-v_x t} \) for \( x'_x \), sacrificing some of the benefit of the knowledge of \( x'_x \) in exchange for smoother trajectories. This worked well, at least within the framework of the following algorithm. Also note that in practice the assumption \( s \gg 0 \) in (19)-(24), purely for simplifying the presentation, is dropped by simply saturating the resulting control into \( \Omega = [-s, s] \).

C. Algorithm for optimal and heuristic control

Besides replacing \( x'_x \) with \( e^{-v_x t} \) in (24), the present algorithm avoids using any single approximation of \( h' \) or \( V_x \). Instead, it directly minimizes the correct combination of control effort at a particular time and the value of \( h' \) or \( V \) interpolated linearly from the subsequent time step. The algorithm is first order in space and time, but uses an adaptive 1d grid in each time slice to capture the extreme variations in \( h' \) and or the cusps in the so-called viscosity solution \( V \) known as shocks. Given the domain \( \mathbb{D} = [x_{\min}, x_{\max}] \) and the problem parameters already described, the main algorithm parameters are simply the number of time steps \( n_t \) and an error tolerance \( \epsilon_{\text{max}} \). First we’ll explain the adaptive grid, then the computation of \( V \), and finally the computation of trajectories via simulation of the optimal and heuristic controls. The computation of \( h'(x, t) = h(x'(x, t)) \) at a single point \((x, t)\) comes down to a simple trajectory computation to obtain \( x' \); for simplicity and consistency we do this via the explicit Euler method.

The adaptive grid approximates an arbitrary function \( f : D \to \mathbb{R} \) (e.g. a time slice of \( h'(x, t) \) or \( V(x, t) \)) using a binary tree of grid elements \( E = [x_L, x_R] \). Grid refinement consists of bisecting (leaf) elements until each one satisfies two simple criteria. First, \( |e| \leq \epsilon_{\text{max}} \), where \( e \) is the linear error estimate \( e := (f(x_L) + f(x_R))/2 - f(x) \) at the midpoint. Second, the size of neighboring grid elements may not vary by more than a factor of two. In other words the grid is graded. This gradedness requirement is implemented via pointers to neighbor elements and simple recursion. There is no need to provide an explicit lower bound on the grid element size, except for if \( f \) is discontinuous.

Whereas the time slices of \( h \) can be computed in any order (since we don’t attempt to re-use later time slices in the computation earlier ones), \( V \) is computed backwards in time from the slice \( V(x, 0) = h(x) \). The core procedure is that of computing a single value \( V(x, t) \) (as dictated by the adaptive grid algorithm) from the subsequent slice \( V(\cdot, t + \Delta t) \), described briefly in the following pseudocode:

1. Set \( V(x, t) = \infty \).
2. For each element \( E \) of the \( V(\cdot, t + \Delta t) \) grid intersecting the reachable interval
   \( R = [x + (v - s)\Delta t, x + (v + s)\Delta t] \):
   a) Set \( V^* = \min_{x^* \in \mathbb{R} \cap E} \{ V(x^*, \cdot + \Delta t) + W(u^*)^2 \Delta t \} \).

6186
b) Set $V(x,t) = \min \{V(x,t), V^*\}$.

The variables $u^* := (x^* - x)/\Delta t - v \in \Omega = [-s,s]$ and $x^* = x + (v + u^*)\Delta t \in \mathbb{R}$ in Step 2(a) effectively provide a semi-Lagrangian discretization of the present HJB equation, namely

$$\min_{u \in \Omega} \{ V_t + (v + u) \cdot V_x + W u^2 \} = 0. \quad (25)$$

The term $V(x^*, t+\Delta t)$ is defined by linear interpolation from the adaptive grid. This allows for the exact solution of the constrained minimum $V^*$.

After computing $V$ or $h'$ on a sequence of $n_t + 1$ adaptive 1d grids, one can compute trajectories for specific initial states $x_0$ (and times $t_0$) by direct simulation of the respective feedback control law. In both cases, the exact same minimization procedure of the pseudocode above is used at each time step of each simulated trajectory. For the heuristic trajectories, one simply replaces $V$ with $h'$ and $W$ with $W + e^{-2u_{max}^2/\Delta t} - 1$, as described in Section VI-B. This approach for computing the heuristic control, globally, as with the optimal control, was chosen for convenience, and because the global picture of $h'$ itself is very illustrative. However, one of the potential benefits of the heuristic control over the optimal control would be the ability to compute it locally without computing it globally. One of the problems with the local approach was the chattering, mentioned in Section VI-B, that tends to result from a typical discretization of $h'_x$ in (24).

D. Numerical results for time-invariant nonlinear 1d flow

The algorithm of Section VI-C was used to compute the optimal control (and cost-to-go) and Lagrangian heuristic control (and pulled-back end cost $h$) for the flow $v(x,t) = \sin(\pi x)$ and the end cost $h(x) = -\sin(\pi x)$, with $s = 1$, $W = 10$, $D = [x_{min}, x_{max}] = [0,3]$, and $[t_{min}, t_{max}] = [-2,0]$. The algorithm parameters were $c_{max} = 1e-6$ and $n_t = 200$. The results are shown in Figures 3-9. Note that $s = 1$ and $W = 10$ turned out to be large enough that the hard bound $h$ had no real effect, except to speed up the grid search procedure.

First of all, the flow and its action on the end cost in backward time are best visualized in terms of $h'(x,t)$, shown in Figure 7. The end cost is 0 at all four visible nodes but $x = 1$ and $x = 3$ are stable and $x = 0$ and $x = 2$ are unstable. The minima of $h(x)\prime$ lie between fixed points and are more easily reached from below, after lingering near the unstable node.

The question, besides which of the two minima to seek, is whether this strategy of arriving from below, precisely at the minimum of $h$, is worth the initial effort of escaping from the basin of the stable node. These questions are answered in terms of three shocks, which divide the trajectories into four distinct groups: two for each minimum of $h$. For example, in Figure 3, the control jumps from $-s$ (blue) to 0 (green) at $x(0) \approx 0.1$, from 0 to $s$ (red) at $\approx 1.9$, and from $-s$ to 0 again at 2.1. For $x(0) \in [1.9,2.1]$ and $[0.0,0.1]$ the optimal strategy is the one described above. For $x(0) \in [0.1,1.9]$ and $[2.1,3.0]$ the optimal strategy is to accept one’s fate so to speak and conserve energy initially, and then apply a (very small) burst of energy near the end, to get slightly closer to the minimum of $h$. These latter trajectories (the majority) are almost indistinguishable in Figures 5 and 6.

This flow was chosen to highlight the non-greedy nature of the optimal control. This is best captured in Figure 6 by the evaluation along the trajectories of the pulled back end cost—the cost-to-go if one were to turn off the control. Several trajectories climb up and over the maximum of this function in order to eventually reach its minimum.

As shown in Figure 9, the pulled back end cost is monotonically decreasing along Lagrangian heuristic trajectories, but these don’t get as far down the slope as the optimal trajectories; the ability to compute the control locally makes it inherently greedy. Only the two trajectories initialized right next to the local minima of $h$ make it there. Nevertheless, the heuristic control appears more or less identical to the optimal control for all except about 3 trajectories in the vicinity of each of the three shocks. This suggests that it does have potential as an alternative to the optimal control. As shown in Section VI-A it is in fact identical to the optimal control for the case of LTI flows and quadratic end costs, if not other more general flows; for example, adding a purely time-varying component to an LTI flow does not change the optimality of the heuristic control law. In addition to improving the algorithm, the performance of the present Lagrangian heuristic control law in 2d and for more general flows is the subject of ongoing work.

VII. Conclusion

This paper explored the relationship between optimal trajectories in a flow field and the spatial structure of that flow field, as captured in by the so-called flow map, its Jacobian, and what we call the pulled-back end cost function. Specifically, the quantity being optimized was a combination of control effort and the value of an end cost at some fixed final time. First, the pulled back end cost $h'$ was shown to be a reasonable first approximation of the cost-to-go function $V$—the solution of the associated time-varying HJB equation—when the vehicle speed $s$ is small. Second, a “Lagrangian
Fig. 2. Level of refinement along the adaptive grid for $V(x, -2)$ (shown in Figure 1). The maxima correspond to the shocks (cusps).

Fig. 3. Optimal trajectories and feedback control law.

Fig. 4. Optimal trajectories and cost-to-go function.

Fig. 5. Control along optimal trajectories.

Fig. 6. Pulled back end cost along optimal trajectories.

Fig. 7. Heuristic trajectories and pulled back end cost function.
temporarily ascending

**Fig. 8.** Control along heuristic trajectories.

**Fig. 9.** Pulled back end cost along heuristic trajectories.

heuristic control” law was defined, for simplicity in the 1d setting, and shown to reduce to the optimal control for the case of linear time-invariant flows and quadratic end costs. Finally, an algorithm was presented for computing both the heuristic control and the optimal control, on adaptive 1d grids at each time slice, and the solutions were demonstrated for a simple example. The heuristic control performed generally well, except in regions where the optimal control required temporarily ascending \( h' \) to ultimately descend \( h' \) farther. Future work include making the algorithm more efficient and extending it to higher dimensions.

**VIII. ACKNOWLEDGEMENTS**

The authors gratefully acknowledge funding from The Office of Naval Research through grants N00014-07-1-0587 and N00014-10-1-0611.

**REFERENCES**


