Kernel Embedding for Particle Gibbs-Based Optimal Control

Scientific thesis for the procurance of the degree M.Sc.
from the TUM School of Computation, Information and Technology
at the Technical University of Munich.

Supervised by Univ.-Prof. Dr.-Ing. Sandra Hirche
M.Sc. Robert Lefringhausen
Chair of Information-Oriented Control

Submitted by cand. ing. Lukas Hochschwarzer
l.hochschwarzer@gmail.com

Submitted on Munich, 06.11.2024

TECHNISCHE UNIVERSITÄT MÜNCHEN
LEHRSTUHL FÜR INFORMATIONSTECHNISCHE REGELUNG
ORDINARIA: UNIV.-PROF. DR.-ING. SANDRA HIRCHE

April 26, 2024

M A S T E R ' S T H E S I S
for
Lukas Sebastian Hochschwarzer
Degree Electrical Engineering and Information Technology

Kernel Embedding for Particle Gibbs-Based Optimal Control

Problem description:

Bayesian learning-based control approaches are promising for safety-critical systems where physical modeling is time-consuming or impossible. A common problem in such systems is that not all states can be measured, resulting in the absence of a closed-form expression for the posterior distribution. However, particle Markov chain Monte Carlo (PMCMC) methods like particle Gibbs sampling can draw samples from the posterior distribution. These samples can be used to formulate a scenario optimal control problem (OCP), for whose solution probabilistic constraint satisfaction guarantees can be inferred [1]. However, representing the unknown dynamics using samples is exceedingly inefficient because many samples are required for a good representation. In addition, the scenario OCP must be solved repeatedly to infer probabilistic guarantees.
Kernel embedding is a promising alternative that allows the representation of unknown distributions with few samples, even if their parametric form is unknown. In recent years, this idea has been increasingly used for stochastic optimal control, among others, in [2] and [3]. A major advantage of these approaches is that the desired robustness level can be specified a priori and does not have to be determined by repeatedly solving the OCP.
This thesis thus aims to implement an optimal control approach that combines PMCMC methods for system identification with kernel embedding. Furthermore, the robustness of the proposed approach shall be analyzed, and the resulting algorithm shall be evaluated using simulations.

Tasks:

- Literature research on kernel embedding and PMCMC-based control
- Implementation of an optimal control approach that utilizes kernel embedding
- Robustness analysis of the proposed approach
- Numerical evaluation of the proposed approach

Bibliography:

[1] R. Lefringhausen, S. Srithasan, A. Lederer, and S. Hirche, "Learning-based optimal control with performance guarantees for unknown systems with latent states," *arXiv preprint*, 2023.
[2] J.-J. Zhu, W. Jitkrittum, M. Diehl, and B. Schölkopf, "Kernel distributionally robust optimization: Generalized duality theorem and stochastic approximation," in *International Conference on Artificial Intelligence and Statistics*, pp. 280–288, PMLR, 2021.
[3] A. Thorpe, T. Lew, M. Oishi, and M. Pavone, "Data-driven chance constrained control using kernel distribution embeddings," in *Learning for Dynamics and Control Conference*, pp. 790–802, PMLR, 2022.

Supervisor:	M.Sc. Robert Lefringhausen
Start:	06.05.2024
Delivery:	05.11.2024

(S. Hirche)

Abstract

As control engineering methods are applied to increasingly complex systems, data-driven approaches for system identification appear as a promising alternative to physics-based modeling. However, in many of these system the states cannot be measured, which complicates the learning process and makes it difficult to quantify the epistemic and aleatory uncertainties. Particle Markov chain Monte Carlo (PMCMC) methods provide ways to jointly estimate the dynamics and latent states while rigorously quantifying uncertainties. Through this, samples can be drawn from the posterior distribution. Combining PMCMC methods with scenario theory enables the computation of optimal input trajectories for unknown systems with latent states. However, deriving guarantees for these input trajectories is computationally complex.

This thesis proposes a novel method that combines PMCMC sampling with maximum mean discrepancy ambiguity sets as an alternative to scenario theory. The goal is to optimize the trajectory of a system while satisfying chance constraints that account for the arising uncertainties. The PMCMC samples are used to estimate the distribution, and the ambiguity set is constructed around that estimate to account for uncertainty due to the limited data. By using ambiguity sets, it is possible to optimize the problem over a set of distributions that we can obtain through our samples rather than optimizing over a single unknown distribution. This allows to find a robust solution for the problem and, in contrast to the scenario theory, also allows for the tuning of the robustness level in the optimization process. This is advantageous since it can lead to a reduction of the costs in exchange for a higher risk of the solution violating the constraints. The effectiveness of this approach is demonstrated in numerical simulations.

Contents

Chapter 1

Introduction

In the realm of model-based control, the development of reliable and accurate mathematical models is crucial to ensuring the success of any control application. However, finding such a model based on physical knowledge is often very time-consuming or even impossible. Because of this, data-driven control approaches that allow the derivation of such models based on previously collected data are gaining attention, and their usefulness for optimal control applications is being explored. For safety-critical applications, it is especially important to be able to account for uncertainties that result from basing the model on a limited set of training data (epistemic uncertainty) and the inherent randomness of the system, such as measurement noise (aleatory uncertainty). To this end, there are already numerous methods, such as a combination of state-space models and Gaussian processes [WR06] or similar Bayesian approaches, that allow for reliable quantifications of uncertain models.

These approaches do, however, require full-state measurements of the unknown system in order to learn the model, quantify uncertainties, and make predictions, which is often not possible as these measurements are not available in numerous practical applications. In many applications, it is unknown which variables directly affect the system, or some states are not measurable. While methods such as NARX approaches [MLF21] exist that do not require the state space representation, they come with the disadvantage of it not being clear how to include the underlying physics of the model in the learning process. Having no knowledge of the states can also lead to these methods being unable to distinguish between the noise that permanently affects the system (process noise) and the noise that only affects one specific measurement (measurement noise).

This means it is often advantageous to work with state-space models and, in the case of latent states, overcome the reliance on state measurements by jointly estimating the unknown dynamics and the latent states to create a reliable model for the system. This idea is already being utilized for optimal control in [LSLH24] to estimate the uncertain elements of the system and use the acquired data combined with scenario theory [GC22] to obtain a robust solution. However, with this approach, the performance guarantees, i.e., a lower bound for the probability that

the solution is feasible, have to be determined retroactively with a computationally complex process.
In this thesis, we propose how the use of kernel embeddings [NKSZ22] as a way to reformulate the chance-constrained problem by using samples drawn from a distribution that is generally analytically intractable. This is possible by using particle Markov chain Monte Carlo (PMCMC) methods which allow us to jointly estimate dynamics and latent states. This enables us to include the allowed risk factor in the optimization process as a way to set performance guarantees in advance.
In this chapter, the problem is introduced in Section 1.1, and related work is discussed in Section 1.2. Finally, a quick overview of the structure of the rest of this thesis is given in Section 1.3.

1.1 Problem Statement

Consider the general nonlinear discrete-time system of the form

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right) + \boldsymbol{v}_t \tag{1.1a}$$

$$\boldsymbol{y}_t = \boldsymbol{g}\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right) + \boldsymbol{w}_t \tag{1.1b}$$

with the state $\boldsymbol{x}_t \in \mathbb{R}^{n_x \in \mathbb{N}}$, the input $\boldsymbol{u}_t \in \mathbb{R}^{n_u \in \mathbb{N}}$, the output $\boldsymbol{y}_t \in \mathbb{R}^{n_y \in \mathbb{N}}$, the process noise $\boldsymbol{v}_t \in \mathbb{R}^{n_x}$, the measurement noise $\boldsymbol{w}_t \in \mathbb{R}^{n_y}$ and time $t \in \mathbb{Z}$.
In our setting, only the output $\boldsymbol{y}_t$ is observed and the state transition function $\boldsymbol{f}(\cdot)$ and the observation function $\boldsymbol{g}(\cdot)$, as well as the distributions $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{W}}$ of the process noise $\boldsymbol{v}_t$ and measurement noise $\boldsymbol{w}_t$ are unknown.
We assume that a dataset $\mathbb{D} = \{\boldsymbol{u}_t, \boldsymbol{y}_t\}_{t=\text{-}T:\text{-}1}$ containing the last $T \in \mathbb{N}$ measurements of the input $\boldsymbol{u}_t$ and output $\boldsymbol{y}_t$ is available.
We further assume that the structure of the model $\{\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot), \boldsymbol{g}_{\boldsymbol{\theta}}(\cdot), \boldsymbol{\mathcal{V}}_{\boldsymbol{\theta}}, \boldsymbol{\mathcal{W}}_{\boldsymbol{\theta}}\}$ is known, for example from a physical insight into the system, and is dependent on a finite number of unknown parameters $\boldsymbol{\theta}$. In addition to that, the priors $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x}_{\text{-}T})$ are available as well.
The objective is to minimize a given cost function

$$J_H = \sum_{t=0}^{H} c(\boldsymbol{u}_t, \boldsymbol{x}_t, \boldsymbol{y}_t) \tag{1.2}$$

over the horizon H while satisfying the constraints

$$\boldsymbol{h}(\boldsymbol{u}_{0:H}, \boldsymbol{x}_{0:H}, \boldsymbol{y}_{0:H}) \leq \boldsymbol{0} \tag{1.3}$$

with $\boldsymbol{h} \in \mathbb{R}^{n_c}$ being a vector of arbitrary deterministic function. As the states $\boldsymbol{x}_{0:H}$ are unknown and there are several uncertain factors in our system, i.e., the process and measurement noise and the unknown parameter $\boldsymbol{\theta}$ that characterizes the system, the constraints are transformed into chance-constraints where only a portion of the

possible cases have to satisfy the constraints. This is done due to the possibility that $\boldsymbol{h}$ is impossible to satisfy for every possible $\boldsymbol{x}_{0:H}$. For this, we also introduce a risk factor $\alpha \in [0, 1]$ that relaxes the constraints, turning them into

$$P\left[\max(\boldsymbol{h}(\boldsymbol{u}_{0:H}, \boldsymbol{x}_{0:H}, \boldsymbol{y}_{0:H})) \leq 0\right] \geq 1 - \alpha \tag{1.4}$$

with the underlying distribution of the data being generally unknown.

1.2 Related Work

The problem presented in Section 1.1 provides several challenges as the available information is very limited. While many methods to solve chance-constrained problems exist, they often rely on knowledge of the posterior distribution, which is generally unknown in our problem. While we do have priors for the uncertain elements, we only have access to the prior for $\boldsymbol{x}_{\text{-}T}$ and the forward propagation over T timesteps will generally lead to the optimization problem becoming infeasible due to the high degree of uncertainty. As such, the priors must be updated based on the input-output measurements $\mathbb{D}$, i.e., the posterior distribution must be inferred, but this distribution is analytically intractable [ADH10]. To draw samples from the posterior distribution, particle Markov chain Monte Carlo (PMCMC) methods were proposed [ADH10].
This has recently been exploited for optimal control in [LSLH24], utilizing such a sampler to generate scenarios that describe possible future trajectories of the unknown system. These scenarios are then used to formulate a deterministic optimal control problem by reformulating the chance constraints with the scenarios [GC22]. However, this usage of the scenarios comes with the drawback that the risk factor cannot be specified for the final optimal control problem (OCP), and the process of estimating it retroactively is resource-intensive.
As such, there is a need to find other methods that allow us to utilize the samples generated by the PMCMC sampler to reformulate the chance constraints to find a distributionally robust solution without losing the risk factor in the process.
As the difficulties with this can be traced back to the unknown distribution, kernel distribution embeddings have been proposed in [TO21] and [TLOZ22] to reformulate chance-constrained control optimal control problems. However, these approaches work under the assumption that the states are known as the transition function is embedded directly. As such, this approach is unsuited for systems with latent states.
Another workaround that has been proposed is the use of ambiguity sets. Here, ambiguity sets are defined as a set of probability distributions that are within a certain radius under an appropriate distance metric. For this purpose, Wasserstein distance was proposed as a metric for the ambiguity set in [HCL19]. It has, however, been proven rather difficult to efficiently construct a Wasserstein ambiguity set for problems with works limiting themselves to affine constraint functions.

Another metric is proposed in [NKSZ22] allows for an efficient construction of an ambiguity set using a maximum mean discrepancy (MMD) metric combined with kernel approximation. In contrast to Wasserstein ambiguity sets, this approach can be applied to general nonlinear and nonconvex constraints.
In this thesis, MMD ambiguity sets are combined with PMCMC sampling methods to solve an OCP. In contrast to the scenario theory, this allows for the reformulation of the chance constraints while including the risk factor as a way to set performance guarantees before solving the problem.

1.3 Structure of this Thesis

The remainder of this paper is structured as follows. In Chapter 2, we review the methods used to create a PMCMC sampler and how the OCP is reformulated with scenario theory. Following that, we describe the alternative approach using ambiguity sets in Chapter 3. These methods are then tested and evaluated in Chapter 4. Finally, the results are summarized, and some concluding remarks are given in Chapter 5.

Chapter 2

Technical Background

In this chapter, an approach is outlined that allows us to effectively draw samples from the unknown distribution defined in Section 1.1 using PMCMC methods. In Section 2.1, we explain how to draw samples from the posterior distribution of model parameters and state trajectories and how to use them to generate scenarios. An existing method that uses the scenarios to formulate an optimal control problem to find a robust solution is then described in Section 2.2, and the disadvantage of this approach is highlighted.

2.1 Particle Markov Chain Monte Carlo Methods

For practical applications, the known priors $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x}_{\text{-}T})$ and the observations $\mathbb{D}$ must be used to infer the posterior $p(\boldsymbol{\theta}, \boldsymbol{x}_{\text{-}T:\text{-}1} \mid \mathbb{D})$. While this is not the true distribution $p(\boldsymbol{\theta}, \boldsymbol{x}_{\text{-}T:\text{-}1})$, it is necessary since the repeated propagation of $p(\boldsymbol{x}_{\text{-}T})$ would otherwise cause an excessively large variance in $p(\boldsymbol{x}_{\text{-}1})$ making stochastic OCP infeasible. We utilize PMCMC methods to draw samples from the posterior distribution. These methods were introduced in [ADH10] and will be summarized in this section.

We use Particle Gibbs (PG) to bypass the issue of an analytically intractable posterior distribution $p(\boldsymbol{\theta}, \boldsymbol{x}_{\text{-}T:\text{-}1} \mid \mathbb{D})$ by iteratively drawing samples from $p(\boldsymbol{\theta} \mid \boldsymbol{x}_{\text{-}T:\text{-}1}, \mathbb{D})$ and $p(\boldsymbol{x}_{\text{-}T:\text{-}1} \mid \boldsymbol{\theta}, \mathbb{D})$. We continually update the distributions with the previously drawn set, i.e., $\boldsymbol{x}_{\text{-}T:\text{-}1}^{[n]}$ is drawn from $p(\boldsymbol{x}_{\text{-}T:\text{-}1}^{[n]} \mid \boldsymbol{\theta}^{[n]}, \mathbb{D})$ and $\boldsymbol{\theta}^{[n+1]}$ is then drawn from $p(\boldsymbol{\theta}^{[n+1]} \mid \boldsymbol{x}_{\text{-}T:\text{-}1}^{[n]}, \mathbb{D})$. This is repeated until the desired number of samples has been achieved.

To ensure that the samples drawn through this method are an accurate representation of the distribution $p(\boldsymbol{\theta}, \boldsymbol{x}_{\text{-}T:\text{-}1})$, additional steps are taken. For one, the first N_p samples must be discarded as they are heavily reliant on the initialization and might show a strong bias. This burn-in period should be chosen large enough that this bias is no longer reflected in the samples. The samples should also be independent of each other which is not given with this method as each $\boldsymbol{\theta}^{[n]}$ is dependent on $\boldsymbol{x}_{\text{-}T:\text{-}1}^{[n]}$

Algorithm 1 Scenario generation

Input: Dataset $\mathbb{D}$, parametric model $\{\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot), \boldsymbol{g}_{\boldsymbol{\theta}}(\cdot), \boldsymbol{\mathcal{V}}_{\boldsymbol{\theta}}, \boldsymbol{\mathcal{W}}_{\boldsymbol{\theta}}\}$, priors $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x}_{\text{-}T})$, N, H, T

Output: Scenarios $\boldsymbol{\delta}^{[1:N]} = \{\boldsymbol{\theta}, \boldsymbol{x}_0, \boldsymbol{v}_{0:H}, \boldsymbol{w}_{0:H}\}^{[1:N]}$

1: **for** $n = 1, \ldots, N$ **do**
2: Sample $\{\boldsymbol{\theta}, \boldsymbol{x}_{\text{-}T:\text{-}1}\}^{[n]}$ from $p\left(\boldsymbol{\theta}, \boldsymbol{x}_{\text{-}T:\text{-}1} \mid \mathbb{D}\right)$ using a PG sampler
3: **for** $t = \text{-}1, \ldots, H$ **do**
4: Sample $\boldsymbol{v}_t^{[n]}$ from $\boldsymbol{\mathcal{V}}_{\boldsymbol{\theta}^{[n]}}$
5: Sample $\boldsymbol{w}_t^{[n]}$ from $\boldsymbol{\mathcal{W}}_{\boldsymbol{\theta}^{[n]}}$
6: **end for**
7: $\boldsymbol{x}_0^{[n]} \leftarrow \boldsymbol{f}_{\boldsymbol{\theta}^{[n]}}\left(\boldsymbol{x}_{\text{-}1}^{[n]}, \boldsymbol{u}_{\text{-}1}\right) + \boldsymbol{v}_{\text{-}1}^{[n]}$
8: **end for**

which in turn is dependent on $\boldsymbol{\theta}^{[n-1]}$. As such, measures must be taken to reduce the correlation between samples as much as possible. One approach to do this is thinning, where only every n_d-th sample is used, and the other samples are discarded. By increasing this parameter, the samples become more uncorrelated, but there will also be a larger amount of samples created, which leads to inefficiency.
The samples $\{\boldsymbol{\theta}, \boldsymbol{x}_{\text{-}T:\text{-}1}\}^{[1:N]}$ can be used to generate so-called scenarios $\boldsymbol{\delta}^{[1:N]}$, which are samples from the distribution $p(\boldsymbol{\theta}, \boldsymbol{x}_0, \boldsymbol{v}_{0:H}, \boldsymbol{w}_{0:H} \mid \mathbb{D})$ and represent possible future system behavior depending on $\boldsymbol{u}_{0:H}$. The generation of these scenarios is outlined in Algorithm 1. The parameters $\boldsymbol{\theta}^{[n]}$ are obtained via PMCMC, and through it we also know the system dynamics and noise distributions which can be used to draw samples of both the processing noise $\boldsymbol{v}_{0:H}$ and measurement noise $\boldsymbol{w}_{0:H}$, which can be seen in the lines 4 and 5 of the algorithm. Those samples can then be combined with the $\boldsymbol{x}_{\text{-}T:\text{-}1}$, or more precisely $\boldsymbol{x}_{\text{-}1}$ to find the initial state $\boldsymbol{x}_0$ to complete the scenario $\boldsymbol{\delta} = \{\boldsymbol{\theta}, \boldsymbol{x}_0, \boldsymbol{v}_{0:H}, \boldsymbol{w}_{0:H}\}$. How these scenarios can be used to find an optimal input $\boldsymbol{u}_{0:H}$ is described in the next section.

2.2 Scenario Approach

In the previous section, we have generated the scenarios $\boldsymbol{\delta}^{[1:N]} = \{\boldsymbol{\theta}, \boldsymbol{x}_0, \boldsymbol{v}_{0:H}, \boldsymbol{w}_{0:H}\}^{[1:N]}$. Each scenario can be used to describe a possible future trajectory based on the input $\boldsymbol{u}_{0:H}$. As such, they can be used to reformulate the chance constraints to ensure a robust solution. We do this by ensuring that the constraints $\boldsymbol{h}(\cdot)$ are satisfied for every scenario, i.e., every possible known future satisfies the constraints [GC22]. As such, the constraints can be written as

$$\max(\boldsymbol{h}(\boldsymbol{u}_{0:H}, \boldsymbol{x}_{0:H}^{[n]}, \boldsymbol{y}_{0:H}^{[n]})) \leq 0, \ \forall n = 1, ..., N. \tag{2.1}$$

This approach is effective at finding a robust solution. However, the constraints no longer contain the risk factor α, which means that determining performance

guarantees requires further calculations. This is a computationally complex process where subsets of the scenarios $\boldsymbol{\delta}^{[1:N]}$ have to be tested to find the smallest possible subset of scenarios that still provides the same solution, for example, through a greedy algorithm [GC21]. The cardinality of this subset can then be used to compute a guarantee. On top of that, it is also impossible to control the risk factor without decreasing or increasing the number of scenarios used in the optimization, and for certain applications, the problem might become infeasible for a high number of scenarios as there is no way to relax the constraints. As such, there is a need for an alternative approach.

Chapter 3

Chance-Constraint Optimization with Kernel Approximation

In this chapter, an approach is outlined that allows us to effectively reformulate the chance-constraint problem defined in Section 1.1 using the samples generated using the method described in Chapter 2. In Section 3.1, a method to construct a maximum mean discrepancy (MMD) ambiguity sets with kernel approximation is proposed, and in Section 3.2, the ambiguity set is used to reformulate the chance constraints from Section 1.1. A method for optimizing the hyperparameters for this approximation is explained in Section 3.3. Finally, in Section 3.4, the results are used to define a new OCP.

3.1 MMD ambiguity sets

As the underlying distribution of the data in the constraints (1.3) is unknown, we aim to expand them to their distributionally robust counterpart in order to allow for the use of scenarios as an approximation of the distribution. For this, we consider $\tilde{P}$ as the worst case distribution within a set $\mathcal{P}$ of plausible distributions, the so-called ambiguity set. This gives us the new constraints

$$\inf_{\tilde{P}\in\mathcal{P}} \tilde{P}\left[\max(\boldsymbol{h}(\boldsymbol{u}_{0:H}, \boldsymbol{x}_{0:H}, \boldsymbol{y}_{0:H})) \leq 0\right] \geq 1-\alpha. \tag{3.1}$$

To construct the set $\mathcal{P}$, a similarity measure is needed to provide a concrete comparison between various distributions $\tilde{P}$. The Wasserstein metric, a common distance function between probability distributions, has been tried for this purpose in the past. However, most of these works, such as [HCL19], limit the class of constraints to affine functions. As such, this approach is not suitable for our general nonlinear constraints. In contrast, maximum mean discrepancy [GBR+12] is able to be applied to nonlinear and non-convex constraints.
We define the positive definite kernel as a symmetric function $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ where $\sum_{i,j=1}^{n} a_i a_j k(z_i, z_j) \geq 0$ for any $n \in \mathbb{N}$, $\{z_i\}_{i=1}^{n} \subset \mathcal{Z}$ and $\{a_i\}_{i=1}^{n} \subset \mathbb{R}$. For

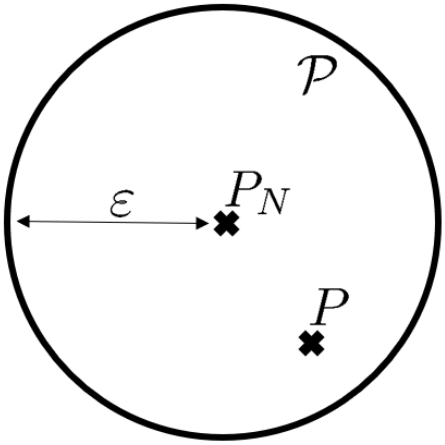


Figure 3.1: Ambiguity Set $\mathcal{P}$

every positive definite kernel, there also exists a feature map $\phi : \mathcal{Z} \to \mathcal{H}$ such that $k(z, z') = \langle \phi(z), \phi(z') \rangle_{\mathcal{H}}$ with $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denoting the inner product of a reproducing kernel Hilbert space $\mathcal{H}$. This inner product can be used to define a norm as $||f||_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$ [MFSS17].
With this, we use the norm of the difference between the kernel mean embeddings (KME) $||\mu_Q - \mu_{Q'}||^2_{\mathcal{H}}$ of two distributions Q and Q' as a metric between two distributions with the KME being defined as $\mu_Q = \int \phi(z) \mathrm{d}z$ [GBR+12]. Utilizing the reproducing property of an RKHS, the metric can then be rewritten as

$$\mathrm{MMD}(Q, Q') = \mathrm{E}_{z,z' \sim Q}[k(z, z')] + \mathrm{E}_{z,z' \sim Q'}[k(z, z')] - 2\mathrm{E}_{z \sim Q, z' \sim Q'}[k(z, z')]. \tag{3.2}$$

The MMD-based ambiguity set $\mathcal{P}$ is then constructed as the set of distributions $\tilde{P}$ in an ε radius centered around the empirical distribution P_N which is given through the scenarios $\boldsymbol{\delta}^{[1:N]}$. This gives us the set

$$\mathcal{P} = \{P : \mathrm{MMD}(P, P_N) \leq \varepsilon\}. \tag{3.3}$$

with the empirical distribution being defined as

$$P_N = \frac{1}{N} \sum_{n=1}^{N} d_{\boldsymbol{\delta}^{[n]}} \tag{3.4}$$

with $d_{\boldsymbol{\delta}}$ being a dirac impulse at the point of the sample $\boldsymbol{\delta}$.
As illustrated in Figure 3.1, this allows us to build a set near the unknown distribution P. As the distribution P_N is built based on the available samples, it represents our best guess at what P looks like. We can assume that for $N \to \infty$, P_N converges towards P, but as we only have a limited number of samples available, we also include everything within an ε radius around P_N. We set this radius by constructing a

Algorithm 2 Bootstrap MMD ambiguity set

Input: Scenarios $\boldsymbol{\delta}^{[1:N]}$, Number of bootstrap samples B, Confidence level β
Output: Gram matrix $\boldsymbol{K}$, Radius of MMD ambiguity set ε

$\boldsymbol{K} \leftarrow \mathit{kernel}(\boldsymbol{\delta}, \boldsymbol{\delta})$
for $m = 1, \ldots, B$ **do**
  $I \leftarrow N$ numbers from $\{1, \ldots N\}$ with replacement
  $K_x \leftarrow \sum_{i,j=1}^{N} K_{ij}$;
  $K_y \leftarrow \sum_{i,j \in I} K_{ij}$;
  $K_{xy} \leftarrow \sum_{j \in I} \sum_{i=1}^{N} K_{ij}$;
  $\mathrm{MMD}[m] \leftarrow \frac{1}{N^2} \left(K_x + K_y - 2K_{xy}\right)$;
end for
$\mathrm{MMD} \leftarrow \mathit{sort}(\mathrm{MMD})$
$\varepsilon \leftarrow \mathrm{MMD}[\mathit{ceil}(B\beta)]$

bootstrap MMD ambiguity set as described in [NKSZ22], which is outlined in Algorithm 2. This procedure requires a number of bootstrap samples B to be chosen and a confidence level β. It then utilizes kernels $k(\boldsymbol{\delta}^{[i]}, \boldsymbol{\delta}^{[j]}) \in \mathbb{R}$ to define the (biased) MMD estimator as

$$\widehat{\mathrm{MMD}}(\tilde{P}, P_N) = \sum_{i,j=1}^{N} k(\boldsymbol{\delta}^{[i]}, \boldsymbol{\delta}^{[j]}) + k(\tilde{\boldsymbol{\delta}}^{[i]}, \tilde{\boldsymbol{\delta}}^{[j]}) - 2k(\boldsymbol{\delta}^{[i]}, \tilde{\boldsymbol{\delta}}^{[j]}) \tag{3.5}$$

with $\tilde{\boldsymbol{\delta}}^{[n]}, n = 1, ..., N$, denoting a bootstrap sample of P_N where samples are drawn with replacement from $\boldsymbol{\delta}^{[1:N]}$. This is done to give us an idea of what the empirical distribution might look like with a different set of samples. This can give us insight into how precise our estimate P_N is and what kind of radius ε should be included in the ambiguity set. We do this by calculating $\widehat{\mathrm{MMD}}(\tilde{P}, P_N)$ for all B bootstrap samples. The results are saved in a list, and ε is chosen as the $\mathit{ceil}(B\beta)$-th element of the sorted list. This means we construct the ambiguity set so that $B\beta$ of our bootstrap samples are included while the rest are either outside or on the edge of the ambiguity set.

3.2 Constraint Reformulation

With a given MMD ambiguity set $\mathcal{P}$, the feasible set of the constraint (1.4) is given as

$$Z := \left\{ \boldsymbol{u}_{0:H} \in \mathcal{U}^{H+1} : \inf_{\tilde{P} \in \mathcal{P}} \tilde{P} \left[\tilde{h}(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}) \leq 0 \right] \geq 1 - \alpha \right\}. \tag{3.6}$$

with $\tilde{h}(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}) = \max(\boldsymbol{h}(\boldsymbol{u}_{0:H}, \boldsymbol{x}_{0:H}, \boldsymbol{y}_{0:H}))$.

We can now use the ambiguity set $\mathcal{P}$ defined in Section 3.1 with the radius ε to reformulate the feasible set. We are able to reformulate this set using the strong duality proven in [ZJDS20] to get the feasible region

$$Z \coloneqq \left\{ \boldsymbol{u}_{0:H} \in \mathcal{U}^{H+1} : \begin{array}{ll} g_0 + \frac{1}{N}\sum_{n=1}^{N} g(\boldsymbol{\delta}^{[n]}) + \varepsilon||g||_{\mathcal{H}} \leq \alpha & (3.7a) \\ 1(\tilde{h}(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}^{[n]}) > 0) \leq g_0 + g(\boldsymbol{\delta}^{[n]}),\ n = 1, ..., N & (3.7b) \\ g_0 \in \mathbb{R}, g \in \mathcal{H} & (3.7c) \end{array} \right\}$$

with reproducing kernel hilbert space function $g \in \mathcal{H}$ and $1(\cdot)$ denoting the indicator function. Using this set feasible region exactly is generally intractable in practice. As such, we further approximate the set based on conditional Value-at-Risk (CVaR). For this, we first define the Value-at-Risk (VaR), which is given as

$$\mathrm{VaR}^{P}_{1-\alpha} := \inf \left\{ t' \in \mathbb{R} : P\left[\tilde{h}(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}) \leq t'\right] \geq 1 - \alpha \right\}. \tag{3.8}$$

This allows us to relax the constraint as it can be easily shown that

$$\mathrm{VaR}^{P}_{1-\alpha} \leq 0 \iff P\left[\tilde{h}(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}) \leq 0\right] \geq 1 - \alpha. \tag{3.9}$$

With this, we have an alternative expression of the chance constraints in terms of the VaR. While the VaR constraint is generally nonconvex, we can use the tightest conservative convex approximation given by the conditional convex approximation, which has been shown in [NS07] to be given by CVaR which is defined as

$$\mathrm{CVaR}^{P}_{1-\alpha} := \inf_{t \in \mathbb{R}} \mathrm{E}_P \left[[\tilde{h}(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}) + t']_+ - t'\alpha \right] \tag{3.10}$$

with $[\cdot]_+ = \max(0, \cdot)$ denoting the max operator. The constraints can then be expressed as

$$\mathrm{CVaR}^{P}_{1-\alpha} \leq 0. \tag{3.11}$$

However, as this constraint depends on the unknown distribution P, we use the ambiguity set defined in Section 3.1 to relax it. As such, our constraint can be rewritten as

$$\begin{aligned} \sup_{\tilde{P} \in \mathcal{P}} \mathrm{CVaR}^{\tilde{P}}_{1-\alpha} &= \sup_{\tilde{P} \in \mathcal{P}} \inf_{t \in \mathbb{R}} \mathrm{E}_{\tilde{P}} \left[[\tilde{h}(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}) + t']_+ - t'\alpha \right] & (3.12a) \\ &= \inf_{t \in \mathbb{R}} \sup_{\tilde{P} \in \mathcal{P}} \mathrm{E}_{\tilde{P}} \left[[\tilde{h}(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}) + t']_+ - t'\alpha \right] \leq 0. & (3.12b) \end{aligned}$$

Applying this to the feasible region defined in (3.7), we arrive at the approximation of the feasible region, which is given as

$$\hat{Z} \coloneqq \left\{ \begin{array}{ll} & g_0 + \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{K}\boldsymbol{\gamma})_n + \varepsilon\sqrt{\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\gamma}} \leq t'\alpha & \text{(3.13a)} \\ \boldsymbol{u}_{0:H} \in \mathcal{U}^{H+1} : & [\tilde{h}(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}^{[n]}) + t']_+ \leq g_0 + (\boldsymbol{K}\boldsymbol{\gamma})_n,\ n = 1, ..., N & \text{(3.13b)} \\ & g_0 \in \mathbb{R}, \boldsymbol{\gamma} \in \mathbb{R}^N, t' \in \mathbb{R} & \text{(3.13c)} \end{array} \right\}$$

with g being expressed as the product of the kernel Gram matrix $\boldsymbol{K}$ with the elements $[\boldsymbol{K}]_{i,j} = k(\boldsymbol{\delta}^{[i]}, \boldsymbol{\delta}^{[j]})$ and a finite dimensional vector $\boldsymbol{\gamma} \in \mathcal{R}^N$.

3.3 Kernel Approximation

In Section 3.2, the chance constraints have been reformulated kernel maximum mean discrepancy where kernels have been used as a universal function approximator based on the available samples. However, for this approximation to be reliable, appropriate kernels must be chosen to give a good estimate. Our samples also consist of multiple elements, i.e., the initial state $\boldsymbol{x}_0$ and the model parameter $\boldsymbol{\theta}$, which are of varying size and magnitude. As such, it is necessary to divide the kernels into multiple factors that are tuned separately and then multiplied.
For these seperate kernels, we use Gaussian kernels $k(\boldsymbol{z}, \boldsymbol{z}') = \exp\left(\text{-}\frac{1}{2\sigma^2}||\boldsymbol{z} - \boldsymbol{z}'||_2^2\right)$ with the bandwidth σ. As such, the elements of the Gram matrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ are defined as

$$K_{ij} = k_{\boldsymbol{\sigma}_\mathcal{X}}(\boldsymbol{x}_0^{[i]}, \boldsymbol{x}_0^{[j]}) k_{\boldsymbol{\sigma}_\Theta}(\boldsymbol{\theta}^{[i]}, \boldsymbol{\theta}^{[j]}) \tag{3.14}$$

with the kernel for the states $\boldsymbol{x}$ and the model parameter $\boldsymbol{\theta}$ being potentially subdivided again into an individual kernel for the process and measurement noise and a kernel for the unknown transition and measurement functions. The individual elements of $\boldsymbol{\sigma} = [\boldsymbol{\sigma}_\mathcal{X}, \boldsymbol{\sigma}_\Theta]$ are initialized via the median heuristic [GJK18] and then increased or decreased to maximize the average probability of a set of N_{test} test samples, i.e. find the $\boldsymbol{\sigma}$ that maximizes

$$\max_{\boldsymbol{\sigma}} \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \mathrm{P}_{\boldsymbol{\sigma}}({\boldsymbol{\delta}_{\text{test}}}^{[n]}) \tag{3.15}$$

with the probability function being the product of multiple Gaussian probability functions

$$\mathrm{P}_{\boldsymbol{\sigma}}({\boldsymbol{\delta}_{\text{test}}}^{[n]}) = \prod_{\sigma_i \in \boldsymbol{\sigma}} \frac{1}{N_{\text{train}}} \sum_{m=1}^{N_{\text{train}}} \frac{1}{\sqrt{2\pi\sigma_i^2}} k_{\sigma_i}({\boldsymbol{\delta}_{\text{train}}}^{[m]}, {\boldsymbol{\delta}_{\text{test}}}^{[n]}) \tag{3.16}$$

over a set of N_{train} training samples that are independent of the N_{test} test samples.

3.4 Problem Formulation

In the previous sections, the chance constraints have been reformulated into a feasible region $\hat{Z}$. We now want to use this to reformulate the chance-constrained OCP defined in Section 1.1.
For this purpose, we can use the variables $\boldsymbol{\gamma}$, g_0 and t' that span the feasible region as optimization parameters while making sure that $\boldsymbol{u}_{0:H}$ is an element of the feasible region.
This way, we can formulate the OCP as

$$\min_{\boldsymbol{u}_{0:H},\boldsymbol{\gamma},g_0,t'} J_H(\boldsymbol{u}_{0:H}) \tag{3.17a}$$

$$\text{s.t. } \forall n = 1, ..., N,\ \forall t = 0, 1, ..., H \tag{3.17b}$$

$$\boldsymbol{x}_{t+1}^{[n]} = \boldsymbol{f}_{\boldsymbol{\theta}^{[n]}}\left(\boldsymbol{x}_t^{[n]}, \boldsymbol{u}_t\right) + \boldsymbol{v}_t^{[n]} \tag{3.17c}$$

$$\boldsymbol{y}_t = \boldsymbol{g}_{\boldsymbol{\theta}^{[n]}}\left(\boldsymbol{x}_t^{[n]}, \boldsymbol{u}_t\right) + \boldsymbol{w}_t^{[n]} \tag{3.17d}$$

$$\boldsymbol{u}_{0:H} \in \hat{Z}(\boldsymbol{\gamma}, g_0, t') \tag{3.17e}$$

As described in Section 1.1, we are minimizing a cost function J_H. The system dynamics are included through the constraints (3.17c) and (3.17d) and must be fulfilled for all scenarios as well. Lastly, the input $u_{0:H}$ is restricted to the feasible set $\hat{Z}$ in (3.17e) which is defined by the optimization variables $\boldsymbol{\gamma}$, g_0 and t'.
The optimization problem (3.17) is deterministic and can be solved with well-known methods [NW06].

Chapter 4

Evaluation

In this section, the effectiveness of the proposed optimal control approach is tested in a simulation and compared to previously used scenario theory. The simulation setup is described in Section 4.1. The results of the OCP are shown in Section 4.2. Afterward, we analyze the results and performance in more detail in Section 4.3 by looking at the robustness of the solution and comparing it to the commonly used scenario approach. Finally, we show the potential of this approach in Section 4.4.

4.1 Simulation Setup

We consider a system with the state transition function

$$\boldsymbol{f}(\boldsymbol{x}, u) = \begin{bmatrix} 0.8x_1 - 0.5x_2 \\ 0.4x_1 + 0.5x_2 + u \end{bmatrix} \tag{4.1}$$

and the process noise distribution

$$\boldsymbol{v}_t \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 0.03 & \text{-}0.004 \\ \text{-}0.004 & 0.01 \end{bmatrix}\right). \tag{4.2}$$

Both the state transition function and the process noise distribution are unknown to the user. Meanwhile, the observation function $g(\boldsymbol{x}, u) = x_1$ and measurement noise $w_t \sim \mathcal{N}(0, 0.1)$ is assumed to be known. This assumption is made without loss of generality since an unknown observation model can be combined with the unknown transition model into an expanded model with $n_x + n_y$ states [Fri15].
For the scenario generation, we consider a set containing $T = 2000$ input and output measurements of the true system for our dataset $\mathbb{D}$. These measurements are obtained with a random input trajectory $u \sim \mathcal{N}(0, 3)$ while starting from a random initial state $\boldsymbol{x}_{\text{-}T} \sim \mathcal{N}([2, 2]^{\mathrm{T}}, \boldsymbol{I}_2)$. To infer the state model parameters, the approach from [SS17] is used. It is assumed that $\boldsymbol{f}(\cdot)$ is a linear combination of n_a basis functions $\boldsymbol{\varphi}(\boldsymbol{x}_t, u_t)$ and the process noise is normally distributed. As such, the state transition can be rewritten as

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{\varphi}(\boldsymbol{x}_t, u_t) + \boldsymbol{v}_t \tag{4.3}$$

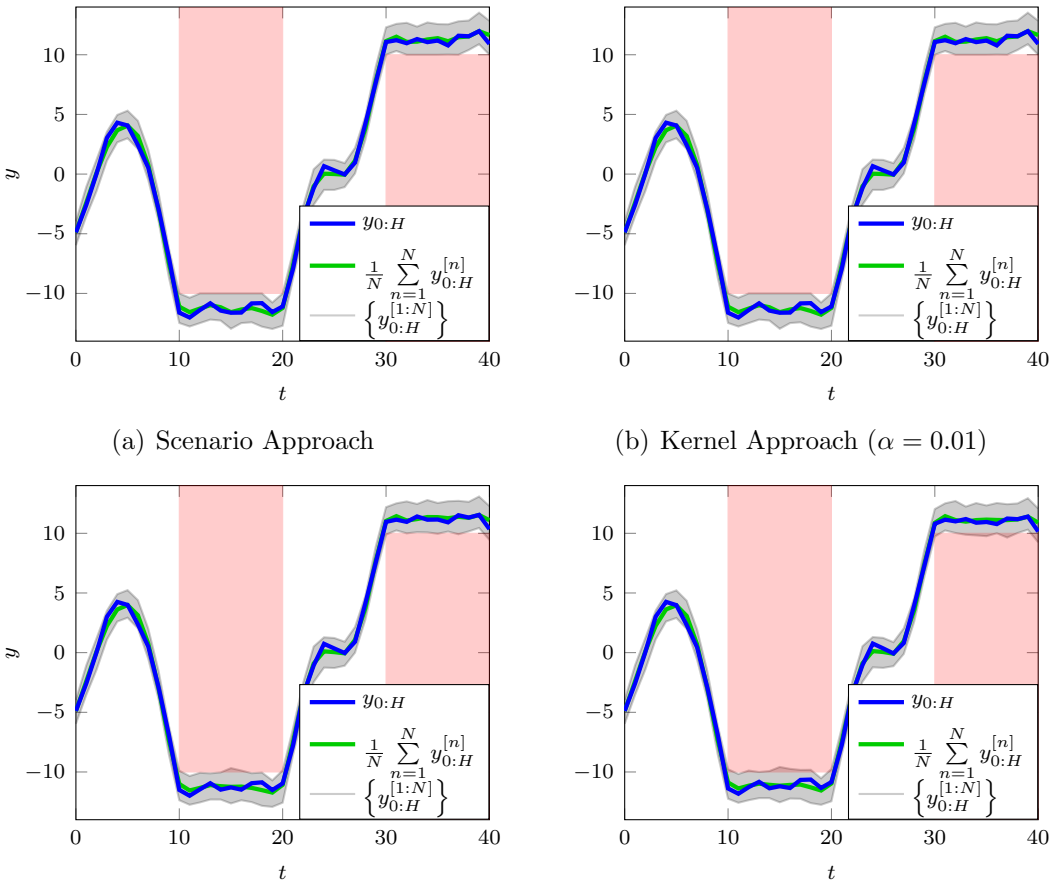
with the basis functions $\boldsymbol{\varphi}(\boldsymbol{x}, u) = [x_1, x_2, u]^{\mathrm{T}}$, the process noise $\boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and the unknown parameters $\boldsymbol{\theta}$ consisting of $\boldsymbol{A} \in \mathbb{R}^{n_x \times n_a}$ and $\boldsymbol{Q} \in \mathbb{R}^{n_x \times n_x}$. An inverse Wishart prior with l degrees of freedom and positive definite scale matrix Λ is assumed for the matrix $\boldsymbol{Q}$. For the matrix $\boldsymbol{A}$, a matrix normal prior is used. As such, the distribution is given as

$$\boldsymbol{A} = \boldsymbol{U}^{\frac{1}{2}}\boldsymbol{G}\boldsymbol{V}^{\frac{1}{2}} + \boldsymbol{M} \tag{4.4}$$

with the mean matrix $\boldsymbol{M} = \boldsymbol{0}$, the right covariance $\boldsymbol{U} = \boldsymbol{Q}$, the left covariance matrix $\boldsymbol{V} \in \mathbb{R}^{n_a \times n_a} = 10\boldsymbol{I}_3$ and the matrix $\boldsymbol{G} \in \mathbb{R}^{n_x \times n_a}$ consisting of the i.i.d. random variables $[\boldsymbol{G}]_{i,j} \in \mathcal{N}(0, 1)$. For the estimation of the posterior distribution with the PG sampler, we scale the basis vector $\boldsymbol{\varphi}$ with the weights $[0.1, 0.1, 1]^{\mathrm{T}}$, i.e. $\boldsymbol{\varphi}'(\boldsymbol{x}, u) = [0.1x_1, 0.1x_2, u]^{\mathrm{T}}$.
As described in Section 3.3, we use Gaussian kernels with the bandwidth σ being selected individually for the initial state and the elements of $\boldsymbol{\theta}$, i.e., $\boldsymbol{\sigma} = [\sigma_A, \sigma_{\mathcal{X}}, \sigma_{\mathcal{V}}, \sigma_{\mathcal{W}}]$. For the optimization of the different bandwidths, we use $N_{\text{train}} = 300$ training samples and $N_{\text{test}} = 2000$ test samples.

4.2 Optimal Control with Constrained Output

In the following, we show how well the proposed optimal control approach works when applied to an OCP with constrained output. We illustrate this by putting the solution side by side with the solution of the same problem where the scenario approach that was described in Section 2.2 was used.
For this simulation, we are using scenarios that have been generated using the PG sampler. To this end, 16130 samples are created. The first $N_p = 2000$ are discarded as training samples, and the remaining samples are thinned with $n_d = 70$. The remaining $N = 200$ samples are then used as scenarios for the OCPs.
For the cost function, we consider a simple quadratic cost $J_H = \sum_{t=0}^{H} u_t^2$ over the horizon $H = 40$. For constraints, we consider the input-constraint $|u| \leq 10$ as well as the temporarily active output-constraints $y_{10:20} \leq$ -10 and $10 \leq y_{30:40}$. The ambiguity set radius ε is chosen through Algorithm 2 with a number of bootstrap samples $B = 1000$ and a confidence level $\beta = 0.95$. In this experiment, we look at the results for the risk level α being chosen as 0.01, 0.2, and 0.5 to determine how much of an influence this parameter has on the solution.
The OCP can then be formulated as described in Section 3.2. Since the system and constraints have been chosen as linear in this example, the resulting problems for both the scenario and kernel approach are convex, and a solution can be found easily by using a convex solver. For this and the subsequent OCPs, we use the CVX to optimize the problems [BCG11].



(a) Scenario Approach

(b) Kernel Approach ($\alpha = 0.01$)

(c) Kernel Approach ($\alpha = 0.2$)

(d) Kernel Approach ($\alpha = 0.5$)

Figure 4.1: Example of the optimal control for scenario approach (top left) and kernel approach for various values of α. The red areas show the output constraints. The gray area encompasses the 200 scenarios that were used in the optimization with the green line being the average. The blue line is one realization of the true output.

The results of an exemplary run are shown in Figure 4.1. The figure includes the four plots for the scenario and kernel approach for each α and shows the output y of their respective OCPs. The graphs show the spread of the $N = 200$ trajectories that are generated when the input $\boldsymbol{u}_{0:H}$ is applied to the scenarios that were used to find the optimal input. On top of that, it also shows the mean of these trajectories and true output. Where the graphs differ, however, is to what extent the solutions fulfill the constraints. By its definition, the scenario approach requires all scenarios to fulfill the constraints that can be seen in the solution. While the gray area touches the lower and upper bounds during several timesteps, it never violates the constraints. The kernel approach, on the other hand, has a risk factor α built in, which allows for a number of scenarios to violate the constraints as long a sufficient number satisfies

them. This can be seen by the gray area following a similar trajectory to the scenario approach while a small portion of the area actively overlaps with the marked area. This is especially apparent in the plots with larger α values. In contrast, the graph where α is chosen as 0.01 is almost identical to the scenario approach. These results show that the kernel approach allows us to find solutions outside the feasible region of the scenario approach and potentially find a solution with a lower cost. This is done in exchange for an increased risk of the true output violating one or more of the constraints, which can be seen by the blue line being closer to the constraints. A particularly close example can be seen at the end of the plot at $t = 40$, where the true output comes very close to the constraints for $\alpha = 0.5$ but stays further away for the scenario approach or $\alpha = 0.01$.

4.3 Robustness

The biggest advantage that this kernel approximation has shown compared to the scenario approach is the adjustable risk factor. This parameter $\alpha \in [0, 1]$ can be chosen depending on how successful the final solution is supposed to be when it comes to satisfying the constraints in future scenarios.
In this section, this parameter is tested by running the same problem setup as was used in Section 4.2 for different values of α as well as increasing and lowering the number of samples N and testing how well the solution holds up for other scenarios that are independent of the ones used in the optimization.
Similar to Section 4.2, a number of scenarios are generated with Algorithm 1. From this set of scenarios, a small subset is then taken and used to formulate several OCPs as was already described in the previous section. The OCPs are then solved, and the resulting optimal input $\boldsymbol{u}_{0:H}$ is applied to $N = 2000$ more independent scenarios from the same system to test how well this solution holds up. For each of the 2000 scenarios, the output is calculated and compared to the constraints that were used in the OCP to check whether or not they are fulfilled. This process is then repeated for various numbers of scenarios. It is done for 1, 5, 10, 25, and 50 samples and then increased with a step size of 50 until $N = 300$ samples are used in the optimization.
In Figure 4.2, the results of this simulation are shown. The percentage of scenarios that fulfill the constraints is plotted over the number of samples used in the initial optimization, which range from $N = 1$ to 300. The various α values are shown as separate lines. Initially, all five lines show very similar results. This can be explained by the fact that at such a low number of scenarios cannot accurately represent the distribution. As the number of scenarios is increased, the approximation of the distribution becomes better leading to a higher percentage of scenarios where the constraints are satisfied.
After around 25 scenarios, the plots start diverging for the first time. While the scenario approach and the plots with smaller α values are very similar, the lines that represent larger α values are starting to display a slightly different percentage

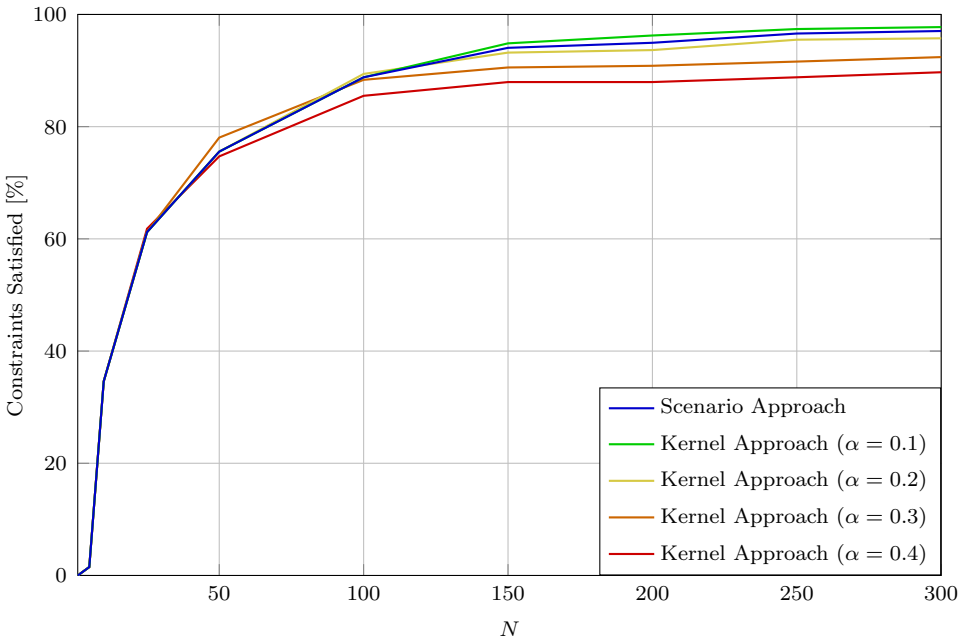


Figure 4.2: Percentage of scenarios where $u_{0:H}$ is a feasible solution. The blue line shows the result of the scenario approach, while the other lines are for the kernel approach with various values of α.

of cases that satisfy the constraints. It can be seen that the kernel approach achieves a more robust solution for $\alpha = 0.3$ and $N = 50$, but since this effect is only present for this one instance, it can be inferred that this is most likely just due to random chance.

For $N \geq 100$ scenarios, a trend starts to become apparent in the different lines as all of them start to slowly converge, with the risk factor α determining to what percentage each line converges. An interesting oddity that can be seen here is that the 3 lines for $\alpha = 0.2, 0.3$, and 0.3 all display the same pattern of their percentage increasing slightly above the rest of the curves before they drop below the rest. After that, the lines converge to various values that are dependent on the value α that's associated with each line.

Meanwhile, the scenario approach displays a similar behavior, with the percentage of scenarios that satisfy the constraints improving as the number of scenarios used for the optimization is increased. Something that stands out here is that, at least within the observed range of samples used in the optimization, the kernel approach seemingly provides a more reliable solution $\boldsymbol{u}_{0:H}$ than the scenario approach. This shows clearly that even if the kernel approach begins with the goal of allowing for a portion of the scenarios to not satisfy the constraints, it can still lead to a more reliable solution than the scenario approach, which, by its definition, tries to satisfy the constraints of as many scenarios as possible.

As the number of scenarios used in the optimization keeps increasing, the first signs of their behavior for $N \to \infty$ become visible. While we require more samples to be

sure of the exact percentages the approaches are converging to, it is already clear that they are not converging towards $(1 - \alpha)$. For example, for $\alpha = 0.4$, it seems to converge towards a value between 85% and 90%, while the same example converges towards somewhere around 95% for $\alpha = 0.1$. This is generally not an issue, however, as α is only an upper bound for the percentage of samples that are allowed to violate the constraints.

4.4 Corridor Test

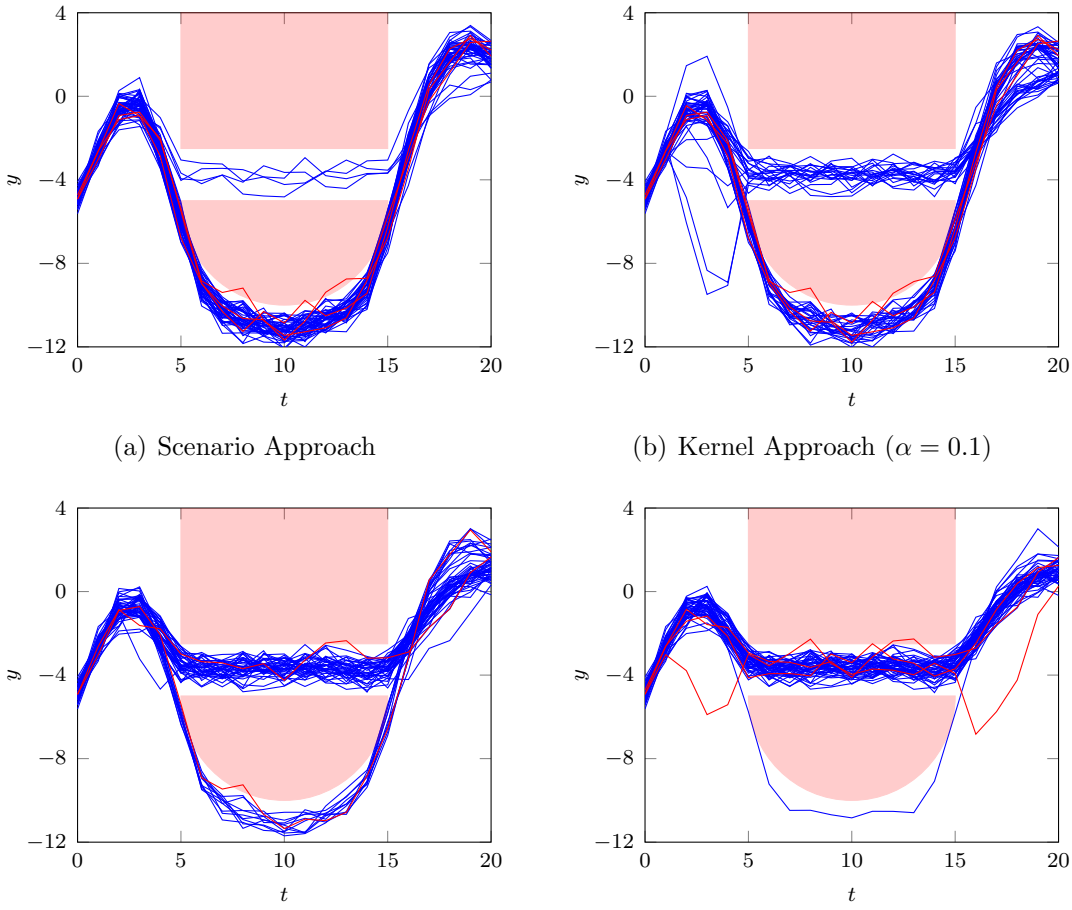
In Section 4.3, it has been shown that the kernel approach allows us to control the risk factor α, leading to a higher percentage of cases violating the constraints. The benefit of this becomes apparent when looking at the cost function J_H. Allowing a higher risk relaxes the constraints, which allows us to choose a solution with a lower cost associated with it to satisfy the constraints. While this can be shown for the example used in Section 4.2 and 4.3, the potential of this approach becomes more clear by looking at an example where the possible solutions are separated into two paths, one with a low cost and hard to satisfy constraints and one with high costs and easy to satisfy constraints.

In this example, we consider the same system as described in Section 4.1 and Section 4.2 where the horizon has been reduced to $H = 20$ and the output y has to either satisfy $-5 \leq y_t \leq -2.5$ or $y_t \leq -5 - \sqrt{25 - (t - 10)^2}$ for $t \in [5, 15]$. We use 2000 scenarios that were generated via Particle Gibbs and randomly select $N = 100$ scenarios to formulate and solve an optimal control problem. The resulting output can then be applied to the true system to obtain a true output which can be analyzed regarding which path it takes and whether or not it satisfies the constraints. This process was repeated 50 times for a different set of randomly chosen scenarios for both the scenario and kernel approach for the risk values $\alpha = 0.1$, 0.2, and 0.3.

In Figure 4.3, the true outputs for these 50 runs have been plotted, as well as the constraints. The runs that arrive at a feasible solution have been colored blue, and the failed runs, i.e., the runs where the true output violated at least one constraint, have been colored red.

Looking at the scenario approach, it can be seen that almost all of the runs end up choosing the less risky path that comes with a higher cost, as can be seen in Table 4.1 where the average cost of 50 runs is shown. In contrast, we can look at the kernel approach for $\alpha = 0.3$, where only a very small portion of the runs take the less risky path despite the same scenarios being used for both approaches. This is also reflected in the average cost, which is significantly lower for the kernel approach. Here, the cost is directly proportional to the percentage of runs that take the safe path. The table also shows the percentage of runs where the true output violated a constraint at least once.

The results for $\alpha = 0.1$ and $\alpha = 0.2$ are also interesting as they show a middle ground between the two extremes. We can see that as α is increased, the number



(a) Scenario Approach

(b) Kernel Approach ($\alpha = 0.1$)

(c) Kernel Approach ($\alpha = 0.2$)

(d) Kernel Approach ($\alpha = 0.3$)

Figure 4.3: Example of the optimal control with known basis functions for scenario approach (top left) and kernel approach. The red areas show the output constraints. The lines show the trajectories of the true system. The outputs that successfully avoid the constraints are colored blue, while all the outputs that violate at least one constraint are colored red.

of runs that go through the risky corridor increases, which leads to a lower average cost and a higher number of failed runs.

4.5 Nonlinear System

In the previous sections, we have looked at how the kernel approach behaves for the linear system (4.1). However, we have chosen the kernel approach over alternative methods such as Wasserstein for its applicability to general nonlinear functions. As such, in this section, we are looking at a nonlinear system and comparing how the system behaves for both the scenario and kernel approach for various risk factors α.

Approach	Scenario	Kernel		
		$\alpha = 10\%$	$\alpha = 20\%$	$\alpha = 30\%$
J_H (mean)	338.4	255.9	129.5	69.9
Risky Runs	8%	40%	78%	98%
Failed Runs	6%	6%	4%	6%

Table 4.1: Results of the 50 exemplary runs with known basis functions for the scenario approach and kernel approach for risk values of 0.1, 0.2, and 0.3. The table shows the average cost of the 50 runs and the percentage of failed Monte Carlo runs, i.e. runs where the true output violated at least one constraint.

We consider the state transition function

$$\boldsymbol{f}(\boldsymbol{x}, u) = \begin{bmatrix} 0.8x_1 - 0.5x_2 + 0.1\cos(3x_1)x_2 \\ 0.4x_1 + 0.5x_2 + (1 + 0.3\sin(2x_2))u \end{bmatrix} \tag{4.5}$$

with the rest of the system being chosen as described in Section 4.1. The new basis functions are chosen as $\boldsymbol{\varphi}(\boldsymbol{x}, u) = [0.1x_1, 0.1x_2, u, 0.01\cos(3x_1)x_2, 0.1\sin(2x_2)u]^{\mathrm{T}}$. With that, we can draw samples as described in Section 4.1 and use them to generate $N = 200$ scenarios. We use a burn-in period of $N_p = 1000$ and thin the samples with $n_d = 50$.
For the cost function, we once again consider a simple quadratic cost $J_H = \sum_{t=0}^{H} u_t^2$ over the horizon $H = 40$. For constraints, we consider the input-constraint $|u| \leq 5$ as well as the temporarily active output-constraints $y_{20:25} \geq 2$. The ambiguity set radius ε is chosen through Algorithm 2 with a number of bootstrap samples $B = 1000$ and a confidence level $\beta = 0.95$ and we look at the kernel approach for the risk factors $\alpha = 0.2$, 0.4 and 0.6 to see what influence this parameter has on the output. Finally, as the problem is no longer convex, we use CasADi to solve the nonlinear problem [AGH$^+$19].
in Figure 4.4, the results of an exemplary run are plotted. Similar to the results in Section 4.2, the output for the scenario approach satisfies the constraints for every single scenario while the kernel approach includes a few scenarios that violate at least one constraint to increasingly stronger degrees as the risk factor α is increased. This shows that despite the system being nonlinear for this example, the kernel approach still works properly and allows for an optimization that allows for some of the output constraints to be violated.
However, in contrast to the linear case, there have been several technical difficulties with the solver. While a convex solver was able to find a solution for the linear case, the nonconvex solver that has been used for this example was unable to solve the reformulated problem directly. While it worked on the scenario approach without issue, the kernel approach provided several challenges for the solver. In order to overcome this, several approximations had to be made before the solver was able to find a feasible solution for the OCP. For one, the max operator had to be approximated with the function

Approach	Scenario	Kernel		
		$\alpha = 20\%$	$\alpha = 40\%$	$\alpha = 60\%$
Computation time	32.8s	176.1s	50.9s	41.0s

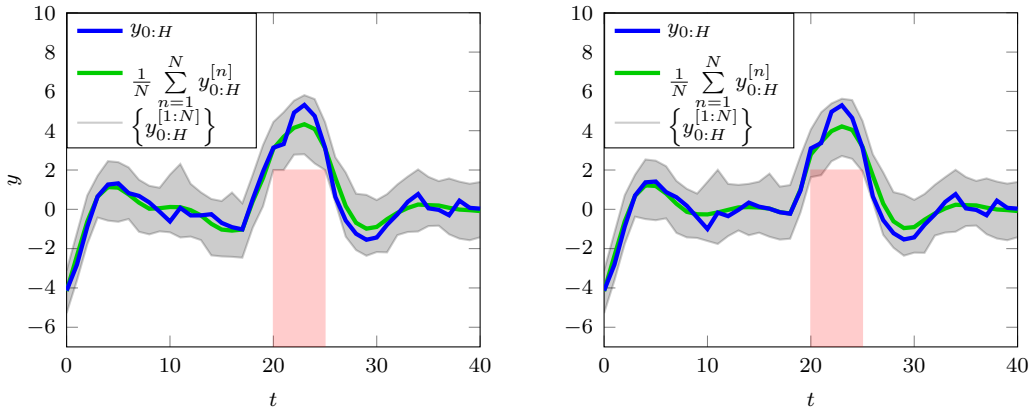
Table 4.2: Computation time for scenario and kernel approach for an exemplary run with a system with nonlinear basis functions.

$$[z]_+ \approx \frac{z + \sqrt{z^2 + 0.01}}{2} \tag{4.6}$$

and the kernel gram matrix had to be changed by adding 0.01 to all diagonal elements. This is necessary as we use the Cholesky decomposition, i.e., $\boldsymbol{K} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ with $\boldsymbol{L}$ being a lower triangle matrix with positive diagonal elements, to rewrite the square root in the constraint (3.13a) as

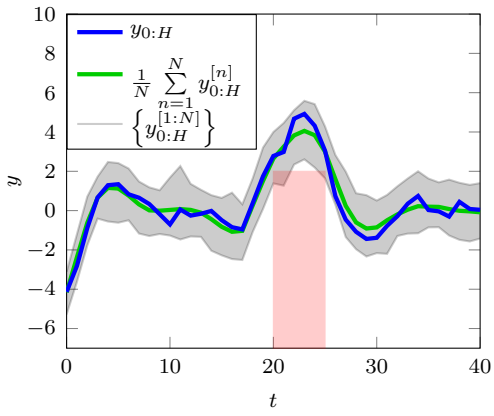
$$\sqrt{\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\gamma}} = ||\boldsymbol{L}^{\mathrm{T}}\boldsymbol{\gamma}||_2. \tag{4.7}$$

This was also done in the simulations for the linear system with a regularisation parameter of 10^{-6} but in the case of the nonlinear dynamics, a parameter of 10^{-2} was used.
With these approximations, the results that can be seen in Figure 4.4 were generated. However, these changes were only sufficient to run the solver for relatively simple problems and only for certain parameters. As can be seen in Table 4.2, the computation time of this simulation increases very quickly for smaller values of the risk factor α. As such, this simulation could only be run for relatively high values of α, which limits the range of problems this solver can be used for. On top of that, the solver also struggles with problems with a higher number of constraints, which is why this simulation has significantly fewer constraints compared to the simulation in Section 4.2. This, too, limits the type of problems this approach can be used for. Lastly, tests also show that the solver struggles with finding a solution for the OCP when a low number of scenarios is used. While this is less of an issue for practical applications, as the results in Section 4.3 show, it nonetheless is another limiting factor for using this approach as an alternative to the scenario approach, which displayed none of these issues for the nonlinear systems. As such, there is a need for further analysis of the solver and what can be done to avoid these issues.

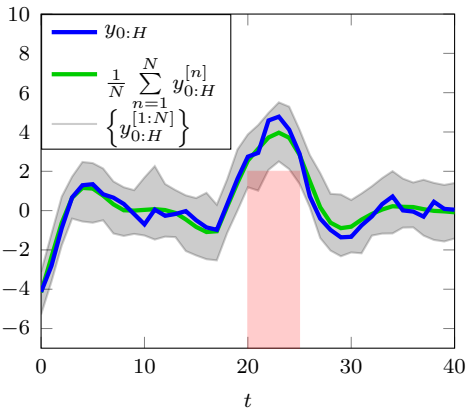


(a) Scenario Approach

(b) Kernel Approach ($\alpha = 0.2$)



(c) Kernel Approach ($\alpha = 0.4$)



(d) Kernel Approach ($\alpha = 0.6$)

Figure 4.4: Example of the optimal control with known nonlinear basis functions for scenario approach (top left) and kernel approach for various values of α. The red areas show the output constraints. The gray area encompasses the 200 scenarios that were used in the optimization, with the green line being the average. The blue line is one realization of the true output.

Chapter 5

Conclusion

This work presents an alternative approach to scenario theory for the use of samples generated using PMCMC methods. We propose the combination of samples drawn from the posterior distribution for the trajectories of an unknown system using a PG sampler with ambiguity sets to find a robust solution to our problem. We do this by defining MMD ambiguity sets with the use of kernel embeddings. These sets are then used to reformulate a chance-constrained OCP while including a parameter for the permitted failure probability. This problem can then be solved using well-known methods.
The viability of this approach is shown in several simulations where the methods are tested for an unknown system with a constrained output. The predicted output has been plotted and compared with the true output for the OCPs that are generated using scenario theory and MMD ambiguity sets. The effect of the permitted failure probability is also shown for various numbers of scenarios being used when formulating the OCP. These results demonstrate that the failure percentage does not not converge to the selected parameter. The potential benefits of this approach are shown in a simulation where a risky path leads to a more cost-effective solution, and the choice of the risk factor has been shown to have a major effect on what path the solver selects. Finally, the approach is also tested on a nonlinear system.

Bibliography

[ADH10] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[AGH+19] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019. doi:10.1007/s12532-018-0139-4.

[BCG11] Stephen R Becker, Emmanuel J Candes, and Michael Grant. Templates for Convex Cone Problems with Applications to Sparse Signal Recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011.

[Fri15] Roger Frigola. Bayesian time series learning with Gaussian processes. *Ph.D. dissertation, University of Cambridge*, 2015.

[GBR+12] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Scholkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research 13*, 2012.

[GC21] Simone Garatti and Marco Campi. The risk of making decisions from data through the lens of the scenario approach. *IFAC-PapersOnLine*, 54(7):607–612, 2021.

[GC22] Simone Garatti and Marco Campi. Risk and Complexity in Scenario Optimization. *Mathematical Programming*, 191(1):243–279, 2022.

[GJK18] Damien Garreau, Wittawat Jitkrittum, and Motonobu Kanagawa. Large Sample Analysis of the Median Heuristic. *arXiv:1707.07269*, 2018.

[HCL19] Ashish R Hota, Ashish Cherukuri, and John Lygeros. Data-Driven Chance Constrained Optimization under Wasserstein Ambiguity Sets. *American Control Conference (ACC)*, 2019.

[LSLH24] Robert Lefringhausen, Supitsana Srithasan, Armin Lederer, and Sandra Hirche. Learning-Based Optimal Control with Performance Guarantees

for Unknown Systems with Latent States. *European Control Conference*, 2024.

[MFSS17] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schoelkopf. Kernel Mean Embedding of Distributions: A Review and Beyond. *Foundations and Trends in Machine Learning*, 10:1–141, 2017.

[MLF21] Michael Maiworm, Daniel Limon, and Rolf Findeisen. Online learning-based model predictive control with Gaussian process models and stability guarantees. *International Journal of Robust and Nonlinear Control*, 31(18):8785–8812, 2021.

[NKSZ22] Yassine Nemmour, Heiner Kremer, Bernhard Schoelkopf, and Jia-Jie Zhu. Maximum Mean Discrepancy Distributionally Robust Nonlinear Chance-Constrained Optimization with Finite-Sample Guarantee. *arXiv preprint arXiv:2204.11564*, 2022.

[NS07] Arkadi Nemirovski and Alexander Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 2007.

[NW06] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, New York, NY, USA, 2. edition, 2006.

[SS17] Andreas Svensson and Thomas Schoen. A flexible state-space model for learning nonlinear dynamical systems. *Automatica*, 80:189–199, 2017.

[TLOZ22] Adam J. Thorpe, Thomas Lew, Meeko M. K. Oishi, and Jia-Jie Zhu. Data-Driven Chance Constrained Control using Kernel Distribution Embeddings. *Learn. for Dynamics and Ctrl. Conf.*, 168:790–802, 2022.

[TO21] Adam J. Thorpe and Meeko M. K. Oishi. Stochastic Optimal Control via Hilbert Space Embeddings of Distributions. *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 904–911, 2021.

[WR06] Christopher K. I. Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA, 2006.

[ZJDS20] Jia-Jie Zhu, Wittawat Jitkrittum, Moritz Diehl, and Bernhard Schoelkopf. Kernel distributionally robust optimization: Generalized duality theorem and stochastic approximation. *International Conference on Artificial Intelligence and Statistics.*, 2020.

License