

# Kernel Embedding for Particle Gibbs-Based Optimal Control

Scientific thesis for the procurement 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./Univ. Tokio habil. Martin Buss  
M.Sc. Robert Lefringhausen  
Chair of Automatic Control Engineering

**Submitted by** Lukas Hochschwarzer  
Koenigswieser Strasse 75e  
81475 Munich  
089 - 1015332

**Submitted on** Munich, XX.XX.2024



April 26, 2024

MASTER'S THESIS  
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

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Problem Statement . . . . .	5
1.2	Related Work . . . . .	5
<b>2</b>	<b>Technical Approach</b>	<b>7</b>
2.1	Particle Gibbs Sampling . . . . .	7
2.2	Chance-Constraint Optimization with Kernel Approximation . . . . .	7
2.2.1	Bootstrap construction of MMD ambiguity sets . . . . .	7
2.2.2	Reproducing Kernel Hilbert Spaces . . . . .	7
<b>3</b>	<b>Evaluation</b>	<b>9</b>
3.1	Setup . . . . .	9
3.2	Optimal Control with known basis functions . . . . .	9
3.3	Robustness . . . . .	10
3.4	Computation Time . . . . .	11
<b>4</b>	<b>Discussion</b>	<b>13</b>
<b>5</b>	<b>Conclusion</b>	<b>15</b>
	<b>List of Figures</b>	<b>17</b>
	<b>List of Tables</b>	<b>19</b>
	<b>Bibliography</b>	<b>21</b>





# 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.

Make sure you clearly state the vision/aims of your work, which problem you are trying to solve, and why it is important. While the introduction is the part that is read first (ignoring title and abstract), it is usually best written last (when you actually know what you have really achieved). Remember, it is the first thing that is read and will have a major influence on the how the reader approaches your work. If you bore them now, you have most likely lost them already. If you make outrageous claims, pretend to solve the world's problems, etc, you are likely fighting an uphill battle later on. Also, make sure you pick up any threads spun in the introduction later on, to ensure that the reader thinks they get what they have been promised. Do not create an expectation that you will deliver more than you actually do. Remember, the reader may be your marker (of a thesis) or referee (of a paper), and you do not want to annoy them.

### 1.1 Problem Statement

You can either state the problem you are trying to solve in the general introduction, providing the transition from the overall picture to your specific approach, or state it in a separate section. Even if you do not use the separate section, writing down in a few sentences why the problem you are trying to solve is actually hard and has not been solved before can give you a better idea of how to approach the topic. This can be also merged with the related work part.

### 1.2 Related Work

“The related work section (sometimes called literature review) is just that, a review of work related to the problem you are attempting to solve. It should identify and

evaluate past approaches to the problem. It should also identify similar solutions to yours that have been applied to other problems not necessarily directly related to the one you are solving. Reviewing the successes or limitations of your proposed solution in other contexts provides important understanding that should result in avoiding past mistakes, taking advantage of previous successes, and most importantly, potentially improving your solution or the technique in general when applied in your context and others.

In addition to the obvious purpose indicated, the related work section also can serve to:

- justify that the problem exists by example and argument
- motivate interest in your work by demonstrating relevance and importance
- identify the important issues
- provide background to your solution

Any remaining doubts over the existence, justification, motivation, or relevance of your thesis topic or problem at the end of the introduction should be gone by the end of related work section.

Note that a literature review is just that, a review. It is not a list of papers and a description of their contents! A literature review should critique, categorize, evaluate, and summarize work related to your thesis. Related work is also not a brain dump of everything you know in the field. You are not writing a textbook; only include information directly related to your topic, problem, or solution.”

Note: Write literature review at an early stage of your project to build on the knowledge of others, not reinvent the wheel over and over again! There is nothing more frustrating after weeks or months of hard work to find that your great solution has been published 5 years ago and is considered old news or that there is a method known that produces superior results.

[LSLH24]

[NKSZ22]

[TLOZ22]

## Chapter 2

# Technical Approach

## 2.1 Particle Gibbs Sampling

---

**Algorithm 1** Scenario generation

---

**Input:** Dataset  $\mathbb{D}$ , parametric model  $\{\mathbf{f}_{\boldsymbol{\theta}}(\cdot), \mathbf{g}_{\boldsymbol{\theta}}(\cdot), \mathbf{V}_{\boldsymbol{\theta}}, \mathbf{W}_{\boldsymbol{\theta}}\}$ ,  
priors  $p(\boldsymbol{\theta})$  and  $p(\mathbf{x}_T)$ ,  $N, H, T$   
**Output:** Scenarios  $\boldsymbol{\delta}^{[1:N]} = \{\boldsymbol{\theta}, \mathbf{x}_0, \mathbf{v}_{0:H}, \mathbf{w}_{0:H}\}^{[1:N]}$

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

---

## 2.2 Chance-Constraint Optimization with Kernel Approximation

### 2.2.1 Bootstrap construction of MMD ambiguity sets

### 2.2.2 Reproducing Kernel Hilbert Spaces

---

**Algorithm 2** Bootstrap MMD ambiguity set
 

---

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

**Output:** Radius of MMD ambiguity set  $\varepsilon$

```

1:  $K \leftarrow \text{kernel}(\delta, \delta)$ 
2: for  $m = 1, \dots, B$  do
3:    $I \leftarrow N$  numbers from  $\{1, \dots, N\}$  with replacement
4:    $K_x \leftarrow \sum_{i,j=1}^N K_{ij}$ ;
5:    $K_y \leftarrow \sum_{i,j \in I} K_{ij}$ ;
6:    $K_{xy} \leftarrow \sum_{j \in I} \sum_{i=1}^N K_{ij}$ ;
7:    $\text{MMD}[m] \leftarrow \frac{1}{N^2} (K_x + K_y - 2K_{xy})$ ;
8: end for
9:  $\text{MMD} \leftarrow \text{sort}(\text{MMD})$ 
10:  $\varepsilon \leftarrow \text{MMD}[\text{ceil}(B\beta)]$ 

```

---

# 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 and the results are shown in Sec. 3.2. Sec. 3.4 and Sec. 3.3 then focuses on the general performance of this new method in comparison to the previously used Scenario approach.

### 3.1 Setup

### 3.2 Optimal Control with known basis functions

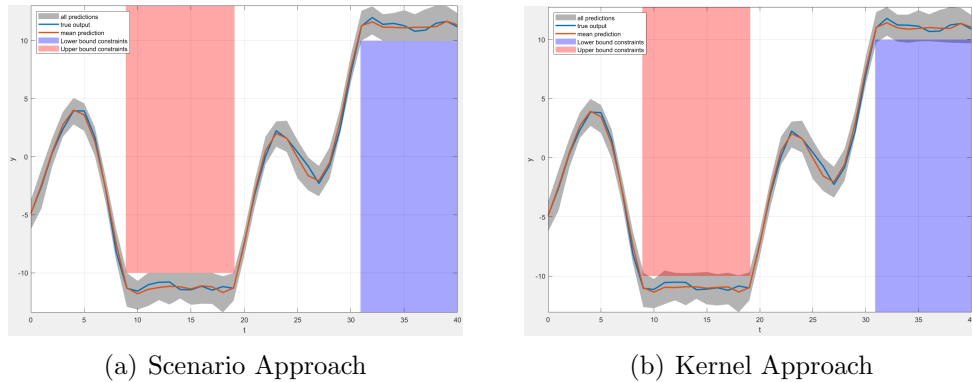


Figure 3.1: Example of the optimal control with known basis functions for scenario approach (left) and kernel approach (right). The red and blue area show the output constraints. The gray area encompasses the 200 scenarios that were used in the optimization with the orange line being the average. The blue line is one realization the true output.

### 3.3 Robustness

The biggest advantage that this kernel approximation proves compared to the scenario approach is the adjustable robustness of the solution. As described in Sec. 2.2, this method includes a parameter  $\alpha \in [0, 1]$  which can be chosen depending on how robust the final solution is supposed to be.

In this section, this parameter is tested by running the same problem setup as was used in Sec. 3.2 for different values of  $\alpha$  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,  $N = 200$  scenarios are generated with Algorithm 1. From this set of 200 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  $\mathbf{u}_{0:H}$  is then used on  $N = 2000$  more scenarios to estimate the robustness of the solution. For each of the 2000 scenarios, the output is calculated and compared to the constraints that were used in the OCP. This process is then repeated over and over for a slightly larger subset of scenarios until finally the full set of  $N = 200$  scenarios is used.

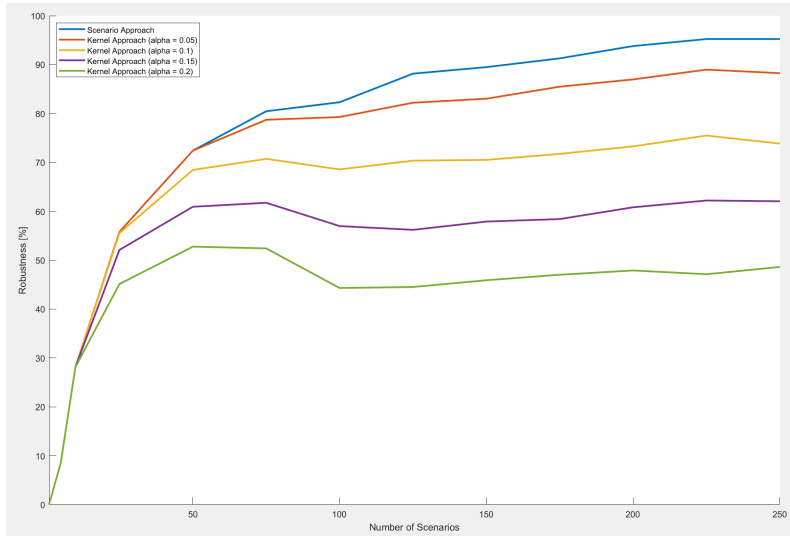


Figure 3.2: Estimated Robustness of solution  $\mathbf{u}$  over number of samples  $N$ . The blue line shows the result of the scenario approach while the other lines are for the Kernel approach with various values of  $\alpha$

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 200. The various  $\alpha$  values are shown as separate lines. Initially, all five plots show very robustness. 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 as well and the solution of the OCP now represents a larger number of the distribution.

After around 10 scenarios, the plots start diverging for the first time. While the scenario approach and the plots with smaller  $\alpha$  values are very similar, the lines that represent larger  $\alpha$  values are starting to display worse robustness. 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  $\alpha$ . This shows that the robustness can be somewhat controlled with the parameter  $\alpha$ .

### 3.4 Computation Time

While the previous sections focused on comparing the results of the Scenario approach to the kernel approach, the runtime of both methods is another important factor that needs to be considered when choosing which algorithm is best suited for a specific problem. As such, Fig. 3.3 shows the runtime of both scenario approach and kernel approach over the number of scenarios that are used to formulate the OCPs. The parameters and constraints are chosen the same as in Sec. 3.2. When looking at the figure, it quickly becomes apparent that while both curves start around the same level, the kernel approach runtime increases significantly faster and with more of a curvature than the scenario approach. By the end of the plot, the kernel approach already takes about 5 times as long as the scenario approach. This shows that at least when it comes to finding a solution quickly, the Scenario approach is still the better choice.

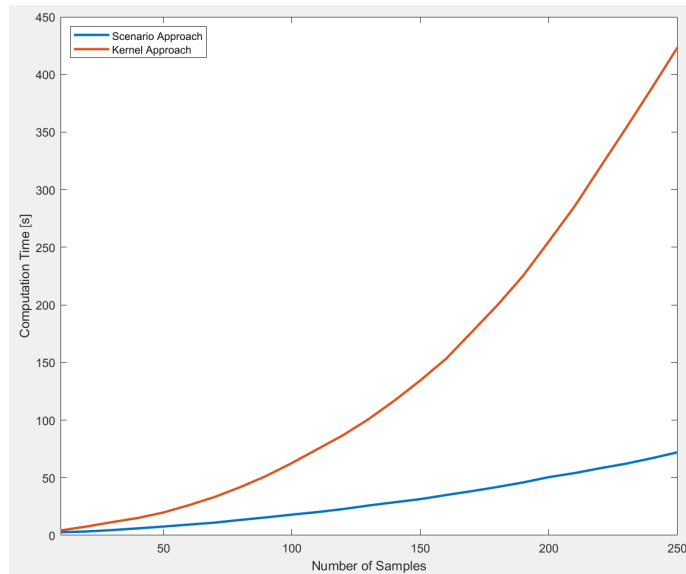


Figure 3.3: Runtime over number of samples  $N$  for scenario and kernel approach





## 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

Do not just leave it at the discussion: discuss what you/the reader can learn from the results. Draw some real conclusions. Separate discussion/interpretation of the results clearly from the conclusions you draw from them. (So-called “conclusion creep” tends to upset reviewers. It means surrendering your scientific objectivity.) Identify all shortcomings/limitations of your work, and discuss how they could be fixed (“future work”). It is not a sign of weakness of your work if you clearly analyze and state the limitations. Informed readers will notice them anyway and draw their own conclusions if not addressed properly.

Recap: do not stick to this structure at all cost. Also, remember that the thesis must be:

- honest, stating clearly all limitations
- self-contained, do not write just for the locals, do not assume that the reader has read the same literature as you, do not let the reader work out the details for themselves

This chapter is followed by the list of figures and the bibliography. If you are using acronyms, listing them (with the expanded full name) before the bibliography is also a good idea. The acronyms package helps with consistency and an automatic listing.



# List of Figures

3.1	Example of the optimal control with known basis functions for scenario approach (left) and kernel approach (right). The red and blue area show the output constraints. The gray area encompasses the 200 scenarios that were used in the optimization with the orange line being the average. The blue line is one realization the true output. . . . .	9
3.2	Estimated Robustness of solution $\mathbf{u}$ over number of samples $N$ . The blue line shows the result of the scenario approach while the other lines are for the Kernel approach with various values of $\alpha$ . . . . .	10
3.3	Runtime over number of samples $N$ for scenario and kernel approach	11



# List of Tables





# Bibliography

- [LSLH24] Robert Lefringhausen, Supitsana Srithasan, Armin Lederer, and Sandra Hirche. Learning-Based Optimal Control with Performance Guarantees for Unknown Systems with Latent States. 2024. URL: [arXiv:2303.17963](#).
- [NKSZ22] Yassine Nemmour, Heiner Kremer, Bernhard Schoelkopf, and Jia-Jie Zhu. Maximum Mean Discrepancy Distributionally Robust Nonlinear Chance-Constrained Optimization with Finite-Sample Guarantee. 2022. URL: [arXiv:2204.11564v1](#).
- [TLOZ22] Adam J. Thorpe, Thomas Lew, Meeko M. K. Oishi, and Jia-Jie Zhu. Data-Driven Chance Constrained Control using Kernel Distribution Embeddings. 2022. URL: [arXiv:2202.04193](#).



# License

This work is licensed under the Creative Commons Attribution 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.