2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

G-Learning: Equivariant Indirect Optimal Control with Generating Function

Taeyoung Lee

Abstract— **This paper presents a new formulation of data-driven, learning-based optimal control with the Hamilton-Jacobi theory. In contrast to the common practice of reinforcement learning for dynamical systems, where the control policy is parameterized by deep neural network and the control parameters are optimized directly, we propose to adopt the indirect optimal control where the necessary conditions for optimality are first constructed by Pontryagin's minimum principle. Then, the resulting two-point boundary value problem is solved by learning the generating function associated with the optimality conditions that are considered as a Hamiltonian system. Further, it is shown that the sampling efficiency can be improved when there exists an invariance in the dynamics. The foremost benefit is that this provides a set of optimal controls for varying boundary conditions, which cannot be systematically addressed in the policy gradient.**

I. Introduction

Deep reinforcement learning has been successfully applied to various fascinating and challenging problems. For dynamic systems with continuous action and state spaces, the most common approaches include Trust Region Policy Optimization (TRPO) to gradually update the control policy [1], Proximal Policy Optimization (PPO) to achieve the goal of TRPO with less computational load [2], and soft actor-critic (SAC) to balance exploration and exploitation [3].

Despite various successful applications, there are several obstructions in utilizing the above policy gradient methods for continuous dynamics that commonly appear in robotics or aerospace engineering. The foremost limitation is that multiple control objectives should be consolidated into a *single* reward function to be optimized. While this is natural for certain types of MDPs like Atari games or chess, we often encounter multiple objectives in practice, such as the boundary condition, inequality constraints, or control regulation beyond the principal goal. When there are multiple control objectives, the reward is often chosen as a weighted sum of several components, and the functional form of each objective function and the corresponding weighting parameters should be selected by trial-and-error to achieve reasonable performance, resulting in the reward engineering problem. Next, as the control policy is represented by deep neural networks with a large capacity of function approximation, the control input may exhibit unintended behaviors such as high-frequency chattering. This motivates additional consideration to improve smoothness of the control, such as a regularization approach presented in [4]. Finally, our prior knowledge of the optimal control of dynamic systems is not well reflected in the learning process.

These are not surprising, considering that the policy gradient methods are direct optimal control. As a numerical optimization technique is applied to construct a sequence of improving approximation to the optimal solution iteratively, it is unclear how the iterations evolve. Nor is the structure of the converged solution comprehensible.

To address these, we present an alternative formulation of learning-based controls in the framework of indirect optimal control, referred to as G-learning. Assuming that the dynamics are given, we first construct necessary conditions for optimality with the variational principle, which are represented by a two-point boundary value problem for the state and the co-state. According to the Hamilton-Jacobi theory, it can be solved in a single step once the associated generating function is constructed [5], [6], [7], thereby yielding optimal feedback controls. This has been extended to discrete-time systems [8] with the discrete Hamilton-Jacobi theory [9]. In these works, the generating function was constructed by a Taylor series expansion or a quadratic form recursively. As such, there is a limitation in applying this method for higher-dimensional systems.

In this paper, we propose to construct the generating function iteratively by machine learning. Specifically, we generate a set of optimal trajectories for varying boundary conditions, which are used to train a neural network representing the generating function. Then, the optimal control is directly constructed by the learned generating function.

Compared with the policy gradient, this provides optimal controls for varying boundary conditions from a learned generating function, i.e., there is no need for retraining or transfer learning. Further, as the boundary condition is enforced by the optimality condition, we do not have to augment the reward or the objective function with an additional penalty term representing the error in satisfaction of the boundary conditions. Also, this is well-suited to incorporate other types of equality or inequality constraints easily in the optimality conditions by alternating the type of the generating function. In addition, we extend this to an equivariant G-learning to show that the sampling efficiency of the learning can be improved if there exists any symmetry in the underlying dynamics and the objective function.

In short, this paper presents a new formulation of data-driven optimal controls, based on the indirect optimal control and the Hamilton-Jacobi theory, which is complementary to the policy gradient of reinforcement learning.

Taeyoung Lee, Mechanical and Aerospace Engineering, George Washington University, Washington, DC 20052. tylee@gwu.edu

The research was partially supported by NSF under the grant CNS-1837382 and by AFOSR under the award MURI FA9550-23-1-0400.

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

II. Problem Formulation

Consider a discrete-time dynamic system defined by

$$q_{k+1} = f(q_k, u_k), \tag{1}$$

where $q \in \mathsf{Q}$ is the state in the configuration space Q and $u \in \mathbb{R}^m$ is the control input.

We formulate an optimal control problem to minimize the cost function:

$$\mathcal{J}(u_0, \dots u_N) = \sum_{k=0}^{N-1} L(q_k, u_k) \tag{2}$$

for a running cost $L : \mathsf{Q} \times \mathbb{R}^m \to \mathbb{R}$ with the given initial and terminal states (q_0, q_N) over a fixed finite period N. While we focus on hard terminal constraints in this paper, the subsequent developments are readily extended to other types of soft constraints or state equality/inequality constraints.

A. Necessary Conditions for Optimality

Here we briefly summarize the optimality conditions for discrete-time systems [10]. Define an augmented cost function:

$$\overline{\mathcal{J}} = \sum_{k=0}^{N-1} p_{k+1} \cdot q_{k+1} - H(q_k, p_{k+1}, u_k),$$

where the Lagrange multiplier or the co-state is denoted by $p_k \in \mathsf{T}^*\mathsf{Q}$, and the Hamiltonian is

$$H(q_k, p_{k+1}, u_k) = -L(q_k, u_k) + p_{k+1} \cdot f(q_k, u_k). \tag{3}$$

Here $\cdot$ denotes the pairing between the tangent space and the cotangent space, and it is interpreted as the usual dot product assuming $\mathsf{T}^*\mathsf{Q} \simeq \mathsf{Q} \times \mathsf{Q}$. Under any discrete trajectory satisfying (1), we have $\overline{\mathcal{J}} = \mathcal{J}$.

It is well known that the optimal control is obtained by

$$u_k = \arg\max_{\tilde{u}} H(q_k, p_{k+1}, \tilde{u}). \tag{4}$$

We assume that H is regular such that the unique optimal control can be determined as a function of the state and the co-state, i.e.,

$$u_k = U(q_k, p_{k+1}) \tag{5}$$

for $U : \mathsf{Q} \times \mathsf{T}^*\mathsf{Q} \to \mathbb{R}^m$. Substituting this into (3), we obtain the optimal Hamiltonian

$$H(q_k, p_{k+1}) \triangleq H(q_k, p_{k+1}, U(q_k, p_{k+1})). \tag{6}$$

The necessary conditions for optimality are given by the following discrete Hamilton's equations:

$$q_{k+1} = D_2 H(q_k, p_{k+1}), \tag{7}$$

$$p_k = D_1 H(q_k, p_{k+1}), \tag{8}$$

where D_1 denotes the derivative with respect to the first argument, and D_2 is defined similarly. For a given (q_0, p_1), the optimal control u_1 is determined by (5). Then, (q_1, p_2) can be obtained by (7) and (8), respectively. These yield the optimal flow $(q_k, p_{k+1}, u_k) \to (q_{k+1}, p_{k+1}, u_{k+1})$ up to the terminal state q_N.

As such, the optimal control problem is interpreted as a two-point boundary value problem to identify the initial multiplier p_1 that ensures the terminal state becomes its desired value. Further, (5) can be utilized as an optimal feedback control, if the value of p_{k+1} to satisfy the terminal boundary condition can be computed in real-time.

Solving the two-point boundary value problem is computationally involved in general. However, it has been proposed that we can construct an algebraic equation to relate the current co-state and the terminal state exploiting that the necessary conditions for optimality are given by a Hamiltonian dynamics as presented in (7) and (8).

B. Discrete Hamilton-Jacobi Theory

More specifically, the transformation between (q_k, p_k) and (q_N, p_N) is canonical, and therefore it can be described by a generating function. Depending on the choice of the independent variables, there are several types of generating functions. Here, we show one type of generating function and the corresponding canonical transform that are useful for the presented optimal control.

Proposition 1: [8] Define the generating function of Type 1 as follows.

$$G_1(q_k, q_N) = -\sum_{i=k}^{N-1} [p_{i+1} \cdot q_{i+1} - H(q_i, p_{k+1})], \tag{9}$$

where two input arguments (q_k, q_N) of the generating function are considered as independent variables, and the remaining state/co-state are chosen according to (7) and (8).

The corresponding canonical transform is

$$p_k = D_1 G_1(q_k, q_N), \quad -p_N = D_2 G_1(q_k, q_N). \tag{10}$$

In (9), the dependency of the generating function on the time step is suppressed, e.g., $G_1(q_k, q_N)$ is a shorthand for $G_1(k, N; q_k, q_N)$. In fact, as the discrete dynamics (1) and the cost function (2) are time-invariant, the generating function depends on the difference of time, $N - k$.

The generating function evolves according to the Hamilton-Jacobi equations as follows.

Proposition 2: [8] The generating function satisfies the following discrete Hamilton–Jacobi equation:

$$\begin{aligned} G_1(q_{k-1}, q_N) = G_1(q_k, q_N) - D_1 G_1(q_k, q_N) \cdot q_k \\ + H(q_{k-1}, D_1 G_1(q_k, q_N)), \end{aligned} \tag{11}$$

with the boundary condition $G_1(q_N, q_N) = 0$.

C. Optimal Control with Generating Function

Consider a boundary value problem to find two unknown variables of (q_k, p_k, q_N, p_N) when the other two variables are given. This can be easily addressed by the canonical transforms between them. For example, if the current state q_k and the terminal boundary condition q_N are given, the corresponding co-state at the current time is simply given by the first equation of (10), thereby addressing the two-point boundary value problem of the indirect optimal control.

TABLE I

PROCEDURES OF G-LEARNING

```
procedure $G_{nn}$ = TRAINING($Q_0, P_0, Q_N, n_{\mathrm{data}}$)
    Set $\mathcal{D} = \{\}$
    repeat
        Sample $(q_0, p_0)$ from $(Q_0, P_0)$
        Generate trajectory $(q_0, p_0, \dots, q_N, p_N)$ from (7), (8)
        if $q_N \in Q_N$ then
            Compute $(G_1(q_0, q_N), \dots, G_1(q_{N-1}, q_N)$ with (13)
            $\mathcal{D} \leftarrow \{(q_0, p_0, q_N, p_N, G(q_0, q_N)), \dots\}$
        end if
    until $|\mathcal{D}| = n_{\mathrm{data}}$
    Train a neural network $G_{nn}(q, p)$ with $\mathcal{D}$
end procedure
procedure $u$=CONTROL($q_k, q_N, G_{nn}, \alpha$)
    Set $u' = 0$
    repeat
        Set $u = u'$
        Compute $u' = U(q_k, D_1 G_{nn}(f(q_k, u), q_N))$
        Set $u' = (1 - \alpha)u + \alpha u'$
    until $\|u' - u\| < \epsilon$
end procedure
```

As such, the generating function yields the solution to the optimal control as summarized below.

Proposition 3: [8] Consider the optimal control problem of (1) and (2). Let G_1 be the generating function satisfying the Hamilton-Jacobi equations (11), respectively for given (6), (7), and (8). Then, the optimal control is determined by

$$u_k = U(q_k, D_1 G_1(f(q_k, u_k), q_N)). \tag{12}$$

Furthermore, the optimal cost-to-go function to transfer the state q_k at t_k to q_{k+j} at t_{k+j} is given by

$$J(q_k, q_{k+j}) = \sum_{i=k}^{k+j-1} L(q_i, u_i) = -G_1(q_k, q_{k+l}). \tag{13}$$

III. LEARNING GENERATING FUNCTION

According to the Hamilton-Jacobi theory, the optimal feedback control can be constructed once the generating function is obtained. The previous approaches include approximating it with a Taylor series expansion or using a quadratic form for the linear dynamics. As it is relatively straightforward to generate sample optimal trajectories from (7) and (8) and to compute the value of the generating function from (13), it is reasonable to utilize deep neural network to model the generating function. This is referred to as G-learning.

A. G-Learning

The training procedure of the G-learning is summarized in Table I. The objective is to generate the set of optimal trajectories and the value of the generating function to be used for training and tests of deep learning. We choose the set of initial states $Q_0 \subset \mathsf{Q}$ and the set of terminal states $Q_N \subset \mathsf{Q}$, in which the controlled system would operate. We further choose $P_0 \subset \mathsf{T}^*\mathsf{Q}$ from which the initial co-state is sampled. Then, we propagate the initial condition (q_0, p_0) along the Hamilton's equations (7) and (8). If the terminal state belongs to the desired operating range represented by Q_N, the corresponding value of the generating function is computed by (13) and they are saved in the data set $\mathcal{D}$. This is repeated until the desired number of data n_{data} is reached.

Then, a neural network $G_{nn} : \mathbb{R} \times \mathsf{Q} \times \mathsf{Q} \to \mathbb{R}$ that takes the time step k and (q_k, q_N) as the input is trained with the following loss function:

$$\begin{aligned}\mathcal{L} = \sum_{\mathcal{D}} &\|G_1(q_k, q_N) - G_{nn}(q_k, q_N)\|^2 \\ &+ c_1 \|p_k - D_1 G_{nn}(q_k, q_N)\|^2 + c_2 \|p_N + D_2 G_{nn}(q_k, q_N)\|^2,\end{aligned}$$

for weighting parameters $c_1, c_2 > 0$. In other words, it is trained to satisfy the value of G_1 and its derivatives. Instead of the L_2 norm presented above, any other loss function can be utilized.

Once G_{nn} is constructed, the optimal control can be constructed via (12) for the given current state q_k and the desired terminal state q_N. Here, we have to solve the implicit equation for (12) for u_k. Depending on the specific structure of U or f, (12) may have an explicit solution. Otherwise, (12) is naturally written as a form of fixed-point iteration. The procedure to solve (12) for u_k with fixed-point iteration is presented in the second part of Table I. One can show that if f is Lipschitz continuous and the gradients of U and G_{nn} are bounded, the presented iteration yields a contraction for a sufficiently small α, and therefore, it converges to the unique solution.

Compared with the policy gradient in reinforcement learning, one distinct feature of G-Learning is that the optimal control is given as a function of the current state q_k and the terminal state q_N. Consequently, it provides optimal feedback controls for varying terminal state q_N. Also, the terminal boundary condition does not have to be enforced through an additional penalty at the objective function. Another benefit is that the iteration is to satisfy the terminal boundary condition, while the optimality is naturally enforced by (7) and (8). As such, this is contrast to the policy gradient, where the optimality is achieved gradually and approximately. Further, other kinds of the terminal boundary conditions or constraints can be addressed by utilizing different types of generating function [8].

B. Equivariant G-Learning

The presented G-Learning is to approximate the generating function on the domain of $\mathbb{R} \times \mathsf{Q} \times \mathsf{Q}$. In this section, we show that the domain of the learning can be reduced if there exits a symmetry in the dynamics and the cost function, thereby improving the sampling efficiency of the G-Learning.

Suppose that there is a Lie group G acting on the configuration manifold Q by its left action $\Phi : \mathsf{G} \times \mathsf{Q} \to \mathsf{Q}$. For a given $g \in \mathsf{G}$, we denote $\Phi(g, q) = \Phi_g(q) = g \circ q$. The group action is *lifted* into the cotangent bundle $\mathsf{T}^*\mathsf{Q}$ by its cotangent lift $\mathsf{T}^*_{gq}\Phi_{g^{-1}} : \mathsf{G} \times \mathsf{T}^*_q\mathsf{Q} \to \mathsf{T}^*_{gq}\mathsf{Q}$. In the subsequent development, we assume that the group action can be represented by a left matrix multiplication for simplicity. This results in $g \circ q = gq$ and $g \circ p = \mathsf{T}^*_{gq}\Phi_{g^{-1}}(p) = g^{-T}p$. Further, it is considered that for each $g \in \mathsf{G}$, there is the corresponding group action on the control input. This is denoted by the same symbol after slight abuse of notation, i.e., $g \circ u = gu$.

Suppose that the discrete dynamic is g-equivariant, or $g \circ f = f \circ g$. This implies

$$gq_{k+1} = f(gq_k, gu_k), \tag{14}$$

for any (s_{k+1}, s_k, u_k) satisfying (1). It follows that if $(q_0, u_0, q_1, u_1, \cdots)$ is a trajectory of the state and the control following the discrete equations of motion, the transformed pair of state and control $(gq_0, gu_0, gq_1, gu_2, \cdots)$ is another trajectory.

Next, assume that the running cost L is g-invariant, i.e., $L \circ g = L$, or

$$L(gq, gu) = L(g, u). \tag{15}$$

Then, the resulting optimal trajectories and the generating function exhibit invariance or equivariance as summarized below.

Proposition 4: Consider the optimal control problem of (1) and (2), and the group action $g \in \mathsf{G}$. Suppose f is g-equivariant and L is g-invariant as presented in (14), and (15). Then, the following properties hold:

(i) The Hamiltonian (3) and the optimal Hamiltonian (6) are g-invariant, i.e., $H \circ g = H$.

(ii) The optimal control of (4) is g-equivariant, i.e., $u \circ g = g \circ u$.

(iii) The discrete Hamilton's equations (7) and (8) are g-equivariant, i.e., $D_i H \circ g = g \circ D_i H$ for $i \in \{1, 2\}$.

(iv) The generating function is g-invariant, i.e., $G_1 \circ g = G_1$.

Proof: Due to the page limit, the proof is omitted. ■

The property (iii) implies that if $(q_0, p_1, q_1, p_2, \ldots)$ is an optimal trajectory satisfying (7) and (8) for a given boundary condition (q_0, q_N), then $g \circ (q_0, p_1, q_1, p_2, \ldots) = (gq_0, g^{-T}p_1, gq_1, g^{-T}p_2, \ldots)$ is another optimal trajectory with the transformed boundary condition (gq_0, gq_N). In other words, the set of optimal trajectories is closed under the group action g, and this provides an equivalent relation $\sim$ for the optimal trajectories.

Also, as the generating function is invariant under g from the property (iv), it can be formulated on the reduced space of $(\mathsf{Q} \times \mathsf{Q})/\mathsf{G}$. For example, G_1 is evaluated by

$$G_1(q_0, q_N) = \tilde{G}_1([q_0, q_N]), \tag{16}$$

where $[q_0, q_N] = \{(gq_0, gq_N) \in \mathsf{Q} \times \mathsf{Q} \,|\, g \in \mathsf{G}\}$ is the equivalent class, and $\tilde{G}_1$ is a *reduced* form of the generating function. There are several options to formulated the reduced generating function, $\tilde{G}_1$:

- Perhaps most naively, define $\tilde{G}_1$ by a neural network on $\mathsf{Q} \times \mathsf{Q}$, but provide additional training data from the equivalent class;
- Formulate $\tilde{G}_1$ with an invariant neural network on $\mathsf{Q} \times \mathsf{Q}$ such that the g-invariance is satisfied inherently;
- Identify a canonical projection (or surjection) $\phi(q_0, q_N) = [q_0, q_N]$ to the equivalent class, and let $G_1 = \tilde{G}_1 \circ \phi$, where $\tilde{G}_1$ is formulated by a neural network on $(\mathsf{Q} \times \mathsf{Q})/\mathsf{G}$.

Depending on the particular configuration space and the Lie group considered, one of these method can be selected.

Additionally, in case the group G acts on Q transitively, the third option can be simplified as follows. The transitivity implies that for any $q_N, \tilde{q}_N \in \mathsf{Q}$, there exists a group element $\tilde{g} \in \mathsf{G}$ such that $q_N = \tilde{g}\tilde{q}_N$. To utilize this property, we train a reduced generating function $\tilde{G}_1 : \mathbb{R} \times \mathsf{Q} \to \mathbb{R}$, which corresponds to the generating function with the *fixed* terminal state $\tilde{q}_N$:

$$\tilde{G}_1(\tilde{q}_k) \triangleq G_1(\tilde{q}_k, \tilde{q}_N). \tag{17}$$

The corresponding optimal feedback control is

$$\tilde{u}_k = U(\tilde{q}_k, D\tilde{G}_1(f(\tilde{q}_k, \tilde{u}_k))), \tag{18}$$

which yields $\tilde{u}_k = \tilde{u}_k(\tilde{q}_k)$ to steer the controlled trajectory to the fixed $\tilde{q}_N$. In other words, this generates the optimal trajectory $(\tilde{q}_k, \tilde{p}_{k+1}, \ldots, \tilde{q}_N)$ from any varying $\tilde{q}_k$ to the fixed $\tilde{q}_N$ according to Proposition 3. The choice of $\tilde{q}_N$ is arbitrary, and for example, it can be chosen as the origin of Q for convenience.

Now, this optimal control for the fixed terminal state $\tilde{q}_N$ can be transformed into the optimal control from any current state q_k to any desired state q_N. For given q_k and q_N, choose $\tilde{g}$ such that

$$q_N = \tilde{g}\tilde{q}_N, \tag{19}$$

and let $\tilde{q}_k = \tilde{q}^{-1}q_k$. Then, the optimal trajectory from $\tilde{q}_k = \tilde{g}^{-1}q_k$ to $\tilde{q}_N$ can be generated by (18) to obtain $(\tilde{q}_k, \tilde{p}_{k+1}, \ldots, \tilde{q}_N)$. This is transformed by the group action $\tilde{g}$ into

$$\tilde{g} \circ (\tilde{q}_k, \tilde{p}_{k+1}, \ldots, \tilde{q}_N) = (q_k, p_{k+1}, \ldots, q_N),$$

which corresponds to the desired optimal trajectory from the current state q_k and the desired terminal state q_N. According to the property (ii) of Proposition 4, the optimal control u is obtained by $u = g\tilde{u}$. These procedures are summarized at Table II. As the neural network is trained on the domain of $\mathbb{R} \times \mathsf{Q}$ instead of $\mathbb{R} \times \mathsf{Q} \times \mathsf{Q}$, the data collection and the training of neural networks can be completed substantially more efficiently.

IV. Numerical Example

A. Dubin's Vehicle

We consider a variable-speed Dubin's vehicle, which is a kinematic model for a two-dimensional non-holonomic vehicle whose turning rate and speed can be controlled directly. The discrete state equations are given by

$$x_{k+1} = x_k + hv_k \cos\theta_k, \tag{20}$$

$$y_{k+1} = y_k + hv_k \sin\theta_k, \tag{21}$$

$$\theta_{k+1} = \theta_k + hw_k, \tag{22}$$

where the state $q = (x, y, \theta) \in \mathsf{Q} = \mathbb{R}^2 \times \mathsf{S}^1$ corresponds to the location of the vehicle in the two-dimensional plane and the heading angle. The control input $u = (v, w) \in \mathbb{R}^2$ is composed of the linear velocity and the angular velocity, and the constant $h > 0$ represents the step size. This is rearranged into a matrix form as

$$q_{k+1} = f(q_k, u_k) = q_k + B(q_k)u_k, \tag{23}$$

TABLE II
PROCEDURES OF EQUIVARIANT G-LEARNING

```
procedure $\tilde{G}_{nn}$ = TRAINING($\tilde{q}_N, Q_0, P_N, n_{\mathrm{data}}$)
    Set $\mathcal{D} = \{\}$
    repeat
        Sample $(\tilde{p}_N)$ from $P_N$
        Generate trajectory $(\tilde{q}_0, \tilde{p}_0, \ldots, \tilde{q}_N, \tilde{p}_N)$ by propagating
    $(\tilde{q}_N, \tilde{p}_N)$ backward with (7), (8)
        if $\tilde{q}_0 \in Q_0$ then
            Compute $(G_1(\tilde{q}_0, \tilde{q}_N), \ldots, G_1(\tilde{q}_{N-1}, \tilde{q}_N)$ with (13)
            $\mathcal{D} \leftarrow \{(\tilde{q}_0, \tilde{p}_0, \tilde{q}_N, \tilde{p}_N, G(\tilde{q}_0, \tilde{q}_N)), \ldots\}$
        end if
    until $|\mathcal{D}| = n_{\mathrm{data}}$
    Train a neural network $\tilde{G}_{nn}(\tilde{q})$ with $\mathcal{D}$
end procedure
procedure $u$=CONTROL($\tilde{q}_N, q_k, q_N, \tilde{G}_{nn}, \alpha$)
    Find $\tilde{g} \in \mathsf{G}$ such that $q_N = \tilde{g}\tilde{q}_N$
    Set $\tilde{q}_k = \tilde{g}^{-1} q_k$
    Set $\tilde{u}' = 0$
    repeat
        Set $\tilde{u} = \tilde{u}'$
        Compute $\tilde{u}' = U(\tilde{q}_k, D\tilde{G}_{nn}(f(\tilde{q}_k, \tilde{u})))$
        Set $\tilde{u}' = (1-\alpha)\tilde{u} + \alpha\tilde{u}'$
    until $\|\tilde{u}' - \tilde{u}\| < \epsilon$
    Set $u = g\tilde{u}$
end procedure
```

where $B(q_k) \in \mathbb{R}^{3\times 2}$ is

$$B(q_k) = h \begin{bmatrix} \cos\theta_k & 0 \\ \sin\theta_k & 0 \\ 0 & 1 \end{bmatrix}. \tag{24}$$

We consider a minimum control to transfer the vehicle from a given initial state q_0 to a terminal state q_N over a fixed time step N, while minimizing the sum of the following running cost:

$$L(u_k) = \frac{1}{2} u_k^T R u_k, \tag{25}$$

for a positive-definite and symmetric matrix $R \in \mathbb{R}^{2\times 2}$.

B. Necessary Conditions for Optimality

From (3), the Hamiltonian is given by

$$H(q_k, p_{k+1}, u_k) = -\frac{1}{2} u_k^T R u_k + p_{k+1}^T (q_k + B(q_k) u_k), \tag{26}$$

The control input that extremizes the Hamiltonian is obtained by solving $\frac{\partial H}{\partial u} = 0$ for u as follows.

$$u_k = R^{-1} B^T(q_k) p_{k+1}. \tag{27}$$

Substituting this back into (26), the optimal Hamiltonian of (6) is given by

$$H(q_k, p_{k+1}) = \frac{1}{2} p_{k+1}^T B(q_k) R^{-1} B^T(q_k) p_{k+1} + p_{k+1}^T q_k. \tag{28}$$

Next, we derive Hamilton's equations. For any $x \in \mathbb{R}^3$,

$$\frac{\partial (B(q)^T x)}{\partial q} = C^T(q) x e_3^T,$$

where $e_3 = [0, 0, 1]^T \in \mathbb{R}^3$ and the matrix $C(q) \in \mathbb{R}^{2\times 3}$ is

$$C(q) = h \begin{bmatrix} -\sin\theta & 0 \\ \cos\theta & 0 \\ 0 & 0 \end{bmatrix}.$$

Using this and from (7) and (8), Hamilton's equations are

$$q_{k+1} = q_k + B(q_k) R^{-1} B^T(q_k) p_{k+1}, \tag{29}$$
$$p_k = p_{k+1} + (p_{k+1}^T C(q_k) R^{-1} B^T(q_k) p_{k+1}) e_3. \tag{30}$$

Due to the factor e_3 of (30), the first two elements of $p = [p_x, p_y, p_\theta]$ remain fixed, i.e.,

$$p_{x_k} = p_{x_{k+1}}, \quad p_{y_k} = p_{y_{k+1}}. \tag{31}$$

As such, we drop the subscript for the time step in p_x and p_y in the subsequent development. Suppose R is diagonal such that $R = \mathrm{diag}[r_1, r_2]$ for $r_1, r_2 > 0$. Then, the third element of (30) is written as

$$p_{\theta_k} = p_{\theta_{k+1}} + \frac{1}{2r_1}((p_y^2 - p_x^2)\sin 2\theta_k + 2 p_x p_y \cos 2\theta_k). \tag{32}$$

In short, the Hamilton's equations describing the evolution of the optimal (q, p) are given by (29), (31), and (32).

C. Equivariarnce

Consider a group action $g \in \mathsf{SE}(2)$ on $q = (x, y, \theta) \in \mathsf{Q}$, parameterized by $(\Delta x, \Delta y, \Delta\theta) \in \mathbb{R}^2 \times [-\pi, \pi)$ as

$$g(\Delta x, \Delta y, \Delta\theta) q = (R(\Delta\theta)\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \theta + \Delta\theta), \tag{33}$$

where $R(\theta) \in \mathsf{SO}(2)$ is

$$R(\Delta\theta) = \begin{bmatrix} \cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta \end{bmatrix}. \tag{34}$$

This corresponds to rotating the vehicle about the origin by $\Delta\theta$ and translating it by $(\Delta x, \Delta y)$. The corresponding group action on the control is the identity map, i.e., $g \circ u = u$.

Since (33) is not written as the form of matrix multiplication, the cotagent lift cannot be simplify obtained by g^{-T} as discussed in III-B. Instead, from the geometric formulation of the cotangent lift (see, for example, [11]), we can show that

$$\mathsf{T}^*_{gq}\Phi_{g^{-1}} \cdot p = (R(\Delta\theta)\begin{bmatrix} p_x \\ p_y \end{bmatrix}, p_\theta), \tag{35}$$

for $p = (p_x, p_y, p_\theta) \in \mathsf{T}^*_q\mathsf{Q}$, i.e., in the preceding development, the operation denoted by $g^{-T}p$ can be replaced by (35).

The resulting equivariance properties of the Dubin's vehicle are summarized as follows.

Proposition 5: Consider the Dubin's vehicle dynamics, given by (23) and the group action of (33) and (35). Then, the following properties hold:

(i) The state equation f is g-equivariant, and the running cost is g-invariant, i.e., $f \circ g = g \circ f$ and $L \circ g = L$.
(ii) The optimal control of (27) is g-invariant, i.e., $u \circ g = u$

(iii) The modified Hamiltonian $H'(q_k, p_{k+1}) = H(q_k, p_{k+1}) - p_{k+1}^T q_k$ is g-invariant, i.e., $H' \circ g = H'$.

(iv) The discrete Hamilton's equations (29), (31), and (32) are g-equivariant.

(v) The generating function is g-invariant, i.e., $G_1 \circ g = G_1$.

Proof: Due to the page limit, the proof is omitted. ■

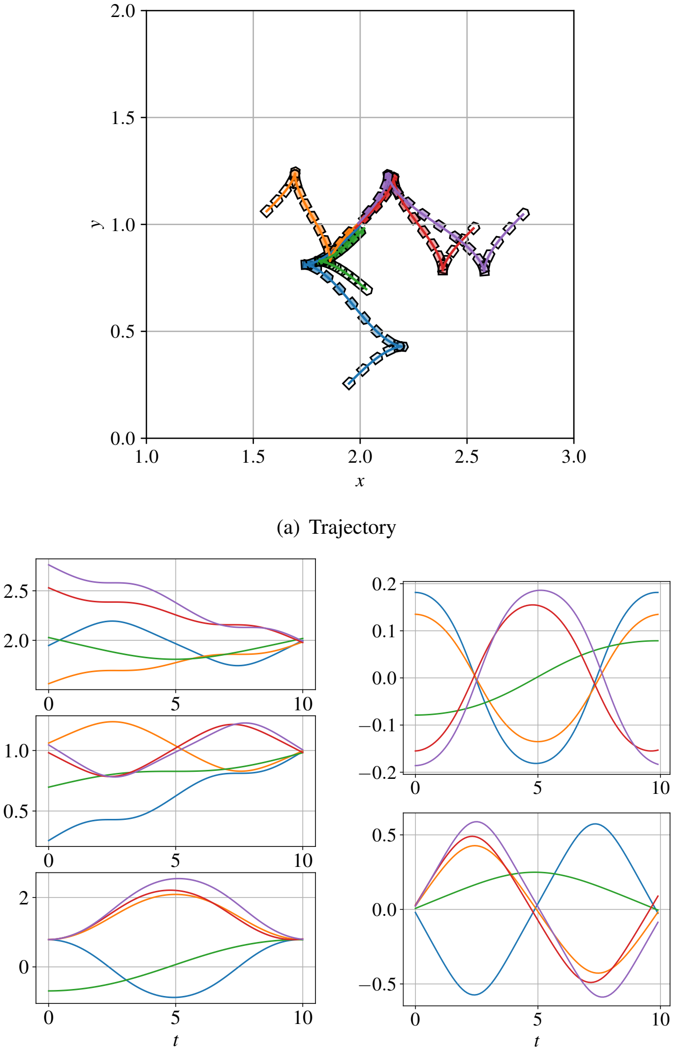
While we cannot directly apply Proposition 4 due to the group action represented by (33) and (35), the equivariance of the Hamilton's equations and the invariance of the generating function hold as indicated by (iv) and (v). As such, the equivariant G-learning can be utilized.

D. Numerical Example

We choose $R = \text{diag}[1, 10]$, $h = 0.1$, and $N = 100$. The fixed terminal state is $\tilde{q}_N = [0, 0, 0]$. The neural network $\tilde{G}_{nn}$ is modeled as a multi-layer perceptron with two hidden layers of the size 32, and the soft L_1 activation function. It is trained with $n_{data} = 10000$ over 6000 epochs with the batch size of 50 and the learning rate of 10^{-4}. These are implemented by pytorch. The numerical results to steer the vehicle into $q_N = (2, 1\frac{\pi}{4})$ from five initial conditions are presented at Figure 1.

E. Discussion

These illustrate the feasibility of the proposed G-learning in data-driven indirect optimal control. One particular point for improvements is that the generating function is trained in prior for a preselected samples. As such, the performance may degrade if the actual states encountered during the implementation stage are not well reflected by the training data. In fact, this is a well understood issue in the behavior cloning of imitation learning: if there is a distribution mismatch between the training data and the test data, the pretrained network performs poorly, and to address it, several techniques, such as [12], have been presented to update the training data distribution. The presented numerical examples exhibits the error in the terminal state at the level of $0.01 \sim 0.03$ in two norm, potentially due to the above issue. The presented G-learning algorithm can be improved by updating the generating function online, and it is currently being investigated.



(a) Trajectory

(b) $q = (q_x, q_y, q_\theta)$

(c) $u = (v, w)$

Fig. 1. Optimal trajectories from random initial states to $q_N = (2, 1, \pi/4)$

REFERENCES

[1] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[3] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[4] S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko, "Regularizing action policies for smooth control with reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1810–1816.

[5] V. M. Guibout and D. J. Scheeres, "Solving relative two-point boundary value problems: Spacecraft formulation flight transfers application," *Journal of guidance, control, and dynamics*, vol. 27, no. 4, pp. 693–704, 2004.

[6] C. Park, V. Guibout, and D. J. Scheeres, "Solving optimal continuous thrust rendezvous problems with generating functions," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 321–331, 2006.

[7] C. Park and D. J. Scheeres, "Determination of optimal feedback terminal controllers for general boundary conditions using generating functions," *Automatica*, vol. 42, no. 5, pp. 869–875, 2006.

[8] T. Lee, "Optimal control of partitioned hybrid systems via discrete-time Hamilton-Jacobi theory," *Automatica*, vol. 50, no. 8, pp. 2062–2069, Aug. 2014.

[9] T. Ohsawa, A. M. Bloch, and M. Leok, "Discrete Hamilton–Jacobi theory," *SIAM Journal on Control and Optimization*, vol. 49, no. 4, pp. 1829–1856, 2011.

[10] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[11] D. D. Holm, T. Schmah, and C. Stoica, *Geometric mechanics and symmetry: from finite to infinite dimensions*. Oxford University Press, 2009, vol. 12.

[12] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.