2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Maximum Entropy Differential Dynamic Programming

Oswin So, Ziyi Wang and Evangelos A. Theodorou

***Abstract*— In this paper, we present a novel maximum entropy formulation of the Differential Dynamic Programming algorithm and derive two variants using unimodal and multimodal value functions parameterizations. By combining the maximum entropy Bellman equations with a particular approximation of the cost function, we are able to obtain a new formulation of Differential Dynamic Programming which is able to escape from local minima via exploration with a multimodal policy. To demonstrate the efficacy of the proposed algorithm, we provide experimental results using four systems on tasks that are represented by cost functions with multiple local minima and compare them against vanilla Differential Dynamic Programming. Furthermore, we discuss connections with previous work on the linearly solvable stochastic control framework and its extensions in relation to compositionality. Link to Video.**

I. Introduction

Existing methods for trajectory-optimization solve the optimization problem by iteratively relying on local information via derivatives [1, 2]. Differential Dynamic Programming (DDP) [3, 4] is a popular trajectory optimization method for nonlinear systems used in model-based Reinforcement Learning (RL) and Optimal Control problems, where the problem is iteratively solved via second order approximations of the cost and dynamics. With stage-wise positive Hessian matrices, DDP enjoys quadratic convergence [5]. However, these methods usually only guarantee convergence to a local minimum and are unable to reach better local minima once converged. In cases where there are dynamic obstacles, the cost landscape can be highly nonconvex with suboptimal local minima that are unsatisfactory. Methods that try to address this problem include random restarts [6] or via topological approaches that explicitly consider the homotopy classes of trajectories [7, 8].

Maximum entropy is a technique widely used in RL and Stochastic Optimal Control (SOC) to improve the robustness of stochastic policies. Performance robustness is achieved through an additional entropy regularization term in the cost function that improves exploration by discouraging policies from converging to a delta distribution over the current optimal control [9, 10]. In RL, Soft Actor Critic uses a maximum entropy objective and is considered to be state of the art for off-policy methods [11]. In SOC, the Information Theoretic Model Predictive Path Integral (IT-MPPI) algorithm [12, 13] is a generalization of this technique which uses the forward Kullback-Leibler (KL) divergence between the controlled distribution and a prior distribution for regularization. The form of maximum entropy is recovered when the uniform distribution is used as the prior distribution. In [14, 15], the Tsallis divergence, a generalization of the KL divergence, is used as a regularization term in the objective, leading to further robustness improvements.

In this paper, we consider discrete time deterministic optimal control problems and take a *relaxed control* approach with entropy regularization similar to [16]. We propose two novel variations of DDP under the Maximum Entropy Optimal Control (MEOC) formulation using unimodal and multimodal Gaussian policies. Finally, we compare the performance of both proposed algorithms against vanilla DDP on 2D Point Mass, 2D Car, Quadcopter and Manipulator in simulation.

The main contributions of this work are threefold:

- We derive the Bellman equation for the discrete time MEOC problem.
- We propose Maximum Entropy DDP (ME-DDP) and Multimodal Maximum Entropy DDP (MME-DDP) to improve exploration over vanilla DDP.
- We showcase the benefit of the improved exploration of ME-DDP and MME-DDP over vanilla DDP in converging to better local minima on four different systems in simulation.

II. Maximum Entropy Bellman Equation

Standard discrete-time deterministic optimal control problems minimize the cost over time horizon $(0, 1, \cdots, T)$

$$J(u) \coloneqq \Phi(x_T) + \sum_{t=0}^{T-1} l_t(x_t, u_t), \tag{1}$$

where l_t and Φ are the running and terminal costs respectively. The state and control trajectories, $(x_t)_{t=0,\cdots,T}, x_t \in \mathbb{R}^{n_x}$ and $(u_t)_{t=0,\cdots,T-1}, u_t \in \mathbb{R}^{n_u}$, satisfy deterministic dynamics

$$x_{t+1} = f(x_t, u_t). \tag{2}$$

In this work, we take a *relaxed control* approach and consider a stochastic control policy $\pi_t(u_t|x_t)$ with the same deterministic dynamics as in (2). In addition, we introduce an entropy term to the original objective (1)

$$J_\pi \coloneqq \mathbb{E}_\pi\left[\Phi(x_T) + \sum_{t=0}^{T-1}\Big(l_t(x_t, u_t) - \alpha H\big[\pi_t\big]\Big)\right], \tag{3}$$

where $\alpha > 0$ is an inverse temperature term, the expectation is taken with respect to $u \sim \pi(\cdot|x)$, and $H[\pi]$ is the Shannon entropy of π defined as

$$H[\pi] = -\mathbb{E}_\pi[\ln \pi] = -\int \pi(u) \ln \pi(u)\, \mathrm{d}u. \tag{4}$$

The authors are with the Autonomous Control and Decision Systems Laboratory, Georgia Institute of Technology, Atlanta, GA, USA. Email correspondance to: oswinso@gatech.edu

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9812228

For this problem formulation and the standard value function definition of $V(x) = \min_\pi J(x,\pi)$, the Bellman equation takes the form

$$V(x) = \inf_\pi \Big\{ \mathbb{E}_\pi \big[l(x,u) + V'(f(x,u)) \big] - \alpha H\big[\pi\big] \Big\}, \quad (5)$$

In (5) and below we omit the time index t for nonterminal times for simplicity and use $V'(f(x,u))$ to denote the value function at the next timestep.

Solving (5) results in a Gibbs distribution for π^* [14, 16]. The form of π^* and V are presented in the following lemma.

Lemma 1. *The optimal policy π^* solving the infimum in (5) is the Gibbs distribution*

$$\pi^*(u|x) = Z^{-1} \exp\left(-\frac{1}{\alpha} \Big[V'(f(x,u)) + l(x,u) \Big] \right), \quad (6)$$

where Z denotes the partition function

$$Z(x) = \int \exp\left(-\frac{1}{\alpha} \Big[V'(f(x,u)) + l(x,u) \Big] \right) \mathrm{d}u. \quad (7)$$

Consequently, the value function V takes the form

$$V(x) = -\alpha \ln Z(x). \quad (8)$$

We refer the readers to Appendix A in [17] for a proof of Lemma 1.

III. Maximum Entropy DDP

We will now use DDP to solve the MEOC problem and derive the ME-DDP algorithm. For notational simplicity, we will drop the second-order approximation of the dynamics as in iterative Linear Quadratic Regulator (iLQR) in our description of DDP. The dropped second-order dynamics terms can easily be added back in the derivations below. We refer readers to [3, 18] for a detailed overview of the vanilla DDP and iLQR algorithms.

The DDP algorithm consists of a forward pass and a backward pass. The forward pass simulates the dynamics forward in time obtaining a set of nominal state and control trajectories $(\bar{x}_{0:T}, \bar{u}_{0:T-1})$, while the backward pass solves the Bellman equation with a 2nd order approximation of the costs and dynamics equations around the nominal trajectories. The boundary conditions for the value function V are obtained by performing a 2nd order Taylor expansion of the terminal cost Φ:

$$V_{xx,T} = \Phi_{xx}, \quad V_{x,T} = \Phi_x, \quad V_T = = \Phi. \quad (9)$$

To derive the backward pass, we first perform a quadratic approximation of the cost function around $(\bar{x}, \bar{u})$

$$l(x,u) \approx l(\bar{x},\bar{u}) + \begin{bmatrix} l_x \\ l_u \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} l_{xx} & l_{xu} \\ l_{ux} & l_{uu} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix},$$

where $\delta x := x - \bar{x}$, $\delta u := u - \bar{u}$. We also perform a linear approximation of the dynamics f:

$$f(x,u) \approx f(\bar{x},\bar{u}) + f_x^{\mathrm{T}} \delta x + f_u^{\mathrm{T}} \delta u.$$

Define $Q := V'(f(x,u)) + l(x,u)$, with subscripts denoting partial derivatives. The next lemma describes the optimal policy and value function.

Lemma 2. *The optimal policy for the approximated problem is Gaussian with mean δu^* and covariance αQ_{uu}^{-1}*

$$\pi^*(\delta u|\delta x) = \mathcal{N}(u; \delta u^*, \alpha Q_{uu}^{-1}) \quad (10)$$

where the mean δu^* has the same form as in vanilla DDP

$$\delta u^* = -Q_{uu}^{-1}\Big(Q_{ux}\delta x + Q_u \Big) = K\delta x + k. \quad (11)$$

Consequently, the value function has the form

$$V(x) = \bar{V}(x) + V_H(\bar{x}) + V_x(\bar{x})^{\mathrm{T}}\delta x + \frac{1}{2}\delta x^{\mathrm{T}} V_{xx}(\bar{x}) \delta x, \quad (12)$$

where

$$\bar{V}(\bar{x}) = \bar{V}'(\bar{x}) + l(\bar{x},\bar{u}) - \frac{1}{2} Q_u^{\mathrm{T}} Q_{uu} Q_u, \quad (13)$$

$$V_H(\bar{x}) = \frac{1}{2}\Big(\ln|Q_{uu}| - n_u \ln(2\pi\alpha) \Big), \quad (14)$$

$$V_x(\bar{x}) = Q_x + K^{\mathrm{T}} Q_{uu} k + K^{\mathrm{T}} Q_u + Q_{ux}^{\mathrm{T}} k, \quad (15)$$

$$V_{xx}(\bar{x}) = Q_{xx} + K^{\mathrm{T}} Q_{uu} K + K^{\mathrm{T}} Q_{ux} + Q_{ux}^{\mathrm{T}} K. \quad (16)$$

Remark. *Note that the update rules for $\bar{V}, V_x$ and V_{xx} are exactly the same as in vanilla DDP. The only difference here is the addition of the V_H term due to the additional entropy regularization. Consequently, this term can be ignored if the $V(x)$ is not needed. This term disappears as $\alpha \to 0$ when the problem reverts to the vanilla case.*

We refer the readers to Appendix B in [17] for a proof of Lemma 2.

IV. Multimodal Maximum Entropy DDP

While a unimodal Gaussian policy is able to achieve better exploration compared to the deterministic policy, it is often the case that multiple modes need to be explored simultaneously to converge to the global minimum [10]. In this section, we derive a multimodal extension to the ME-DDP introduced.

Let $\{\bar{x}^{(n)}, \bar{u}^{(n)}\}_{n=1}^N$ denote N different nominal state and control trajectories, and let $\Phi^{(n)}$ corresponds to the respective quadratic approximation of Φ around $\bar{x}^{(n)}$ and $\bar{u}^{(n)}$. Instead of using a single quadratic approximation of Φ for the terminal cost as in (9), we use the combined approximation $\tilde{\Phi}$

$$V_T(x) = \tilde{\Phi}(x) := -\alpha \ln \sum_{n=1}^N \exp\left(-\frac{1}{\alpha} \Phi^{(n)}(x) \right). \quad (17)$$

The log-sum-exp is a smoothed combination of the local quadratic approximation and approaches $\min_n\{\Phi^{(n)}\}$ as $\alpha \to 0$ as shown in Fig. 1.

The above equation becomes easier to work with when considering the *exponential transform* of the problem. Define $\mathcal{E}_\alpha$ to be the following function

$$\mathcal{E}_\alpha(y) := \exp\Big(-\frac{1}{\alpha} y \Big). \quad (18)$$

We now define the *reward* r, R_T and *desirability* z as

$$r_t := \mathcal{E}_\alpha(l_t), \quad r_T := \mathcal{E}_\alpha(\Phi), \quad z := \mathcal{E}_\alpha(V(x)). \quad (19)$$

With this transformation, note that the *desirability* function z is exactly the partition function Z from (8) and is linear in both z' and r (denoting z' for the next timestep):

$$\begin{cases} z(x) = Z(x) = \int z'(f(x,u))\, r(x,u)\, \mathrm{d}u, & \text{(20a)} \\ z_T(x_T) = r_T(x_T). & \text{(20b)} \end{cases}$$

With this transformation, the optimal policy in (6) has the following elegant form

$$\pi(u|x) = z(x)^{-1} z'\big(f(x,u)\big)\, r(x,u). \tag{21}$$

Additionally, using the desirability function z to write (17), we see that z has an additive structure:

$$z_T(x_T) = \sum_{n=1}^{N} z_T^{(n)}(x_T), \quad z_T^{(n)}(x_T) := r^{(n)}(x_T). \tag{22}$$

The following lemma now shows that this structure holds for all time.

Lemma 3. *Suppose that the terminal cost has the form (17). Then, for all $t = 0, \dots, T$, the desirability function z has the following additive structure*

$$z(x) = \sum_{n=1}^{N} z^{(n)}(x), \quad z^{(n)} := \int z'^{(n)}(f(x,u))\, r(x,u)\, \mathrm{d}u,$$

Proof. This holds at the terminal time from (22). Suppose that $z'(x) = \sum_{n=1}^{N} z'^{(n)}(x)$. Substituting this in (20a), we get

$$\begin{aligned} z(x) &= \int \left(\sum_{n=1}^{N} z'^{(n)}(f(x,u)) \right) r(x,u)\, \mathrm{d}u, \\ &= \sum_{n=1}^{N} \int z'^{(n)}(f(x,u))\, r(x,u)\, \mathrm{d}u, \\ &= \sum_{n=1}^{N} z^{(n)}(x). \end{aligned} \tag{23}$$

By induction, this holds for all time. □

Remark. *Note that the form of $z^{(n)}$ is identical to the Bellman equation (20a). In other words, each $z^{(n)}$ and $V^{(n)}$ is computed in exactly the same way as z and V in the unimodal case but with a different terminal condition $\Phi^{(n)}$.*

Substituting V back for z in (23) yields

$$V(x) = -\alpha \ln \sum_{n=1}^{N} \exp\left(-\frac{1}{\alpha} V^{(n)}(x) \right). \tag{24}$$

This result makes sense intuitively—the combined value function should be related to the minimum of the individual approximated value functions resulting from the different nominal states.

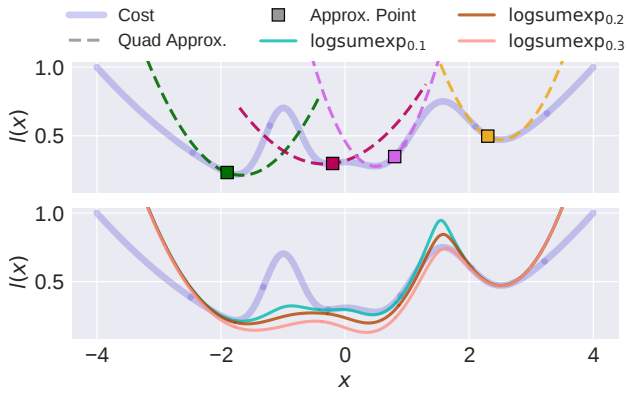


Fig. 1: Comparison of the individual quadratic approximation (top) and the log-sum-exp approximation (bottom) of the cost function $l(x)$ with varying choices of inverse temperature α. Higher α leads to smoother approximation.

Using the linearity of the desirability function (23), the optimal policy (21) has the form

$$\begin{aligned} \pi(u|x) &= z(x)^{-1} \left(\sum_{n=1}^{N} z'^{(n)}(f(x,u)) \right) r(x,u), \\ &= \sum_{n=1}^{N} \frac{z^{(n)}(x)}{z(x)} z^{(n)}(x)^{-1} z'^{(n)}(f(x,u))\, r(x,u), \\ &= \sum_{n=1}^{N} w^{(n)}(x) \pi^{(n)}(u|x), \end{aligned} \tag{25}$$

where

$$\begin{aligned} \pi^{(n)}(u|x) &:= z^{(n)}(x)^{-1} z'^{(n)}(f(x,u))\, r(x,u), \\ w^{(n)}(x) &:= z(x)^{-1} z^{(n)}(x), \quad \sum_{n=1}^{N} w^{(n)} = 1. \end{aligned}$$

This is exactly the policy obtained in the normal case, except that we consider $z^{(n)}$ instead of z. Since $V^{(n)}$ is quadratic in the state, each $\pi^{(n)}$ will be Gaussian as before:

$$\pi^{(n)}(\delta u^{(n)} | \delta x^{(n)}) = \mathcal{N}\Big(\delta u^{(n)}; \delta u^{(n)^*}, \alpha (Q_{uu}^{(n)})^{-1} \Big) \tag{26}$$

where $\delta x^{(n)} = x - \bar{x}^{(n)}$ and $\delta u^{(n)} = u - \bar{u}^{(n)}$ are now evaluated relative to the nominal trajectories for corresponding to the nth approximation, with $\delta u^{(n)^*}$ defined analogous to (11) but using the approximations around $(\bar{x}^{(n)}, \bar{u}^{(n)})$. Importantly, from the form of (25), we see that π is a mixture of Gaussians with component weights $w^{(n)}$ computed using the quadratic approximation of the value function (12) and are *adaptive to disturbances*. We refer the readers to Appendix C in [17] for more details.

Since both z and π are weighted sums of $z^{(n)}$ and $\pi^{(n)}$, computing the solution to the backward pass of MME-DDP is equivalent to solving for ME-DDP around the N different nominal trajectories and then *composing* the value functions and policies using (24) and (25).

Algorithm 1: Backward Pass

Compute $V(T), V_x(T)$ and $V_{xx}(T)$ using Φ
for $t = T-1$ **to** 0 **do**
  Compute l, Q and their derivatives for timesteps t
  Regularize Q_{uu} to be PD
  Compute $k_t, K_t, V_x(t-1), V_{xx}(t-1)$ as in Vanilla DDP
  $\Sigma_t \leftarrow \alpha Q_{uu}^{-1}$
  $V_H \leftarrow V_H + \frac{\alpha}{2}(\ln|Q_{uu}| - n_u \ln(2\pi\alpha))$

V. Connections to Existing Works

A. Compositionality and Linear Solvable Optimal Control

A key component of our work is the compositionality of policies–solving for the full policy π by solving for the individual policies $\pi^{(n)}$ then combining them via (25). In [19], the KL Divergence regularized control is considered and the compositionality of controllers is introduced by exploiting the linearity of the exponential value function and the optimal policy. Unlike [19], we allows the running cost $l(x, u)$ to be an arbitrary function of the controls u. Furthermore, we provide practical algorithms in the form of ME-DDP and MME-DDP. Similarly, in the field of RL [20], compositionality has been used on maximum entropy optimal policies to solve a conjunction of tasks by combining maximum entropy policies which solve each of the tasks individually.

Unlike the above works, the approach our work takes focuses on the topic of exploration rather than compositionality. Our work is most similar to [10], which shows that the multimodal exploration is able to outperform similar methods which only consider unimodal exploration policies. However, we make use of compositionality and solve for the value function explicitly using DDP which allows for the policy to be recomputed online in realtime.

B. Exploration and Control-as-Inference

Our work is related to the Control-as-Inference framework [14, 21, 22, 23], where finding the optimal policy is posed as an inference problem by minimizing the KL divergence, with maximum entropy emerging as a special case of this when the prior is uniform. This framework provides a natural exploration strategy based on entropy maximization. Since the optimal policy is usually intractable, approaches in this area approximate the optimal policy distribution using either neural networks or by using tractable surrogates such as Gaussian distributions. Our work can be viewed as an extension to the latter approach by considering mixtures of Gaussians instead of unimodal Gaussians.

Our work has similar flavors to SaDDP in [24] as both methods rely on DDP and incorporate sampling. In their case, sampling is leveraged to address the problem of discontinuity and bypass the use of analytical derivatives, whereas this work relies on analytical derivatives but uses sampling to explore multiple modes simultaneously.

Algorithm 2: (Unimodal) Maximum Entropy DDP

Input: Number of iterations I, Resample frequency m
Initialize $x^{(1:2)}, u^{(1:2)}, K^{(1:2)}, \Sigma^{(1:2)}$
for $i = 1$ **to** I **do**
  if $i \ \% \ m = 0$ **then**
    $x^{(1)}, u^{(1)}, K^{(1)}, \Sigma^{(1)} \leftarrow$ lowest code mode
    $x^{(2)}, u^{(2)}, K^{(2)} \sim \pi^{(1)}$
  for $n = 1$ **to** 2 **in parallel do**
    $x^{(n)} \leftarrow$ Rollout dynamics
    $k^{(n)}, K^{(n)}, \Sigma^{(n)}, V_H^{(n)} \leftarrow$ Backward Pass
    $x^{(n)}, u^{(n)}, J^{(n)} \leftarrow$ Line Search

Algorithm 3: Multimodal Maximum Entropy DDP

Input: Number of GMM components N, Number of iterations I, Resample frequency m
Initialize $x^{(1:N)}, u^{(1:N)}, K^{(1:N)}, \Sigma^{(1:N)}, \pi$
for $i = 1$ **to** I **do**
  if $i \ \% \ m = 0$ **then**
    $x^{(1)}, u^{(1)}, K^{(1)} \leftarrow$ lowest code mode
    $x^{(2:N)}, u^{(2:N)}, K^{(2:N)} \sim$ GMM π with weights $w^{(1:N)}$
  for $n = 1$ **to** N **in parallel do**
    $x^{(n)} \leftarrow$ Rollout dynamics
    $k^{(n)}, K^{(n)}, \Sigma^{(n)}, V_H^{(n)} \leftarrow$ Backward Pass
    $x^{(n)}, u^{(n)}, J^{(n)} \leftarrow$ Line Search
    Compute $w^{(n)}$ using $J^{(n)}$ and $V_H^{(n)}$

VI. Algorithms

There are three main algorithmic issues that need to be addressed when implementing ME-DDP and MME-DDP.

Forward Pass: The derivation in earlier sections only describes how to perform the backward pass of DDP to compute the optimal Gaussian mixture policy (25), leaving the question of how to apply the new stochastic policy for the forward pass unanswered.

For ME-DDP, we perform multiple realizations of the stochastic policy at each timestep. Taking N realizations for each of the T timesteps will result in polynomial growth $O(T^N)$ of required samples. Instead, we sample the entire feedforward controls from the stochastic policy at $t = 0$ and then apply the deterministic feedback policy for times $t = 1$ to $T - 1$. To handle the added multi-modality of the optimal MME-DDP policy, we sample from a categorical distribution to determine which of the N modes will be used for the control of a particular sample trajectory, and then sample the feedforward controls as in the ME-DDP case.

Convergence: With a stochastic policy, the cost of the trajectory after sampling may be higher in cost than the original trajectory or even unbounded. To guarantee the convergence of the algorithms, we draw inspirations from [25] and apply the mean deterministic controls from the mode with the smallest cost to at least one sampled trajectory. This guarantees that the minimum cost over all N samples

TABLE I: Comparison of the mean and standard deviations of the cost for vanilla DDP, ME-DDP and MME-DDP, computed on 16 different DDP runs. The best mean cost for each system is boldfaced. Positive values of mean reduction correspond to a reduction. Significant reduction in mean and standard deviation can be observed from MME over both ME and vanilla DDP.

	Vanilla		ME		MME		MME vs Vanilla	MME vs ME
System	Mean	Std	Mean	Std	Mean	Std	ΔMean%	ΔMean%
2D Point Mass	32.25	0.00	10.76	9.55	**1.76**	0.00	94.55	83.69
Car	5.31	0.00	4.99	0.64	**3.76**	0.87	29.16	24.63
Quadcopter	0.98	0.00	0.90	0.18	**0.54**	0.02	45.08	40.25
Manipulator	22.84	0.00	20.28	4.68	**12.56**	3.68	45.00	38.06

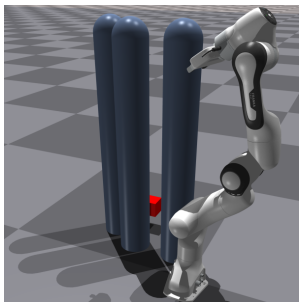


Fig. 2: Task setup for the manipulator. The goal is to reach the red block past the obstacles while avoiding collisions.

Algorithm	Mean (ms)	Std (ms)
DDP	1.537	0.016
MMEDDP	2.826	0.118

TABLE II: Comparison of computation times for 16 iterations averaged over 16 different runs.

is monotonically decreasing, preserving the convergence properties of DDP:

Lemma 4. *Each iterate of ME-DDP and MME-DDP results in a cost that is no worse than vanilla DDP given the current best nominal control is identical.*

We refer the readers to Appendix D in [17] for the proof.

Frequency of Sampling: Since the weights $w^{(n)}$ for each mode for MME-DDP are proportional to the value function $Z^{(n)}$, modes which have high cost are unlikely to be resampled in the next iteration of the forward pass even if they will converge to a more optimal local minimum given enough DDP iterations. To alleviate this issue, we only resample the controls for each mode after every m iterations, increasing the probability of jumping out of suboptimal local minima.

The full ME-DDP and MME-DDP algorithms are presented in Algorithm 2 and Algorithm 3, along with their backward pass in Algorithm 1, where variables without indices denote the entire trajectory, ($x^{(n)}$ denotes $x^{(n)}_{0:T}$). In short, both algorithms consist of keeping the lowest cost sample and sampling the rest from the stochastic policy π every m iterations for the forward pass, then running the backward pass for each sample. Both passes can be executed in parallel for each sample.

VII. Simulations

In this section, we compare the performance of the proposed MME-DDP algorithm against the ME-DDP and vanilla DDP algorithms on four systems: 2D Point Mass, 2D Car, Quadcopter and Manipulator. Obstacle avoidance is implemented as a soft-constraint with $l_{\text{obs}} = \exp\left(-\frac{d^2_{\text{obs}}}{2r^2_{\text{obs}}}\right)$, where d_{obs} and r_{obs} are the distance and radius of the obstacle respectively. The controls are zero-initialized for all systems. For the resampling frequency, we set $m = 8$. Table II compares the mean and standard deviation of the solver times on the 2D Car problem for DDP and MME-DDP with 8 modes on a Ryzen 9 3950X processor. A video comparison of the Quadcopter and Manipulator systems is available at this url.

A. 2D Point Mass

We first test the algorithms on an illustrative 2D point-mass double integrator reaching task while while avoiding obstacles in a maze-like environment. Both the top and middle paths are suboptimal local minima as they are blocked by obstacles, with the top path having an obstacle near the end of the path. The results are shown in the first row of Fig. 3.

B. 2D Car

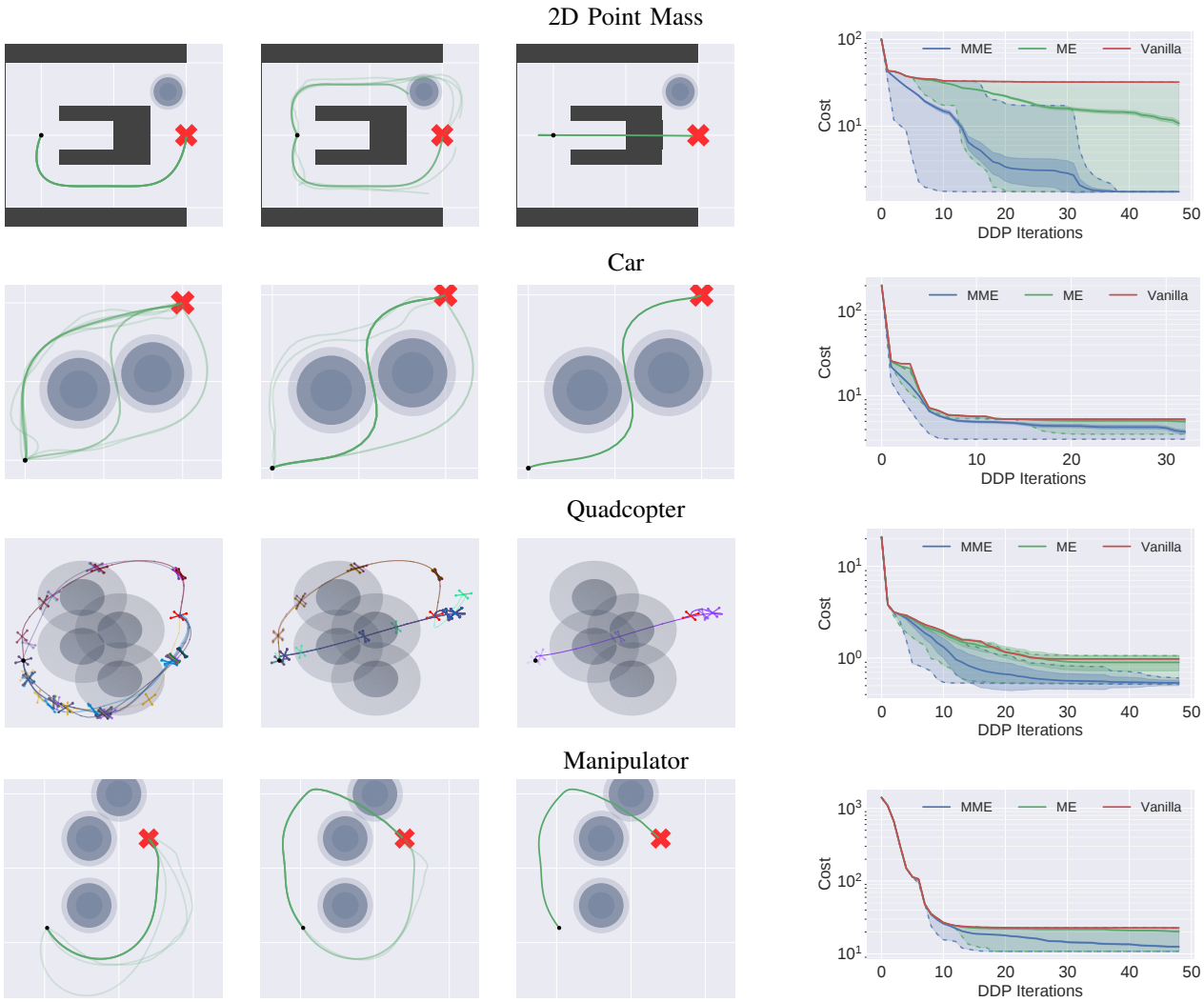
We next test on a 2D Car with dynamics of Dubin's vehicle under jerk control. The task here is again a reaching task while avoiding two circular obstacles. A suboptimal local minimum exists in the middle which goes in between both obstacles. The results are shown in the second row of Fig. 3.

C. Quadcopter

We test on a 3D quadcopter with states $x = [p_x, p_y, p_z, \Psi, \theta, \phi, v_x, v_y, v_z, p, q, r]^{\mathrm{T}} \in \mathbb{R}^{12}$ and controls $u = [f_t, \tau_x, \tau_y, \tau_z]^{\mathrm{T}} \in \mathbb{R}^4$. We refer readers to [26] for a full description of the dynamics. The task is to reach a target on the other side of four spherical obstacles set up in a square pattern. A suboptimal local minima is present in the intersection of all four obstacles in the center which only MME-DDP is able to consistently escape from, as shown in the third row of Fig. 3.

D. Manipulator

Finally, we test on a torque-controlled 7-DOF manipulator based off a simplified version of the Franka EMIKA Panda arm. The task here is for the end effector to reach the goal position without colliding with obstacles (see Fig. 2). Cylindrical obstacles are placed between the starting position and the end effector, creating multiple suboptimal local minima in the cost landscape. Again, only MME-DDP is able to consistently reach the target without intersecting any of the obstacles, as shown in the bottom row of Fig. 3.



(a) MME-DDP (b) ME-DDP (c) Vanilla DDP (d) Convergence Plot

Fig. 3: (a)–(c) Position trajectories for the 2D point mass, car, quadcopter and manipulator systems from 16 different DDP runs. For the manipulator, the projections of the end effector trajectories on the XY-plane are plotted. (d) Convergence plots for all three algorithms. The solid line denotes the mean, the dark shaded region represents the 2σ relative uncertainty, while the dotted lines denote the minimum and maxmimum costs. In all examples, MME-DDP is able converge to a better global minimum due to having better exploration.

Performance Comparison: Comparing the three algorithms, we observe that vanilla DDP consistently gets stuck in local minima and unimodal ME-DDP explores several minima but cannot explore each consistently. In contrast, the additional exploration capability helps MME-DDP find the best minimum. As obstacles are implemented as soft constraints, the shaded region around obstacles in Fig. 3 only provides a visualization and is **not** the obstacle boundary. We also present a comparison of convergence for each algorithm in Table I and Fig. 3d. Across all tasks, it is clear that both ME-DDP and MME-DDP are able to achieve a lower mean cost than vanilla DDP due to converging to a more optimal local minimum. Furthermore, MME-DDP is able to consistently achieve a significantly lower cost, highlighting the advantages of multimodal exploration.

VIII. Conclusion

In this paper, we derived ME-DDP and MME-DDP, two algorithms based off the maximum entropy formulation of DDP which provide improved exploration capabilities over the vanilla algorithm. Our results suggest that the added stochasticity and multimodal exploration improves the ability of DDP to escape from suboptimal local minima in environments with multiple local minima.

Future work include hardware implementation to verify the exploration benefits of the proposed algorithms. On the theoretical side we will investigate the conditions and rate of convergence, as well as generalizations that include the stochastic, risk sensitive and model predictive control cases.

Acknowledgment

Z. Wang and E.A. Theodorou was supported by NSF-CMMI Award #1936079.

REFERENCES

[1] Lev Semenovich Pontryagin. Mathematical theory of optimal processes. CRC press, 1987.

[2] Mustafa Mukadam, Xinyan Yan, and Byron Boots. Gaussian process motion planning. In 2016 IEEE international conference on robotics and automation (ICRA), pages 9–15. IEEE, 2016.

[3] David Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. International Journal of Control, 3(1): 85–95, 1966.

[4] David H Jacobson and David Q Mayne. Differential dynamic programming. Number 24. Elsevier Publishing Company, 1970.

[5] L-Z Liao and Christine A Shoemaker. Convergence in unconstrained discrete-time differential dynamic programming. IEEE Transactions on Automatic Control, 36(6):692–706, 1991.

[6] Helen Oleynikova, Michael Burri, Zachary Taylor, Juan Nieto, Roland Siegwart, and Enric Galceran. Continuous-time trajectory optimization for online uav replanning. In 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), pages 5332–5339. IEEE, 2016.

[7] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. Integrated online trajectory planning and optimization in distinctive topologies. Robotics and Autonomous Systems, 88:142–153, 2017.

[8] Boyu Zhou, Fei Gao, Jie Pan, and Shaojie Shen. Robust real-time uav replanning using guided gradient-based optimization and topological paths. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 1208–1214. IEEE, 2020.

[9] Brian D Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. PhD thesis, Carnegie Mellon University, 2010.

[10] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In International Conference on Machine Learning, pages 1352–1361. PMLR, 2017.

[11] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning, pages 1861–1870. PMLR, 2018.

[12] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 1714–1721. IEEE, 2017.

[13] Ziyi Wang, Grady Williams, and Evangelos A Theodorou. Information theoretic model predictive control on jump diffusion processes. In 2019 American Control Conference (ACC), pages 1663–1670. IEEE, 2019.

[14] Ziyi Wang, Oswin So, Jason Gibson, Bogdan Vlahov, Manan Gandhi, Guan-Horng Liu, and Evangelos Theodorou. Variational Inference MPC using Tsallis Divergence. In Proceedings of Robotics: Science and Systems, Virtual, July 2021. doi: 10.15607/RSS.2021.XVII.073.

[15] Kyungjae Lee, Sungyub Kim, Sungbin Lim, Sungjoon Choi, Mineui Hong, Jaein Kim, Yong-Lae Park, and Songhwai Oh. Generalized tsallis entropy reinforcement learning and its application to soft mobile robots. Robotics: Science and Systems Foundation, 2020.

[16] Jeongho Kim and Insoon Yang. Hamilton-jacobi-bellman equations for maximum entropy optimal control. arXiv preprint arXiv:2009.13097, 2020.

[17] Oswin So, Ziyi Wang, and Evangelos A Theodorou. Maximum entropy differential dynamic programming. arXiv preprint arXiv:2110.06451, 2021. URL `https://arxiv.org/pdf/2110.06451.pdf`.

[18] Weiwei Li and Emanuel Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In ICINCO (1), pages 222–229. Citeseer, 2004.

[19] Emanuel Todorov. Compositionality of optimal control laws. Advances in neural information processing systems, 22:1856–1864, 2009.

[20] Tuomas Haarnoja, Vitchyr H. Pong, Aurick Zhou, Murtaza Dalal, P. Abbeel, and Sergey Levine. Composable deep reinforcement learning for robotic manipulation. 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 6244–6251, 2018.

[21] Marc Toussaint. Robot trajectory optimization using approximate inference. In Proceedings of the 26th annual international conference on machine learning, pages 1049–1056, 2009.

[22] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. Proceedings of Robotics: Science and Systems VIII, 2012.

[23] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. arXiv preprint arXiv:1805.00909, 2018.

[24] Joose Rajamäki, Kourosh Naderi, Ville Kyrki, and Perttu Hämäläinen. Sampled differential dynamic programming. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1402–1409. IEEE, 2016.

[25] Jing Dong and Xin T Tong. Replica exchange for non-convex optimization. Journal of Machine Learning Research, 22(173):1–59, 2021.

[26] Francesco Sabatino. Quadrotor control: modeling, nonlinearcontrol design, and simulation. Master's thesis, KTH Royal Institute of Technology, 2015.