Data-Driven Distributionally Robust Optimal Control with State-Dependent Noise

Rui Liu, Guangyao Shi, Pratap Tokekar

***Abstract*— Distributionally Robust Optimal Control (DROC) is a technique that enables robust control in a stochastic setting when the true distribution is not known. Traditional DROC approaches require given ambiguity sets or a KL divergence bound to represent the distributional uncertainty. These may not be known a priori and may require hand-crafting. In this paper, we lift this assumption by introducing a data-driven technique for estimating the uncertainty and a bound for the KL divergence. We call this technique D3ROC. To evaluate the effectiveness of our approach, we consider a navigation problem for a car-like robot with unknown noise distributions. The results demonstrate that D3ROC provides robust and efficient control policies that outperform the iterative Linear Quadratic Gaussian (iLQG) control. The results also show the effectiveness of our proposed approach in handling different noise distributions.**

I. Introduction

The objective of optimal control [1] is to determine the optimal control inputs and state trajectories for a dynamical system that minimize or maximize a pre-defined objective. Most real-world systems are faced with significant uncertainties [2] coming from various sources such as measurement noise, inaccurate dynamics models, and environmental disturbances. There are two popular control techniques to address these sources of uncertainty: stochastic control [3, 4] and robust control [5, 6]. Stochastic control incorporates probabilistic models of uncertainty into the control design process. The underlying distribution of the uncertainty must be known in order for the control design to be effective. Robust control provides robustness to a wide range of uncertainties by considering worst-case scenarios. Both approaches provide practical solutions for controlling dynamical systems in the presence of uncertainty. However, both approaches have limitations. For stochastic control, the probability distribution of uncertainty may not always be available or may be difficult to determine. For robust control, considering worst-case scenarios can sometimes lead to overly conservative control designs that are not necessary and even sub-optimal.

The limitations of stochastic and robust control methods have led to the development of Distributionally Robust Optimal Control (DROC) [7, 8], a rapidly advancing area in control theory that addresses uncertainty in both the system model and the process noise. It merges the robustness properties of Distributionally Robust Optimization (DRO) [9, 10] with the predictive capabilities of Model Predictive Control (MPC) [11, 12]. It can provide control policies that are robust to uncertain and changing conditions. Unlike traditional stochastic control methods, DROC does not assume prior knowledge of the underlying distributions of the uncertainty. Instead, it assumes there is an ambiguity set of probability measures $\mathbb{P}$ that contain the true distribution, and this $\mathbb{P}$ should be as small as possible [9].

A common way of modeling the ambiguity set is by assuming a reference distribution such that the true distribution is within a given KL divergence bound of this reference distribution [7, 13–15]. However, in practical applications, it may not be possible to obtain these values in advance. In this paper, we aim to address the following question: *Can we learn the reference distribution and KL divergence bound to enhance the effectiveness of DROC?*

Building on DROC [7], we present Data-Driven Distributionally Robust Optimal Control (D3ROC) that learns the reference distirbution as well as the KL divergence bound. To achieve this, we utilize Gaussian Process (GP) regression [16] to estimate the reference noise distribution, and we employ k Nearest Neighbor (kNN) [17] to estimate the KL divergence bound. Here, we consider both stationary and state-dependent noise distributions.

The numerical results show that D3ROC achieves smaller mean final distances between the robot and the origin, along with lower corresponding standard deviations, compared to iLQG. Furthermore, under D3ROC, the robot tends to avoid regions with higher noise variance due to its risk-averse nature. Conversely, the risk-neutral control approach, iLQG, leads the robot to pass through regions with higher variance. These findings demonstrate the ability of D3ROC to provide more robust and efficient control policies compared to iLQG.

II. Problem Formulation

Consider the following general nonlinear stochastic system:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{g}(\mathbf{x}_t, \mathbf{u}_t)\mathbf{w}(\mathbf{x}_t, \mathbf{u}_t), \tag{1}$$

where $\mathbf{x}_t \in \mathrm{X} \subset \mathbb{R}^n$ and $\mathbf{u}_t \in \mathrm{U} \subset \mathbb{R}^m$ denote the state and control of the system at step t, respectively. $\mathbf{f}$ is the dynamics model of the system, $\mathbf{g}$ is a mapping function, $\mathbf{w} \in \mathbb{R}^c$ is the process noise with unknown distribution. $\mathbf{w}$ may be stationary or vary with states or control inputs.

Even though the true distribution p of the process noise $\mathbf{w}$ is unknown, we assume it is contained in an ambiguity set $\mathbb{P}$ with reference distribution q [13]. The ambiguity set $\mathbb{P}$ is

All authors are from the University of Maryland, College Park, MD 20742 USA. {ruiliu, gyshi, tokekar}@umd.edu

This work is supported in part by the National Science Foundation Grant No. 1943368 and the Army Grant No. W911NF2120076.

constructed by:

$$\mathbb{P} = \{p : D(p||q) \leq d\} \ , \tag{2}$$

where $D(\cdot||\cdot)$ is the KL divergence and $d > 0$ is the bound.

J is the cost function over a finite horizon n:

$$J = \sum_{t=0}^{n-1} l_t(\mathbf{x}_t, \mathbf{u}_t) + l_f(\mathbf{x}_n) \ , \tag{3}$$

where l_t is the stage cost, l_f is the terminal cost, we consider linear quadratic regulator:

$$l_t(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^\intercal \mathbf{Q}_t \mathbf{x} + \frac{1}{2}\mathbf{u}^\intercal \mathbf{R}_t \mathbf{u} \ , \tag{4}$$

$$l_f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\intercal \mathbf{Q}_f \mathbf{x} \ . \tag{5}$$

In formulating the DROC problem, we are interested in finding a control policy that minimizes the worst-case expected value of the cost function J:

$$\min_{\mathbf{u}\in \mathrm{U}} \max_{p\in\mathbb{P}} \mathbb{E}_p[J] \ . \tag{6}$$

According to [7, 13], the above min-max problem over true distribution p can be converted to the following min-min problem by taking the expectation over the reference distribution q:

$$\min_{\theta\in\Xi} \min_{\mathbf{u}\in \mathrm{U}} \left\{ \frac{1}{\theta} \log[\mathbb{E}_q(e^{\theta J})] + \frac{d}{\theta} \right\} \ , \tag{7}$$

where θ is called the risk-sensitivity parameter, and Ξ is a non-empty set of positive θ that gives finite entropic risk measure.

To solve this DROC problem, knowledge of the reference distribution q of the noise, and the KL divergence bound d, are required. In the next section, we show how to compute q and d in a data-driven fashion when they are unknown.

III. The D3ROC Solution

In this section, we present D3ROC. The objective of D3ROC is to first estimate the uncertainty distribution and a bound for the KL divergence, and then solve the optimization problem of Eq. 7. D3ROC distinguishes itself from traditional DROC approaches that rely on given ambiguity sets [7, 13–15] by utilizing data-driven techniques. In Section III-A, we address the inner minimization over control inputs $\mathbf{u}$ using Differential Dynamic Programming (DDP) [7, 18, 19]. However, for DDP to be effective, we require the reference distribution q, which we estimate using observed data. Next, for the outer minimization over the risk-sensitivity parameter θ, we adopt the cross-entropy method as described in [7]. To apply this method successfully, we estimate the KL divergence bound d. We study two scenarios for the parameters estimation: (1) stationary noise distribution in Section III-B, and (2) state-dependent noise distribution in Section III-C.

A. Differential Dynamic Programming

We utilize DDP to handle the inner minimization of Eq. 7, which is equivalent to minimizing $\frac{1}{\theta}\log[\mathbb{E}_q(e^{\theta J})]$, defined as $R_\theta(J)$, also known as the entropic risk measure [7, 20]. We assume $\mathbf{g}$ in Eq. 1 as an identity mapping for simplicity. The process noise $\mathbf{w}$ is estimated as a Gaussian with zero mean and covariance matrix $\mathbf{W}$. We linearly approximate the system dynamics and quadratically approximate the cost function in terms of state and control deviations $\delta\mathbf{x}_t = \mathbf{x}_t - \mathbf{x}_t^{nom}$ and $\delta\mathbf{u}_t = \mathbf{u}_t - \mathbf{u}_t^{nom}$, where $\mathbf{x}_t^{nom}$ and $\mathbf{u}_t^{nom}$ are the nominal trajectories.

The linearized model:

$$\delta\mathbf{x}_{t+1} = \mathbf{A}_t\delta\mathbf{x}_t + \mathbf{B}_t\delta\mathbf{u}_t + \mathbf{w}_t \ , \tag{8}$$

the stage cost approximation:

$$\begin{aligned}\tilde{l}_t(\delta\mathbf{x}_t, \delta\mathbf{u}_t) = q_t + \mathbf{q_t}^\intercal\delta\mathbf{x}_t + \mathbf{r_t}^\intercal\delta\mathbf{u}_t + \frac{1}{2}\delta\mathbf{x}_t^\intercal\mathbf{Q}_t\delta\mathbf{x}_t \\ + \frac{1}{2}\delta\mathbf{u}_t^\intercal\mathbf{R}_t\delta\mathbf{u}_t \ ,\end{aligned} \tag{9}$$

and the terminal cost approximation:

$$\tilde{l}_f(\delta\mathbf{x}_n) = q_n + \mathbf{q}_n^\intercal\delta\mathbf{x}_n + \frac{1}{2}\delta\mathbf{x}_n^\intercal\mathbf{Q}_n\delta\mathbf{x}_n \ . \tag{10}$$

Then apply the principle of optimality, the Bellman equation for solving the optimal value function is:

$$V_t(\delta\mathbf{x}_t) = \min_{\delta\mathbf{u}_t} \left\{ \tilde{l}_t(\delta\mathbf{x}_t, \delta\mathbf{u}_t) + R_\theta\left(V_{t+1}(\delta\mathbf{x}_{t+1})\right) \right\} \ , \tag{11}$$

where R_θ is the entropic risk measure.

Suppose the value function is of quadratic form expressed as:

$$V_t(\delta\mathbf{x}) = \frac{1}{2}\delta\mathbf{x}^\intercal\mathbf{S}_t\delta\mathbf{x} + \mathbf{s}_t^\intercal\delta\mathbf{x} + s_t \ . \tag{12}$$

Then,

$$\begin{aligned}&V_{t+1}(\delta\mathbf{x}_{t+1}) \\ &= \frac{1}{2}\delta\mathbf{x}_{t+1}^\intercal\mathbf{S}_{t+1}\delta\mathbf{x}_{t+1} + \mathbf{s}_{t+1}^\intercal\delta\mathbf{x}_{t+1} + s_{t+1} \\ &= \frac{1}{2}(\mathbf{A}_t\delta\mathbf{x}_t + \mathbf{B}_t\delta\mathbf{u}_t + \mathbf{w}_t)^\intercal\mathbf{S}_{t+1}(\mathbf{A}_t\delta\mathbf{x}_t + \mathbf{B}_t\delta\mathbf{u}_t + \mathbf{w}_t) + \\ &\quad \mathbf{s}_{t+1}^\intercal(\mathbf{A}_t\delta\mathbf{x}_t + \mathbf{B}_t\delta\mathbf{u}_t + \mathbf{w}_t) + s_{t+1} \ .\end{aligned} \tag{13}$$

Let $\mathbf{z}_t = \mathbf{A}_t\delta\mathbf{x}_t + \mathbf{B}_t\delta\mathbf{u}_t + \mathbf{w}_t$, where $\mathbf{A}_t$ and $\mathbf{B}_t$ are Jacobian matrices of the system model. The true distribution of the noise $\mathbf{w}$ is unknown. But we model it as a Gaussian distribution with zero mean and covariance $\mathbf{W}$, then $\mathbf{z}_t \sim$

$\mathcal{N}(\mathbf{A}_t\delta\mathbf{x}_t+\mathbf{B}_t\delta\mathbf{u}_t,\mathbf{W}_t)$. Put Eq. 13 into Eq. 11, we have:

$$
\begin{aligned}
&V_t(\delta\mathbf{x}_t)\\
&=\min_{\delta\mathbf{u}_t}\Bigg\{q_t+\mathbf{q_t}^\intercal\delta\mathbf{x}_t+\mathbf{r_t}^\intercal\delta\mathbf{u}_t+\frac{1}{2}\delta\mathbf{x}_t^\intercal\mathbf{Q}_t\delta\mathbf{x}_t+\frac{1}{2}\delta\mathbf{u}_t^\intercal\mathbf{R}_t\delta\mathbf{u}_t\\
&\quad+\frac{1}{\theta}log\Bigg\{\mathbb{E}_{\mathbf{w}_t}\Bigg[exp\Bigg(\theta\Bigg(\frac{1}{2}(\mathbf{A}_t\delta\mathbf{x}_t+\mathbf{B}_t\delta\mathbf{u}_t+\mathbf{w}_t)^\intercal\mathbf{S}_{t+1}\\
&\quad(\mathbf{A}_t\delta\mathbf{x}_t+\mathbf{B}_t\delta\mathbf{u}_t+\mathbf{w}_t)+\mathbf{s}_{t+1}^\intercal(\mathbf{A}_t\delta\mathbf{x}_t+\mathbf{B}_t\delta\mathbf{u}_t+\mathbf{w}_t)\\
&\quad+s_{t+1}\Bigg)\Bigg)\Bigg]\Bigg\}\Bigg\}\\
&=\min_{\delta\mathbf{u}_t}\Bigg\{q_t+\mathbf{q_t}^\intercal\delta\mathbf{x}_t+\mathbf{r_t}^\intercal\delta\mathbf{u}_t+\frac{1}{2}\delta\mathbf{x}_t^\intercal\mathbf{Q}_t\delta\mathbf{x}_t+\frac{1}{2}\delta\mathbf{u}_t^\intercal\mathbf{R}_t\delta\mathbf{u}_t+\\
&\quad\frac{1}{2}(\mathbf{A}_t\delta\mathbf{x}_t+\mathbf{B}_t\delta\mathbf{u}_t)^\intercal\mathbf{S}_{t+1}(\mathbf{A}_t\delta\mathbf{x}_t+\mathbf{B}_t\delta\mathbf{u}_t)\\
&\quad+(\mathbf{A}_t\delta\mathbf{x}_t+\mathbf{B}_t\delta\mathbf{u}_t)^\intercal\mathbf{s}_{t+1}\\
&\quad+\frac{1}{\theta}log\Bigg\{\mathbb{E}_{\mathbf{z}_t}\Bigg[exp\Bigg(\theta\Bigg(\frac{1}{2}\mathbf{z}_t^\intercal\mathbf{S}_{t+1}\mathbf{z}_t+\mathbf{s}_{t+1}^\intercal\mathbf{z}_t\Bigg)\Bigg)\Bigg]\Bigg\}\\
&\quad+s_{t+1}\Bigg\}\ ,
\end{aligned}
\tag{14}
$$

where $q_t,\mathbf{q}_t,\mathbf{r}_t,\mathbf{Q}_t,\mathbf{R}_t$ are the Taylor expansion coefficients of the cost function around the nominal trajectory.

The expectation on the right hand side of Eq. 14 can be calculated using characteristic function of Gaussian distribution. Then we can minimize over $\delta\mathbf{u}$ and get the optimal control policy:

$$
\begin{aligned}
\delta\mathbf{u}_t&=\mathbf{k}_t+\mathbf{K}_t\delta\mathbf{x}_t\ ,\\
\mathbf{k}_t&=-\mathbf{H}_t^{-1}\mathbf{g}_t\ ,\\
\mathbf{K}_t&=-\mathbf{H}_t^{-1}\mathbf{G}_t\ ,
\end{aligned}
\tag{15}
$$

where

$$
\begin{aligned}
\mathbf{M}_t&=\mathbf{W}_t^{-1}-\theta\mathbf{S}_{t+1}\ ,\\
\mathbf{H}_t&=\mathbf{R}_t+\mathbf{B}_t^\intercal\big(\mathbf{I}+\theta\mathbf{S}_{t+1}\mathbf{M}_t^{-1}\big)\mathbf{S}_{t+1}\mathbf{B}_t\ ,\\
\mathbf{G}_t&=\mathbf{B}_t^\intercal\big(\mathbf{I}+\theta\mathbf{S}_{t+1}\mathbf{M}_t^{-1}\big)\mathbf{S}_{t+1}\mathbf{A}_t\ ,\\
\mathbf{g}_t&=\mathbf{r}_t+\mathbf{B}_t^\intercal\big(\mathbf{I}+\theta\mathbf{S}_{t+1}\mathbf{M}_t^{-1}\big)\mathbf{s}_{t+1}\ .
\end{aligned}
\tag{16}
$$

The backward recursions are:

$$
\begin{aligned}
\mathbf{S}_t&=\mathbf{Q}_t+\mathbf{A}_t^\intercal\big(\mathbf{I}+\theta\mathbf{S}_{t+1}\mathbf{M}^{-1}\big)\mathbf{S}_{t+1}\mathbf{A}_t\\
&\quad+\mathbf{K}_t^\intercal\mathbf{H}_t\mathbf{K}_t+\mathbf{K}_t^\intercal\mathbf{G}_t+\mathbf{G}_t^\intercal\mathbf{K}_t\ ,\\
\mathbf{s}_t&=\mathbf{q}_t+\mathbf{A}_t^\intercal\big(\mathbf{I}+\theta\mathbf{S}_{t+1}\mathbf{M}_t^{-1}\big)\mathbf{s}_{t+1}\\
&\quad+\mathbf{K}_t^\intercal\mathbf{H}_t\mathbf{k}_t+\mathbf{K}_t^\intercal\mathbf{g}_t+\mathbf{G}_t^\intercal\mathbf{k}_t\ ,\\
s_t&=q_t+s_{t+1}-\frac{1}{2\theta}\log\big(\det(\mathbf{I}-\theta\mathbf{W}_t\mathbf{S}_{t+1})\big)\\
&\quad+\frac{\theta}{2}\mathbf{s}_{t+1}^\intercal\mathbf{M}_t^{-1}\mathbf{s}_{t+1}+\frac{1}{2}\mathbf{k}_t^\intercal\mathbf{H}_t\mathbf{k}_t+\mathbf{k}_t^\intercal\mathbf{g}_t\ ,
\end{aligned}
\tag{17}
$$

for $t=n$ goes backward to 0, with initial conditions $s_n=q_n,\mathbf{s}_n=\mathbf{q}_n,\mathbf{S}_n=\mathbf{Q}_n$.

After addressing the inner minimization using DDP, the outer minimization is solved using the cross-entropy method, following the approach described in [7]. However, for both DDP and cross-entropy method to be effective, it is required to estimate the noise reference distribution and the KL divergence bound.

We will focus on the next sections to address these requirements. Firstly, we examine the case of a stationary noise distribution in section III-B. Then, we investigate the case of a state-dependent noise distribution in section III-C.

B. Stationary Noise Distribution

1) Estimating Reference Distribution: We employ Maximum likelihood estimation (MLE) for estimating the parameters of a probability distribution based on observed data. Once we have estimated the parameters of q, we can use them to construct the ambiguity set for the DROC problem.

2) Estimating KL Divergence Bound: We utilize k Nearest Neighbor (kNN) [17] to estimate the KL divergence bound. This method is based on the assumption that the KL divergence between two distributions can be estimated from their samples. Euclidean distance is used to measure the distance between samples. According to [17], the estimated KL divergence $\hat{D}(p||q)$ between distributions p and q can be written as:

$$
\hat{D}(p\|q)=\frac{r}{N}\sum_{i=1}^{N}\ln\frac{\nu_i}{\rho_i}+\log\frac{M}{N-1}\ ,
\tag{18}
$$

where r is the dimension of the data, N is the number of samples drawn i.i.d from distribution p, M is the number of samples drawn i.i.d from distribution q. ρ_i is the distance between i-th element drawn from p and its k-th nearest neighbor in samples drawn from p except itself, ν_i is the distance between i-th element drawn from p and its k-th nearest neighbor in samples drawn from q.

Once the reference distribution q is estimated as a Gaussian, M samples can be drawn from q. The KL divergence bound can then be estimated using kNN estimation with these samples and N true noise samples.

C. State-Dependent Noise Distribution

1) Estimating Reference Distribution: In cases where the true distribution p of the noise is non-stationary and varies with the system's state, we can learn the reference distribution q of the noise from observed data, as shown in Algorithm 1. For each training state $\bar{\mathbf{x}}_j$ ($j=1,\ 2,\cdots,\ m$), N true noise samples $\{\mathbf{w}_j^{(1)},\ \mathbf{w}_j^{(2)},\ \cdots,\ \mathbf{w}_j^{(N)}\}$ are available. We assume that the noise of each dimension does not impact other dimensions for simplicity. Consequently, we can separate each dimension of the state and its corresponding noise. Then, we employ MLE to estimate the variance of the noise for each dimension. For instance, when considering the x coordinate of the state, we can obtain a set of tuples $\{(\bar{x}_1,\ v_1),\ (\bar{x}_2,\ v_2),\ \cdots,\ (\bar{x}_m,\ v_m)\}$, where v denotes the variance for the x coordinate.

We train a Gaussian Process (GP) estimator [16] for each dimension of the state to learn the reference distribution. We use the set $\{(\bar{\mathbf{x}}_1^{(i)},\ v_1^{(i)}),\ (\bar{\mathbf{x}}_2^{(i)},\ v_2^{(i)}),\ \cdots,\ (\bar{\mathbf{x}}_m^{(i)},\ v_m^{(i)})\}$, where $\bar{\mathbf{x}}_j^{(i)}$ denotes the ith dimension of the jth training state and $v_j^{(i)}$ denotes the corresponding variance of the

noise. We use zero mean and squared-exponential kernel function for GP. The kernel function $k(\cdot,\ \cdot)$ is defined as $k(a,a') = \sigma^2 \exp\left(-\frac{(a-a')^2}{2l^2}\right)$, where σ^2 is the signal variance, l is the length scale. The hyperparameters signal variance and length scale are learned by maximizing the log-likelihood of the training data.

When given a new state $\mathbf{x}$, we use the trained GPs to predict the variances of the noise for each dimension, denoted as $\{\hat{v}_1,\ \hat{v}_2,\ \ldots,\ \hat{v}_r\}$, where r is the number of dimensions. These predicted variances are then combined into a diagonal covariance matrix, which represents the reference distribution q for the given state.

Algorithm 1 Estimation of reference distribution for state-dependent noise

Input: State $\mathbf{x}$ of the system
Output: State-dependent reference distribution $q(\mathbf{x})$
Require: Training data $\{\{(\bar{\mathbf{x}}_j,\ \mathbf{w}_j)\}_{j=1}^{m}\}^N$

```
# Training stage
for i = 1 to r do
    for j = 1 to m do
        v_j^(i) ← MLE({w_ij^(1), w_ij^(2), ..., w_ij^(N)})   ▷ w_ij is the ith dimension of the jth noise
    end for
    Get a set {(x̄_1^(i), v_1^(i)), (x̄_2^(i), v_2^(i)), ..., (x̄_m^(i), v_m^(i))}
    Train a GP to predict variance v̂_i for ith dimension
end for
# Prediction stage
for i = 1 to r do
    v̂_i ← GP_i(x^(i))
end for
Set W(x) = diag([v̂_1, v̂_2, ..., v̂_r])
return q(x) ~ N(0, W(x))
```

2) Estimating KL Divergence Bound: If the true distribution of the noise is non-stationary and varies with the system's state, we need to estimate the joint reference distribution for each horizon, resulting in a varying KL divergence bound d for different horizons. The goal is to find a global maximum d that can bound the ambiguity sets for all horizons. To achieve this, we first combine $n+1$ noise samples from the training data as a joint vector $\mathbf{w}_{(1:n+1)}$, where n is the horizon steps. Subsequently we obtain the joint reference distribution $q_{(1:n+1)}$ using MLE. Next, we draw M samples from this joint reference distribution, each sample being a joint vector of dimension $c \times (n+1)$, where c is the dimension of one noise sample. Then we employ kNN estimation with N true samples of joint noise vectors to get a KL divergence bound for one entire horizon. We then recede the horizon, and the joint noise vector becomes $\mathbf{w}_{(2:n+2)}$, allowing us to obtain another d. We repeat this process for each horizon until we have $m-n$ KL divergence bounds, where m is the number of noise samples in the observed data. Finally, we take the maximum of these bounds to obtain the estimated global maximum bound of the KL divergence, as outlined in Algorithm 2.

Algorithm 2 Estimation of global maximum KL divergence bound

Require: Training data $\{\{(\bar{\mathbf{x}}_j,\ \mathbf{w}_j)\}_{j=1}^{m}\}^N$, receding horizon steps n, number of samples M drawn from joint reference distribution

```
Set d_max = 0
for j = 1 to m − n do
    Combine n noise samples as a joint vector: w_(j:j+n) = [w_j, ..., w_{j+n}]^T
    Estimate joint reference distribution: q_(j:j+n) = MLE(w_(j:j+n))
    Draw M samples of joint vector: {w̃_(j:j+n)}^M ~ q_(j:j+n)
    Obtain kNN estimation of bound with N true joint noise vectors: d_j = kNN({w_(j:j+n)}^N, {w̃_(j:j+n)}^M)
    if d_j > d_max then
        d_max = d_j
    end if
end for
return d_max
```

IV. Experiments

To validate D3ROC, we consider a navigation problem with a standard car-like robot [21], which is a widely used model due to its simplicity and effectiveness. The state of the robot is represented by a vector $\mathbf{x} = [x,\ y,\ \theta,\ v]$, where x and y are the coordinates of the robot, θ is the yaw angle which represents the angle between the orientation of the car and the x-axis, and v is the velocity. The control inputs for the robot are the acceleration a and steering angle δ. Our goal is to design a control policy that brings the robot to the origin as close as possible while accounting for the unknown noise. The length of the car-like robot is $0.3\ m$. We discretize the system with $dt = 0.1\ s$ and use horizon step $n = 10$.

We begin by generating training data through uniform discretization of the state space that the robot can traverse. This process results in a set of states $\{\bar{\mathbf{x}}_1,\ \bar{\mathbf{x}}_2,\ \cdots,\ \bar{\mathbf{x}}_m\}$. For each state $\bar{\mathbf{x}}_j$, we apply a known control signal, which allows us to obtain the true next state. By subtracting the noiseless next state generated by the dynamical model, we obtain the corresponding true state-dependent noise $\mathbf{w}_j$. We repeat this process N times, resulting in the training data $\{(\bar{\mathbf{x}}_j,\ \mathbf{w}j)_{j=1}^{m}\}^N$ with $m = 1000$ and $N = 1000$.

A. True Noise Distribution Construction

Suppose the true distribution p is a mixture of Gaussians with h finite components:

$$p(\mathbf{x}) = \sum_i^h \pi_i \mathcal{N}_i(\mathbf{0}, \mathbf{W}(\mathbf{x}))\ ,$$
$$\sum_i^h \pi_i = 1\ , \tag{19}$$

where each component is a multivariate normal distribution with zero mean and covariance matrix dependent on state $\mathbf{x}$:

$$\mathbf{W}_i(x) = diag([{}_i\sigma_x^2,\ {}_i\sigma_y^2,\ {}_i\sigma_\theta^2,\ {}_i\sigma_v^2)],\ \ i = 1,\ 2,\ \cdots,\ h. \tag{20}$$

In the experiments, we consider three different cases of the true noise distribution $p(\mathbf{x})$ as Gaussian mixtures: (a) two-component, (b) three-component, and (c) four-component. The formulas of these true distributions are given in the appendix.[1].

Taking the two-component Gaussian mixtures $p^{(b)}$ for illustration, the heatmap of variances of each component of $p^{(b)}$ for the x and y coordinates are shown in Fig. 1. As we can see, the variances in x, y coordinates are larger in the region $2.0 \leq x \leq 3.0$ m and $2.0 \leq y \leq 3.0$ m. The variances do not change in θ and v coordinates in the experiments. However, we note that the proposed approach is capable of handling cases where the variances may change in all dimensions of the states.

B. Reference Distribution Estimation

Considering the three different cases of state-dependent noise described above, we trained a Gaussian process (GP) for each dimension of the state to model the reference distribution q, as described in Section III-C.1. The GP training results of predicting the variances of $p^{(b)}$ are shown in Fig. 2 for instance. The black dots represent the variances estimated using MLE for the x and y coordinates. The line represents the mean, and the shaded area represents the 95% confidence interval. As shown in Fig. 2, the estimated variances in the x and y coordinates are larger in the region $2.0 \leq x \leq 3.0$ m and $2.0 \leq y \leq 3.0$ m, which captures the distribution of the true noise $p^{(b)}$. In this specific example, the variances do not change in the θ and v coordinates. The GP-estimated variances in the θ and v coordinates are ${}^{(b)}\sigma_{\theta}^2 = 1.3e^{-4}$ and ${}^{(b)}\sigma_{v}^2 = 2.2e^{-3}$, respectively. Combining the variances of each dimension into a diagonal covariance matrix allows us to obtain the state-dependent reference distribution q, and the results verify our proposed approach.

C. KL Divergence Bound Estimation

In the example of state-dependent noise, we estimated the KL divergence bound d using the kNN method. For each horizon, we drew $M = 100$ samples from the estimated joint reference distribution and used $k = 10$ for the kNN estimation. As a result, we obtained KL divergence bound estimates for three different true noise distributions: $d^{(a)} = 5.65$ for $p^{(a)}$, $d^{(b)} = 7.28$ for $p^{(b)}$, and $d^{(c)} = 6.22$ for $p^{(c)}$.

D. Comparison with iLQG

We compared the performance of D3ROC with the risk-neutral control approach iLQG [22]. The robot's initial state was $x = 5.0$ m, $y = 5.0$ m with a yaw angle of -0.75π and velocity of zero. The objective was to navigate to the origin under unknown noise. Both control approaches were executed as a MPC for 22 iterations. To ensure accuracy, we performed 15 runs for each true noise distribution, and the resulting paths of the robot under noise $p^{(b)}$ are plotted in Fig. 3. Both control approaches enable the robot to approach the origin. However, D3ROC outperforms iLQG with smaller distances to the origin and more compact paths as the robot gets closer. The plot also reveals D3ROC's risk-averse behavior, leading the robot to navigate around the region $2.0 \leq x \leq 3.0$ m and $2.0 \leq y \leq 3.0$ m to avoid areas with higher noise variance. In contrast, iLQG results in more divergent paths, passing through the region $2.0 \leq x \leq 3.0$ m and $2.0 \leq y \leq 3.0$ m.

Additionally, we present the mean distance between the final position of the robot and the origin for multiple runs under different true noise distributions p in Table I. It is evident that the mean final distances and their corresponding standard deviations are smaller under D3ROC compared to iLQG. This finding further supports the effectiveness of our proposed data-driven approach, D3ROC, and demonstrates its ability to successfully handle various true noise distributions.

	$p^{(a)}$		$p^{(b)}$		$p^{(c)}$	
Distance	iLQG[m]	D3ROC[m]	iLQG[m]	D3ROC[m]	iLQG[m]	D3ROC[m]
Mean	**0.37**	**0.25**	**0.36**	**0.17**	**0.33**	**0.28**
Std	0.14	0.08	0.11	0.09	0.15	0.10

TABLE I: The mean distance between the final position of the robot and the origin, along with its standard deviation, for multiple runs under iLQG and D3ROC. The experiments were conducted using three different true noise distributions.

V. Conclusions

In conclusion, this paper presents D3ROC, a data-driven approach that overcomes the limitation of traditional DROC methods requiring known ambiguity sets for noise distribution. We evaluate our approach through a navigation problem for a car-like robot with unknown noise distributions. The numerical results demonstrate that D3ROC achieves smaller mean final distances between the robot and the origin, along with lower corresponding standard deviations, compared to iLQG. Additionally, the risk-averse behavior of D3ROC enables the robot to tend to avoid regions with higher noise variance, whereas iLQG leads to paths passing through such regions. Furthermore, our approach proves effective in handling various noise distributions. Overall, D3ROC offers a promising solution to real-world DROC problems where noise distribution and KL divergence bound are unknown, making the DROC framework more practical and applicable.

[1] https://github.com/ruiiu/DROC_Variance_Formula

References

[1] *OPTIMAL CONTROL OF DISCRETE-TIME SYSTEMS*, ch. 2, pp. 19–109. 2012.

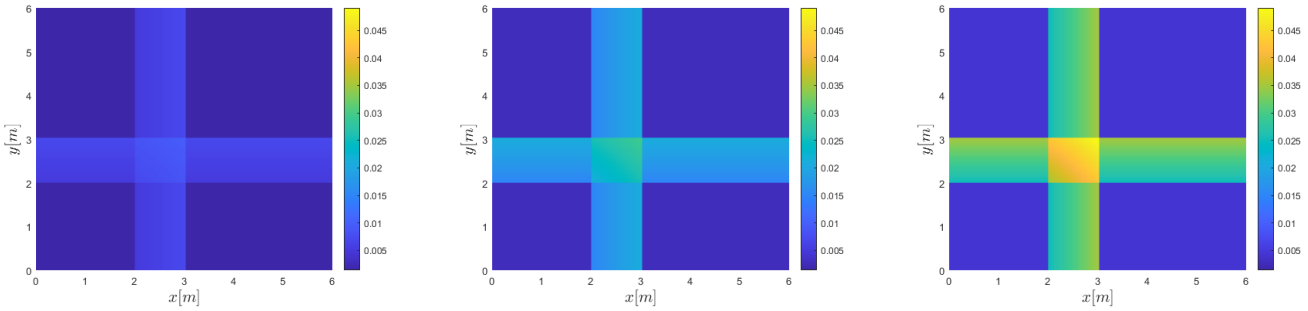
[2] M. Sanjari and H. Karami, "Optimal control strategy of battery-integrated energy system considering load demand uncertainty," *Energy*, vol. 210, p. 118525, 2020.

[3] A. Mesbah, "Stochastic model predictive control with active uncertainty learning: A survey on dual control," *Annual Reviews in Control*, vol. 45, pp. 107–117, 2018.

[4] S. Tang, "Dynamic programming for general linear quadratic optimal stochastic control with random coefficients," *SIAM Journal on Control and Optimization*, vol. 53, no. 2, pp. 1082–1106, 2015.

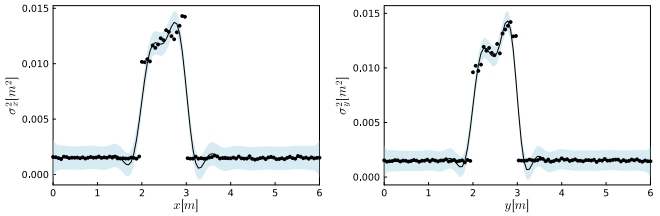
[5] G. Lee, S. S. Srinivasa, and M. T. Mason, "Gp-ilqg: Data-driven robust optimal control for uncertain nonlinear dynamical systems," *arXiv preprint arXiv:1705.05344*, 2017.

[6] D. Wang, D. Liu, H. Li, B. Luo, and H. Ma, "An approximate optimal control approach for robust stabilization of a class of discrete-time nonlinear systems with uncertainties," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 5, pp. 713–717, 2015.

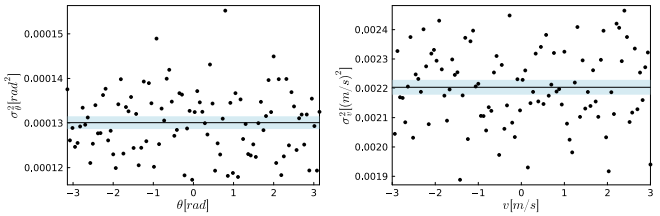


(a) The first component (b) The second component (c) The third component

Fig. 1: The heatmap of variances of each component of $p^{(b)}$ for the x and y coordinates.



(a) GP prediction for the variance of the x coordinate (b) GP prediction for the variance of the y coordinate



(c) GP prediction for the variance of the θ coordinate (d) GP prediction for the variance of the v coordinate

Fig. 2: Visualization of the GP used to predict variances for $p^{(b)}$, based on 100 data points, with black dots representing the variances calculated using MLE from observed data, the line depicting the fitted mean, and the shaded area indicating the 95% confidence interval.

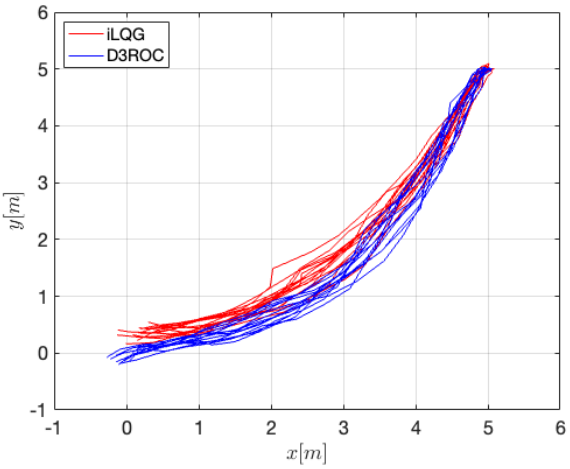


Fig. 3: Comparison of a car-like robot navigating to the origin under unknown true noise $p^{(b)}$ with D3ROC and iLQG. The starting position is $(5.0, 5.0)$ and the goal position is $(0.0, 0.0)$. The results of multiple runs are shown in the figure, where the red lines and blue lines represent different paths of the robot under iLQG and D3ROC, respectively.

[7] H. Nishimura, N. Mehr, A. Gaidon, and M. Schwager, "Rat ilqr: A risk auto-tuning controller to optimally account for stochastic model mismatch," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 763–770, 2021.

[8] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3289–3304, 2021.

[9] H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review," *arXiv preprint arXiv:1908.05659*, 2019.

[10] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations research*, vol. 58, no. 3, pp. 595–612, 2010.

[11] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[12] B. Kouvaritakis and M. Cannon, "Model predictive control," *Switzerland: Springer International Publishing*, vol. 38, 2016.

[13] Z. Hu and L. J. Hong, "Kullback-leibler divergence constrained distributionally robust optimization," *Available at Optimization Online*, vol. 1, no. 2, p. 9, 2013.

[14] Z. Li, W. Wu, B. Zhang, and X. Tai, "Kullback–leibler divergence-based distributionally robust optimisation model for heat pump day-ahead operational schedule to improve pv integration," *IET Generation, Transmission & Distribution*, vol. 12, no. 13, pp. 3136–3144, 2018.

[15] J. Duchi and H. Namkoong, "Learning models with uniform performance via distributionally robust optimization," *arXiv preprint arXiv:1810.08750*, 2018.

[16] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018.

[17] Q. Wang, S. R. Kulkarni, and S. Verdú, "Divergence estimation for multidimensional densities via k-nearest-neighbor distances," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2392–2405, 2009.

[18] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1168–1175, IEEE, 2014.

[19] S. Bechtle, Y. Lin, A. Rai, L. Righetti, and F. Meier, "Curious ilqr: Resolving uncertainty in model-based rl," in *Conference on Robot Learning*, pp. 162–171, PMLR, 2020.

[20] D. Nass, B. Belousov, and J. Peters, "Entropic risk measure in policy search," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1101–1106, IEEE, 2019.

[21] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," *Robot motion planning and control*, pp. 171–253, 2005.

[22] M. Athans, "The role and use of the stochastic linear-quadratic-gaussian problem in control system design," *IEEE transactions on automatic control*, vol. 16, no. 6, pp. 529–552, 1971.