TUM

Kernel Embedding for Particle Gibbs-Based Optimal Control

Scientific thesis for the procurance of the degree M.Sc.
(Intermediate report)
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	??? Lukas Hochschwarzer l.hochschwarzer@gmail.com
Submitted on	Munich, XX.XX.2024

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

A short (1–3 paragraphs) summary of the work. Should state the problem, major assumptions, basic idea of solution, results. Avoid non–standard terms and acronyms. The abstract must be able to be read completely on its own, detached from any other work (e.g., in collections of paper abstracts). Do not use references in an abstract.

Contents

Chapter 1

Introduction

Your first chapter in the document. Introduce the problem (gently!). Try to give the reader an appreciation of the difficulty, and an idea of how you will go about it. It is like the overture of an opera: it plays on all the relevant themes.
In this chapter, the problem is introduced in Sec. 1.1 and some

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 $\boldsymbol{x} \in \mathbb{R}^{n_x \in \mathbb{N}}$, the input $\boldsymbol{u} \in \mathbb{R}^{n_u \in \mathbb{N}}$, the output $\boldsymbol{y} \in \mathbb{R}^{n_y \in \mathbb{N}}$, 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 $\boldsymbol{y}$ is observable 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 $\boldsymbol{v}$ and measurement noise $\boldsymbol{w}$ are unknown.
We assume that a dataset $\mathbb{D} = \{\boldsymbol{u}_t, \boldsymbol{y}_t\}_{t=-T:-1}$ containing the last $T \in \mathbb{N}$ measurements of the input $\boldsymbol{u}$ and output $\boldsymbol{y}$ is available.
We further assume that the structure of the model $\{\boldsymbol{f_\theta}(\cdot), \boldsymbol{g_\theta}(\cdot), \boldsymbol{\mathcal{V}_\theta}, \boldsymbol{\mathcal{W}_\theta}\}$ is known and is dependent on a finite number of unknown parameters $\boldsymbol{\theta}$. In addition to that, the priors $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x}_{-T})$ 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 to us and there are several uncertain factors in our system, the constraints are transformed into chance-constraints and since it is possible that $\boldsymbol{h}$ is impossible to satisfy for every possible $\boldsymbol{x}_{0:H}$, we also introduce a risk factor $\alpha \in [0, 1]$ that relaxes these constraints, turning them into

$$P\left[h_i(\boldsymbol{u}_{0:H}, \boldsymbol{x}_{0:H}, \boldsymbol{y}_{0:H}) \leq 0\right] \geq 1 - \alpha, \ \forall i = 1, ..., n_c \tag{1.4}$$

with h_i being the i-th element of $\boldsymbol{h}$ and P being generally unknown.

1.2   Related Work

The problem presented in Sec. 1.1 provides several challenges as the information we have available is very limited. While many methods to solve chance constraint problems exist, they often rely on samples from the distribution P which is generally unknown in our problem. While we do have priors for the uncertain elements, samples drawn from this distributions do not allow as to quantify the system enough for practical applications. As such, the priors must be updated based on the input-output measurements $\mathbb{D}$ which generally results in an analytically intractable posterior distribution.

To draw samples from this distribution, methods such as particle Markov chain Monte Carlo Methods [ADH10] can be used. This has recently been exploited for optimal control in [LSLH24] , utilizing such a sampler to generate scenarios for the system and using these scenarios as a representation of the unknown distribution to formulate a deterministic optimal control problem by reformulating the chance-constraints. However, the usage of the scenarios in this paper comes with the drawback of the risk factor α not being part of the final optimal control problem (OCP) and the process of estimating it retroactively being quite ressource intensive. As such, there is a need to find other methods that allow us to utilize the samples generated by PG 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 P, ambiguity sets have been proposed as a possible workaround. Here, ambiguity sets are defined as a set of probability distributions that are within a certain radius under an appropriate distance function. This was used in [HCL19] with Wasserstein distance as the metric for the ambiguity set. It has however been proven rather difficult to efficiently construct a Wasserstein ambiguity set for problems.

In contrast, the metric proposed in [NKSZ22] allows for an efficient construction of an ambiguity set using a maximum mean discrepancy (MMD) metric combined with kernel approximation and can be applied to gneral non-linear and non-convex constraints.

The remainder of this paper is structured as follows. In chapter 2, we review the methods used to solve the OCP. These methods are then tested and evaluated in

chapter 3 and further discussed in chapter 4. Finally, the results are summarized and some concluding remarks are given in chapter 5.

Chapter 2

Technical Approach

In this chapter, an approach is outlined that allows us to effectively solve the chance-constraint problem defined in 1.1. In section 2.1 we draw samples from unknown system and use them to generate scenarios. These scenarios are then used in 2.2 to reformulate and solve the OCP.

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 $p(\boldsymbol{\theta}, \boldsymbol{x}_{\text{-}T:\text{-}1} \mid \mathbb{D})$. This 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 then draw samples from this distribution are particle Markov chain Monte Carlo (PMCMC) methods which is explained in [ADH10] and will be summarized in this section.

While in general, the inference of the psoterior PMCMC methods $p(\boldsymbol{\theta}, \boldsymbol{x}_{\text{-}T:\text{-}1} \mid \mathbb{D})$ is analytically intractable, PMCMC bypass this issue by iteratively drawing samples from $p(\boldsymbol{\theta} \mid \boldsymbol{x}_{\text{-}T:\text{-}1}, \mathbb{D})$ and $p(\boldsymbol{x}_{\text{-}T:\text{-}1} \mid \boldsymbol{\theta}, \mathbb{D})$ while updating 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 as such might show a strong bias. This burn in period should be chosen large enough so that this bias is no longer reflected in the samples. The samples should also be indepedendent of each other which is not given with this method as each $\boldsymbol{\theta}^{[n]}$ is dependent on $\boldsymbol{x}_{\text{-}T:\text{-}1}^{[n]}$ 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.

Algorithm 1 Scenario generation

    Input: Dataset $\mathbb{D}$, parametric model $\{\boldsymbol{f_\theta}(\cdot), \boldsymbol{g_\theta}(\cdot), \boldsymbol{\mathcal{V}_\theta}, \boldsymbol{\mathcal{W}_\theta}\}$,
        priors $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x}_{-T})$, 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 $\{\boldsymbol{\theta}, \boldsymbol{x}_{-T:-1}\}^{[n]}$ from $p\left(\boldsymbol{\theta}, \boldsymbol{x}_{-T:-1} \mid \mathbb{D}\right)$ using a PMCMC method
3:     **for** $t = -1, \ldots, H$ **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:     $\boldsymbol{x}_0^{[n]} \leftarrow \boldsymbol{f}_{\boldsymbol{\theta}^{[n]}}\left(\boldsymbol{x}_{-1}^{[n]}, \boldsymbol{u}_{-1}\right) + \boldsymbol{v}_{-1}^{[n]}$
8: **end for**

The samples $\{\boldsymbol{\theta}, \boldsymbol{x}_{-T:-1}\}^{[1:N]}$ can be used to generate 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 can be seen in Algorithm 1. $\boldsymbol{\theta}^{[n]}$ is 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 $\boldsymbol{w}_{0:H}$. Those can then be combined with the $\boldsymbol{x}_{-T:-1}$, or more precisely $\boldsymbol{x}_{-1}$ 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   Chance-Constraint Optimization with Kernel Approximation

In the previous section, a method that enables us to generate a finite number of scenarios $\boldsymbol{\delta}^{[1:N]}$ was presented. These scenarios can be used to formulate an OCP to find an optimal $\boldsymbol{u}$ or a control law $\boldsymbol{\pi}$. In this section, a method to use maximum mean discrepancy (MMD) ambiguity sets and kernel approximation to reformulate the OCP is proposed.

2.2.1   MMD ambiguity sets

As the underlying data distribution P in the constraints 1.3 is unknown, we first expand the 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 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

$$\inf_{\tilde{P}\in\mathcal{P}} \tilde{P}\left[h(\boldsymbol{u}_{0:H}, \boldsymbol{x}_{0:H}, \boldsymbol{y}_{0:H}) \leq 0\right] \geq 1 - \alpha. \tag{2.1}$$

To construct the set $\mathcal{P}$, a similarity measure is needed to provide a concrete comparison between various distributions $\tilde{P}$. Maximum mean discrepancy (MMD) [GBR+12] is able to do that by using the norm of the difference between the kernel mean embeddings (KME) $||\mu_Q - \mu_{Q'}||_{\mathcal{H}}^2$ of two distributions Q and Q' as a metric between two distributions. The KME are given as $\mu_Q = \int k(x, \cdot)\mathrm{d}x$ with $k(x, \cdot) \in \mathcal{H}$ being the feature map of the kernel function k. The metric can then be rewritten as

$$\mathrm{MMD}(Q, Q') = \mathrm{E}_{x,x'\sim Q}[k(x, x')] + \mathrm{E}_{y,y'\sim Q'}[k(y, y')] - 2\mathrm{E}_{x\sim Q, y\sim Q'}[k(x, y)] \tag{2.2}$$

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 $\hat{P}_N$ which is given through the scenarios $\boldsymbol{\delta}^{[1:N]}$. This gives us the set

$$\mathcal{P} = \{P : \mathrm{MMD}(P, P_N) \leq \varepsilon\}. \tag{2.3}$$

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

$$\widehat{\mathrm{MMD}}(\tilde{P}, P_N) = \sum_{i,j=1}^{N} k(\boldsymbol{\delta}^{[i]}, \boldsymbol{\delta}^{[j]}) + k(\tilde{\boldsymbol{\delta}}^{[i]}, \tilde{\boldsymbol{\delta}}^{[j]}) - 2k(\boldsymbol{\delta}^{[i]}, \tilde{\boldsymbol{\delta}}^{[j]}) \tag{2.4}$$

with $\tilde{\boldsymbol{\delta}}^{[n]}, n = 1, ..., N$, denoting a bootstrap sample of P_N where samples are drawn with replacement from $\boldsymbol{\delta}^{[1:N]}$. Finally, $\widehat{\mathrm{MMD}}(\tilde{P}, P_N)$ is calculated for all B bootstrap samples, the results are saved in list and ε is chosen as the *ceil*$(B\beta)$-th element of the sorted list.

2.2.2 Constraint Reformulation

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

$$Z_i := \left\{\boldsymbol{u}_{0:H} \in \mathcal{U}^{H+1} : \inf_{P\in\mathcal{P}} P\left[\tilde{h}_i(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}) \leq 0\right] \geq 1 - \alpha\right\}. \tag{2.5}$$

with $\tilde{h}_i(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}) = h_i(\boldsymbol{u}_{0:H}, \boldsymbol{x}_{0:H}, \boldsymbol{y}_{0:H})$
We can now use the ambiguity set $\mathcal{P}$ defined in Sec. 2.2.1 with the radius ε, as well as the kernel matrix $\boldsymbol{K}$ to reformulate the feasible set. By following the steps described in [NKSZ22], we can obtain the new reformulated feasible set as

---

Algorithm 2 Bootstrap MMD ambiguity set

$\quad$ **Input**: Scenarios $\boldsymbol{\delta}^{[1:N]}$, Number of bootstrap samples B, Confidence level β

$\quad$ **Output**: Gram matrix $\boldsymbol{K}$, Radius of MMD ambiguity set ε

1: $\boldsymbol{K} \leftarrow kernel(\boldsymbol{\delta}, \boldsymbol{\delta})$

2: **for** $m = 1, \ldots, B$ **do**

3: $\quad$ $I \leftarrow N$ numbers from $\{1, \ldots N\}$ with replacement

4: $\quad$ $K_x \leftarrow \sum_{i,j=1}^{N} K_{ij}$;

5: $\quad$ $K_y \leftarrow \sum_{i,j \in I} K_{ij}$;

6: $\quad$ $K_{xy} \leftarrow \sum_{j \in I} \sum_{i=1}^{N} K_{ij}$;

7: $\quad$ $\mathrm{MMD}[m] \leftarrow \frac{1}{N^2} \left(K_x + K_y - 2K_{xy} \right)$;

8: **end for**

9: $\mathrm{MMD} \leftarrow sort(\mathrm{MMD})$

10: $\varepsilon \leftarrow \mathrm{MMD}[ceil(B\beta)]$

$$Z_i := \left\{ \boldsymbol{u}_{0:H} \in \mathcal{U}^{H+1} : \begin{array}{ll} g_0 + \dfrac{1}{N} \sum_{n=1}^{N} (\boldsymbol{K}\boldsymbol{\gamma})_n + \varepsilon \sqrt{\boldsymbol{\gamma}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\gamma}} \leq t\alpha & \text{(2.6a)} \\[2ex] [\tilde{h}_i(\boldsymbol{u}_{0:H}, \boldsymbol{\delta}^{[n]}) + t]_+ \leq g_0 + (\boldsymbol{K}\boldsymbol{\gamma})_n, \ n = 0, ..., N & \text{(2.6b)} \\[1ex] g_0 \in \mathbb{R}, \boldsymbol{\gamma} \in \mathbb{R}^N, t \in \mathbb{R} & \text{(2.6c)} \end{array} \right.$$

where $[\cdot]_+ = \max(0, \cdot)$ denotes the max operator. The new set also contains new variables that have to be taken into account when solving the OCP. The variables g_0, γ and t are the degrees of freedom that u has at each constraint. The former are parameters of the RKHS function that was used to transform the problem into a kernel machine learning problem while t has been introduced as a way to relax the risk factor α.

This approximation can then be repeated for all other constraints $\boldsymbol{h}(\cdot)$ to obtrain the feasible sets $Z_i, i = 1, ..., n_c$. We can then use those constraints to formulate the OCP as

$$\min_{\boldsymbol{u}_{0:H}, \{\boldsymbol{\gamma}, g_0, t'\}^{[1:n_c]}} J_H(\boldsymbol{u}_{0:H}) \tag{2.7a}$$

$$\text{s.t. } \forall n = 1, ..., N, \ \forall t = 0, 1, ..., H, \ \forall i = 1, ..., n_c \tag{2.7b}$$

$$\boldsymbol{x}_{t+1}^{[n]} = \boldsymbol{f}_{\boldsymbol{\theta}^{[n]}} \left(\boldsymbol{x}_t^{[n]}, \boldsymbol{u}_t \right) + \boldsymbol{v}_t^{[n]} \tag{2.7c}$$

$$\boldsymbol{y}_t = \boldsymbol{g}_{\boldsymbol{\theta}^{[n]}} \left(\boldsymbol{x}_t^{[n]}, \boldsymbol{u}_t \right) + \boldsymbol{w}_t^{[n]} \tag{2.7d}$$

$$\boldsymbol{u}_{0:H} \in Z_i(\boldsymbol{\gamma}^{[i]}, g_0^{[i]}, t'^{[i]}) \tag{2.7e}$$

As described in Sec. 1.1, we are minimizing a cost function J_H. The system dynamics are included through the constraints 2.7c and 2.7d and must be fulfilled for all scenarios as well. Lastly, the input $u_{0:H}$ is restricted to the feasible sets Z_i, i.e. $u_{0:H}$ must be an element of all $Z_i, i = 1, ..., n_c$.

The optimization problem 2.7 is deterministic and, with the uncertainties removed, can be solved with well known methods.

Chapter 3

Evaluation

In this section, the proposed optimal control approach is implemented and its effectiveness tested and compared to the previous approach. The simulation setup is described in Sec. 3.1. The results of the OCP are shown in Sec. 3.2. Afterwards, we analyse the results and performance in more detail in Sec. 3.3 by looking at the robustness of the solution and compare it to the commonly used scenario approach.

3.1 Simulation Setup

We consider a system with the state transition function

$$\boldsymbol{f}(\boldsymbol{x}, u) = \begin{bmatrix} 0.8x_1 - 0.5x_2 \\ 0.4x_1 + 0.5x_2 + u \end{bmatrix} \tag{3.1}$$

and the process noise distribution

$$\boldsymbol{v}_t \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} 0.03 & \text{-}0.004 \\ \text{-}0.004 & 0.01 \end{bmatrix}\right). \tag{3.2}$$

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 $w_t \sim \mathcal{N}(0, 0.1)$ are known.

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 $\boldsymbol{x}_{\text{-}T} \sim \mathcal{N}([2, 2]^{\text{T}}, \boldsymbol{I}_2)$. To infer the state model parameters, the approach from [ADH17] 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

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{\varphi}(\boldsymbol{x}_t, u_t) + \boldsymbol{v}_t \tag{3.3}$$

with the basis functions $\varphi(\boldsymbol{x}, u) = [x_1, x_2, u]^{\mathrm{T}}$, the process noise $\boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and the unknown parameters $\boldsymbol{\theta}$ consisting of $\boldsymbol{A}$ and $\boldsymbol{Q}$. 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 $\boldsymbol{A}$ matrix normal prior with mean matrix $\boldsymbol{M} = \boldsymbol{0}$, right covariance $\boldsymbol{U} = \boldsymbol{Q}$ and left covariance matrix $\boldsymbol{V} \in \mathbb{R}^{n_a \times n_a}$. For the estimation of the posterior disdribution with the PG sampler, we scale the basisvector with the weights $[0.1, 0.1, 1]^{\mathrm{T}}$ and for the prior the weights are chosen as $\boldsymbol{V} = 10\boldsymbol{I}_5$.

We use gaussian kernels $k(x, y) = \exp\left(-\frac{1}{2\sigma^2}||x - y||_2^2\right)$ with the bandwidth σ set individually for all random parameters $\left\{\boldsymbol{x}_0^{[k]}, \boldsymbol{v}_{0:H}^{[k]}, w_{0:H}^{[k]}, \boldsymbol{A}^{[k]}\right\}$ via the median heuristic [GJK18] and scaled with the factors $[1.5, 5, 5, 1]^{\mathrm{T}}$. As such, the elements of the Gram matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ are defined as

$$K_{ij} = k_A(\boldsymbol{A}^{[i]}, \boldsymbol{A}^{[j]})k_{\mathcal{X}}(\boldsymbol{x}_0^{[i]}, \boldsymbol{x}_0^{[j]})k_{\mathcal{V}^H}(\boldsymbol{v}_{0:H}^{[i]}, \boldsymbol{v}_{0:H}^{[j]})k_{\mathcal{W}^H}(\boldsymbol{w}_{0:H}^{[i]}, \boldsymbol{w}_{0:H}^{[j]}). \qquad (3.4)$$

3.2 Optimal Control with Constrained Output

In the following, we show how well the proposed optimal control approach works when applied to a OCP with constrained output by putting it side by side with the solution of the same problem where we have used the Scenario approach which implements the chance-constraints by ensuring that the constraints are satisfied for every scenario $n \in \mathbb{N}_{\leq N}$.

For this simulation, we are scenarios that have been generated using the PG sampler. To this end, 9170 samples were created and the first $N_p = 1000$ were discarded as training samples and the remaining samples were once again thinned with $n_d = 30$. 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 risk level α is fixed at 0.1 for this experiment and ϵ is chosen through Algorithm 2.

The OCP can then be formulated as described in **??**. Since the problem has been chosen as convex in this example, a solution for both the scenario and kernel approach can be found easily by using a convex solver. The results of an exemplary run is shown in Fig. 3.1.

The figure includes the two plots for the scenario and kernel approach and shows the output y of their respective OCPs. The graphs shows 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 that, it also shows of the mean of these trajectories and true output. Where the two graphs differ however is to what extend the solutions fulfill the constraints. By its definition, the scenario approach requires all scenarios to fulfill the constraints which can be seen in the solution. While the gray area touches the min and max constraints in several places, it never violates the constraints. The kernel approach on the other hand has a risk factor α built in

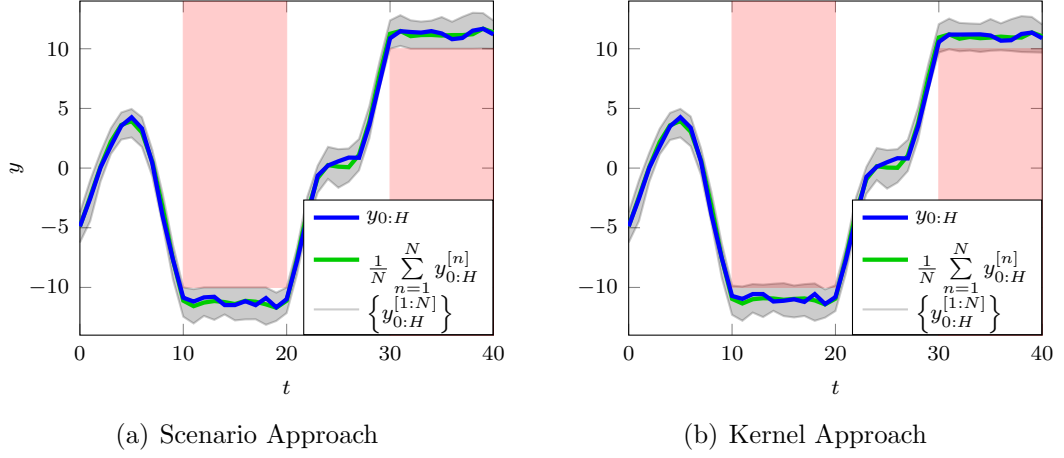(a) Scenario Approach   (b) Kernel Approach

Figure 3.1: Example of the optimal control with known basis functions for scenario approach (left) and kernel approach (right). 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.

which allows for a number of scenarios to violate the constraints as long a sufficient number satisfies them. This can be seen in the constraint still being met by the majority of the scenarios and only the trajectory of a small number of scenarios actively overlapping with the marked area. As a result, a solution with a lower cost was found in exchange for an increased risk of the true output violating one or more of the constraints.

3.3 Robustness

The biggest advantage that this kernel approximation proves compared to the scenario approach is the adjustable risk factor. As described in Sec. 2.2, this method includes a parameter $\alpha \in [0,1]$ which 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 Sec. 3.2 for different values of α as well as an increasing number of samples N and testing how well the solution holds up for future scenarios.

Similar to Sec. 3.2, a number of scenarios are generated with Algorithm 1. From this set scenarios, a small subset is then taken and used to formulate several OCPs as was already described in Sec. 3.2. The OCPs are then solved and the resulting optimal input $\boldsymbol{u}_{0:H}$ is then used on $N = 2000$ more independant 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
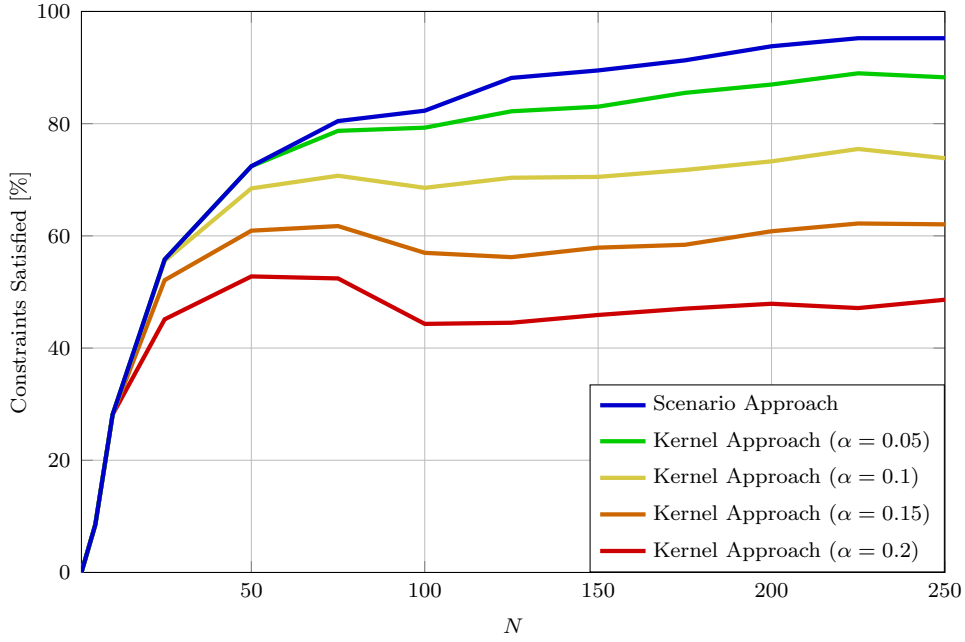
Figure 3.2: Percentage of scenarios where $u_{0:H}$ is a viable solution. The blue line shows the result of the scenario approach while the other lines are for the kernel approach with various values of α.

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 and 25 samples and then increased with a step size of 25 until a $N = 250$ samples are used in the optimization.

In Fig. 3.2 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 250. The various α values are shown as separate lines. Initially, all five plots 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 10 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 worse percentage of cases that satisfy the constraints. This trend continues as the number of scenarios used in the optimization keeps increasing. While the scenario approach keeps increasing, the kernel approach seems to converge to a significantly lower level of robustness based on the selection of the parameter α. This shows that through α we are able to control how distributionally robust the solution is.

Chapter 4

Discussion

Discuss and explain your results. Show how they support your thesis (or, if they do not, give a convincing explanation). It is important to separate objective facts clearly from their discussion (which is bound to contain subjective opinion). If the reader does not understand your results, reconsider if you have managed to extract the core information and explain it in a straightforward way.

Chapter 5

Conclusion

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.

[ADH17]   Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. A flexible
           state-space model for learning nonlinear dynamical systems. *Automatica*,
           80:189–199, 2017.

[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.

[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.

[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.

License