KernelSOS for Global Sampling-Based Optimal Control and Estimation via Semidefinite Programming

Antoine Groudiev
École Normale Supérieure

Frederike Dümbgen
WILLOW Team, Inria

Justin Carpentier
WILLOW Team, Inria

August 6, 2025

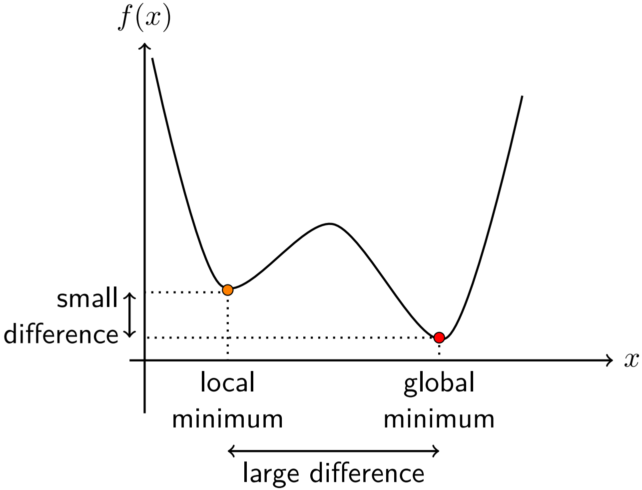
Plan

KernelSOS method
- Problem statement
- Range-only localization
- KernelSOS algorithm

Application to robotics
- Range-only localization
- Trajectory optimization
- Warm start for trajectory optimization

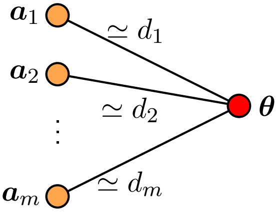
Conclusion and future work
- Conclusion
- Other future works

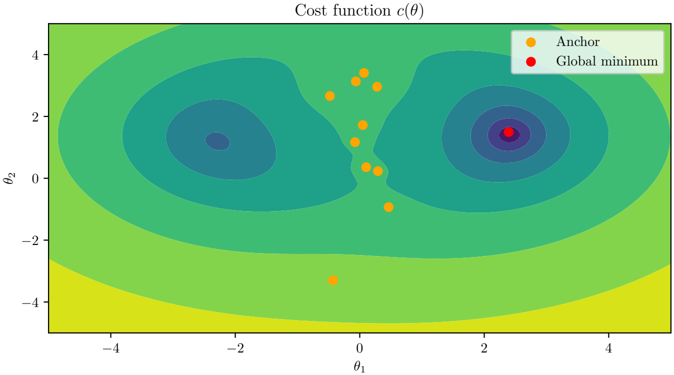
Problem statement: global optimization



- Many problems in robotics are non-convex $\Rightarrow$ local minima
- **Estimation**: local solution can lead to unsafe behavior
- **Control**: local solution can lead to suboptimal trajectories
- **Learning**: global solution can improve sample efficiency

An example: range-only localization





Assuming $d_i \sim \mathcal{N}(\|\boldsymbol{\theta} - \boldsymbol{a}_k\|_2, 1)$: $\boldsymbol{\theta}_{\mathsf{ML}} = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{k=1}^{m} \left(d_k - \|\boldsymbol{\theta} - \boldsymbol{a}_k\|_2\right)^2$

An example: range-only localization

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{k=1}^{m} \left(d_k - \|\boldsymbol{\theta} - \boldsymbol{a}_k\|_2\right)^2$$

Not polynomial because of the norm! Existing approaches:

- Square the measurements [1]:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{k=1}^{m} \left(d_k^2 - \|\boldsymbol{\theta} - \boldsymbol{a}_k\|_2^2\right)^2$$

- Not equivalent to ML problem, but often effective.

- Substitution trick [6]:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d, \boldsymbol{n}_k\in\mathsf{S}^d} \sum_{k=1}^{m} \|\boldsymbol{n}_k d_k - (\boldsymbol{\theta} - \boldsymbol{a}_k)\|_2^2$$

- Equivalent to ML problem, but higher-dimensional and constrained.

Question: is there a better way?

KernelSOS reformulation [9]

$$\min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x})$$

Non-convex

$$\max_{c\in\mathbb{R}} c \quad \text{s.t.} \quad \forall \boldsymbol{x} \in \Omega,\ f(\boldsymbol{x}) - c \geqslant 0$$

Convex but ∞ constraints

$$\max_{c\in\mathbb{R}, \boldsymbol{A}\in\mathbb{S}_+(\mathcal{H})} c \quad \text{s.t.} \quad \forall x \in \Omega,\ f(\boldsymbol{x}) - c = \langle \phi(\boldsymbol{x}), \boldsymbol{A}\phi(\boldsymbol{x})\rangle$$

∞ space $\mathcal{H}$, ∞ constraints

$$\max_{c\in\mathbb{R}, \boldsymbol{A}\in\mathbb{S}_+(\mathcal{H})} c - \lambda \operatorname{Tr}(\boldsymbol{A}) \quad \text{s.t.} \quad \forall i \in [\![1, n]\!], f(\boldsymbol{x}_i) - c = \langle \phi(\boldsymbol{x}_i), \boldsymbol{A}\phi(\boldsymbol{x}_i)\rangle$$

$\begin{cases} \infty \text{ space } \mathcal{H} \\ n \text{ constr.} \end{cases}$

$$\max_{c\in\mathbb{R}, \boldsymbol{B}\in\mathbb{S}^n_+(\mathbb{R})} c - \lambda \operatorname{Tr}(\boldsymbol{B}) \quad \text{s.t.} \quad \forall i \in [\![1, n]\!],\ f(\boldsymbol{x}_i) - c = \boldsymbol{\Phi}_i^\top \boldsymbol{B} \boldsymbol{\Phi}_i$$

SDP

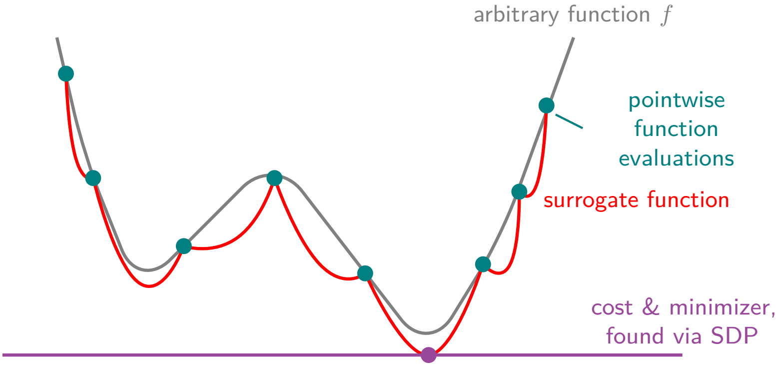


Figure 1: Illustration of the kernelSOS method.
The surrogate function is built using a kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ and the samples in green.

The kernelSOS algorithm

- The $\boldsymbol{\Phi}_i$ can be computed using the kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$
- If $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, then $\boldsymbol{\Phi}_i$ is the i-th column of the Cholesky factorization of $\boldsymbol{K}_{ij}$
- Choice of kernel:
  - Gaussian kernel $k_\sigma(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|^2}{2\sigma^2}\right)$ of scale factor σ
  - Laplace kernel $k_\sigma(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|}{\sigma}\right)$ of scale factor σ (for non-smooth functions)
- We solve the SDP using a custom damped Newton solver that exploits the structure of the problem
- We can retrieve the minimizer using the dual variable

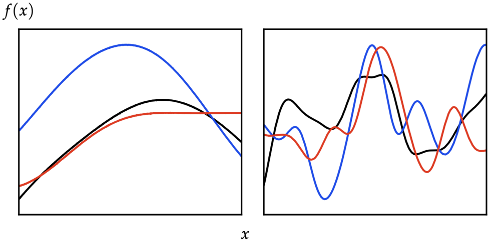


Figure 2: Gaussian kernel

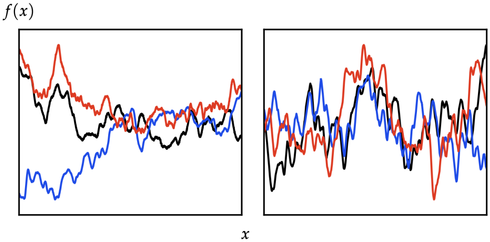


Figure 3: Laplace kernel

Warm restarts

We use a *warm restarts* procedure, using each time the previous solution as the center of the search space

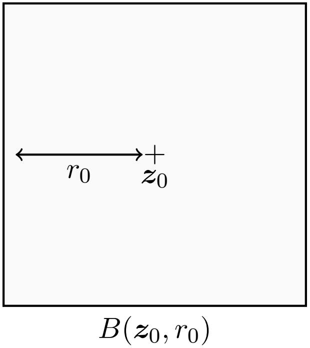


Figure 4: Illustration of the warm restarting mechanism in 2D. The algorithm starts with a large region $B(\boldsymbol{z}_0, r_0)$, and iteratively shrinks it down to $B(\boldsymbol{z}_w, r_w)$.

Warm restarts

We use a *warm restarts* procedure, using each time the previous solution as the center of the search space

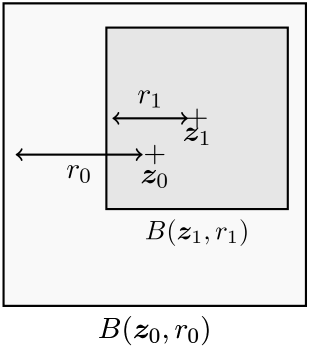


Figure 4: Illustration of the warm restarting mechanism in 2D. The algorithm starts with a large region $B(\boldsymbol{z}_0, r_0)$, and iteratively shrinks it down to $B(\boldsymbol{z}_w, r_w)$.

Warm restarts

We use a *warm restarts* procedure, using each time the previous solution as the center of the search space

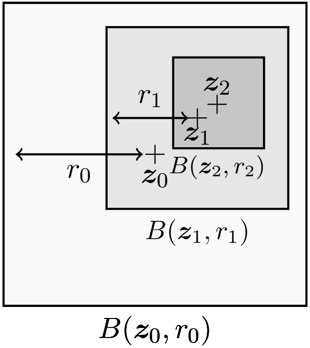


Figure 4: Illustration of the warm restarting mechanism in 2D. The algorithm starts with a large region $B(z_0, r_0)$, and iteratively shrinks it down to $B(z_w, r_w)$.

Plan

KernelSOS method
- Problem statement
- Range-only localization
- KernelSOS algorithm

Application to robotics
- Range-only localization
- Trajectory optimization
- Warm start for trajectory optimization

Conclusion and future work
- Conclusion
- Other future works

Range-only localization – Problem statements

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{k=1}^{m} \left(d_k - \|\boldsymbol{\theta} - \boldsymbol{a}_k\|_2\right)^2$$

Not polynomial because of the norm! Existing approaches:

- Square the measurements [1]:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{k=1}^{m} \left(d_k^2 - \|\boldsymbol{\theta} - \boldsymbol{a}_k\|_2^2\right)^2$$

- Not equivalent to ML problem, but often effective.

- Substitution trick [6]:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d, \boldsymbol{n}_k\in\mathsf{S}^d} \sum_{k=1}^{m} \|\boldsymbol{n}_k d_k - (\boldsymbol{\theta} - \boldsymbol{a}_k)\|_2^2$$

- Equivalent to ML problem, but higher-dimensional and constrained.

Question: is there a better way?

Range-only localization – Methods

Different methods to solve the problem:

- **global**: local solver initialized at ground truth
- **equationSOS**: Shor's relaxation of the POP lifted as a QCQP
- **sampleSOS**: parameterize problem using feasible samples and cost evaluations [2]
- **kernelSOS**: kernel sums of squares on the cost function
- **naive**: sample the search space and take the best sample

(Ground truth is not necessarily the global minimum for high noise)

Range-only localization – Error results

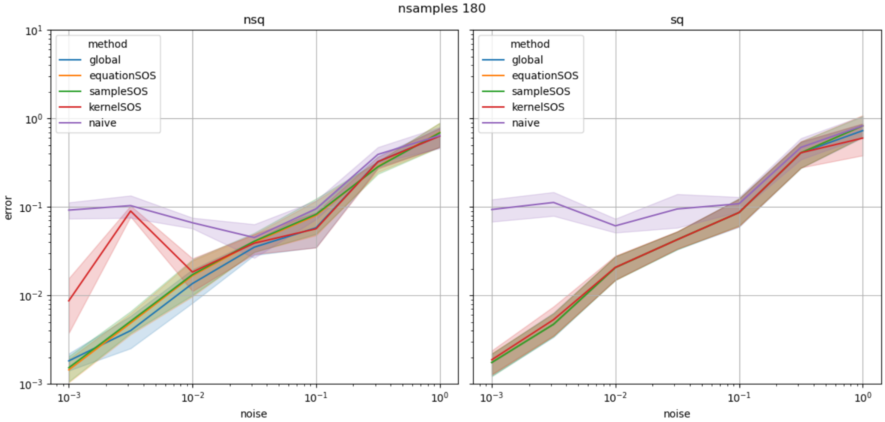


Figure 5: Distance to ground truth as a function of the noise level.

Range-only localization – Time results

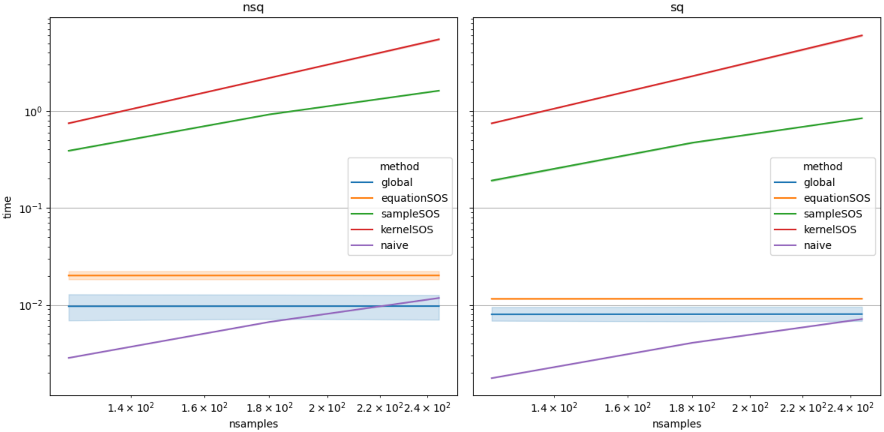


Figure 6: Time taken to solve the range-only localization problem w.r.t the number of samples.

Trajectory optimization

- Trajectory optimization problem:

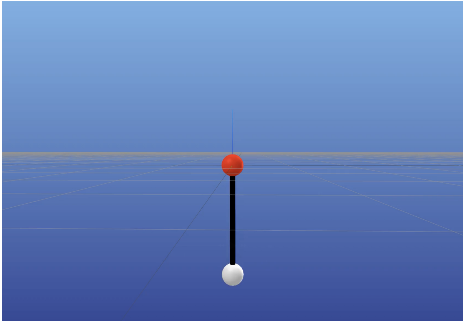
$$\min_{\boldsymbol{u}_{1:T}} \|\boldsymbol{x}_{T+1}(\boldsymbol{u}_{1:T})\|^2 + \rho \sum_{t=1}^{T} \|\boldsymbol{u}_t\|^2 =: f_{\mathrm{TO}}\left(\boldsymbol{u}_{1:T} \mid \boldsymbol{x}_{\mathrm{start}}\right),$$

$$\text{s.t. } \boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t),\ \boldsymbol{x}_1 = \boldsymbol{x}_{\mathrm{start}}$$

  for a number of steps T, control penalty ρ, and initial state $\boldsymbol{x}_{\mathrm{start}}$.
- Optimize over $\boldsymbol{u}_{1:T}$ only (single shooting)
- Black-box approach: evaluate final cost for different choices of $\boldsymbol{u}_{1:T}$

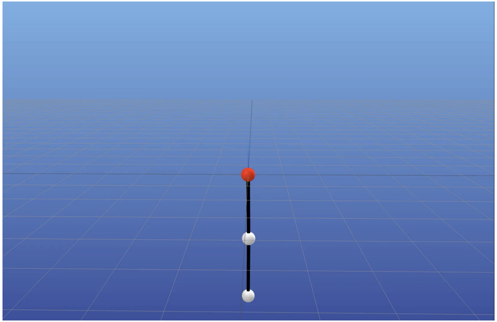
Trajectory optimization – Problems

Single pendulum swing-up



State $\boldsymbol{x} = [\theta, \dot{\theta}]$
Control $\boldsymbol{u} = [\tau]$

Double endulum swing-up



State $\boldsymbol{x} = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]$
Control $\boldsymbol{u} = [\tau_1, \tau_2]$

Trajectory optimization – Results

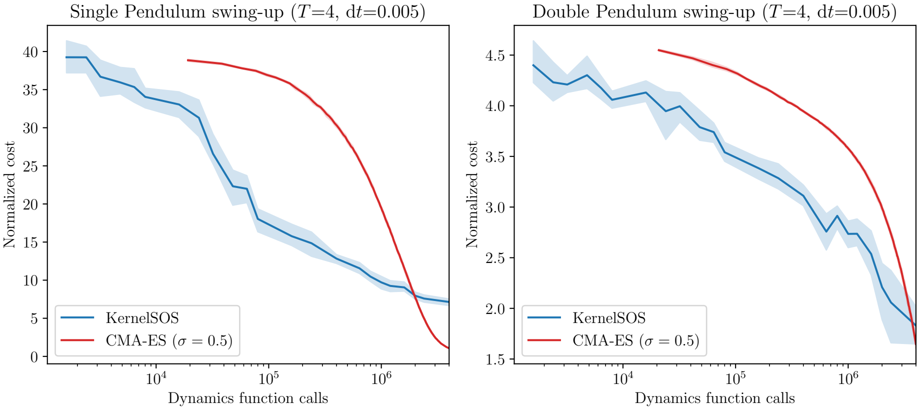


Figure 7: Comparison of performance of kernelSOS and CMA-ES [7] on black-box trajectory optimization.

KernelSOS to warm start a first-order method

- Double pendulum swing-up problem
- KernelSOS is used to warm start `aligator`'s iLQR algorithm

- Initialization methods:
  - **Random**: take one random sample from the search space
  - **Best sample**: take the best sample out of n in the search space
  - **KernelSOS**: use the kernelSOS solution as the initial guess for iLQR

KernelSOS as a warm start – Cost results

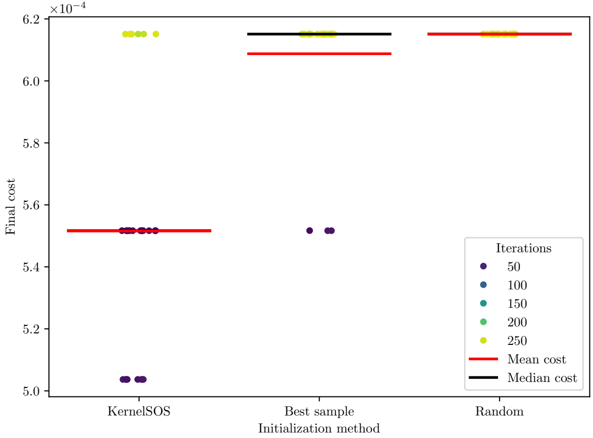
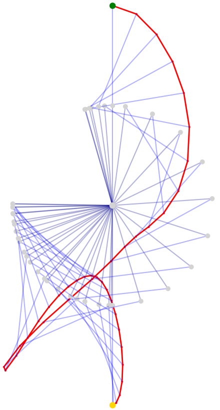


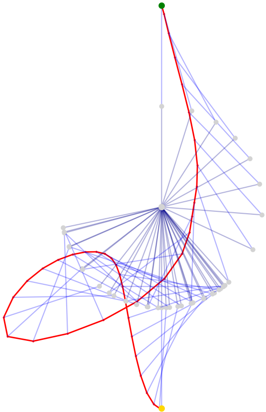
Figure 8: Warm starting of iLQR using kernelSOS.

- Average cost improvement: 12%
- Max. cost improvement: 20%
- Discover new trajectories
  → next slide

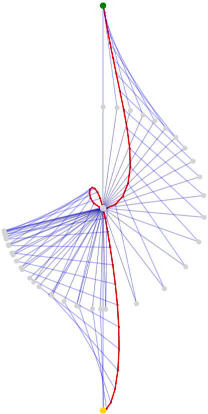
Visualization of the trajectories



Trajectory found by random init. (cost: 6.2×10^{-4})



Trajectory found by best sample init. (cost: 5.5×10^{-4})



Trajectory found by kernelSOS init. (cost: 5.0×10^{-4})

KernelSOS as a warm start – Time results

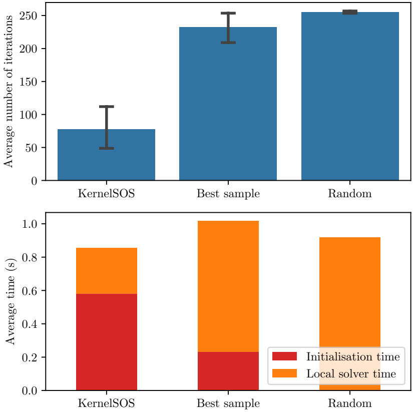


Figure 9: Warm starting of iLQR using kernelSOS.

- Reduces by 77% the average number of iterations needed for convergence
- Similar total time despite longer initialization time

KernelSOS finds trajectories with:

→ lower cost

→ faster convergence

Plan

KernelSOS method
- Problem statement
- Range-only localization
- KernelSOS algorithm

Application to robotics
- Range-only localization
- Trajectory optimization
- Warm start for trajectory optimization

Conclusion and future work
- Conclusion
- Other future works

Conclusion

- KernelSOS is a strong new contender in the **black-box optimization** field
- Generalizes the SOS method to **non-polynomial functions**
- Applicable to a **wide range of problems**
- Discover **improved solutions** in the search space
- Lack of precision $\longrightarrow$ **local solver to refine the solution**

Other future directions

- Application to **policy optimization**
- **Automatic parameter tuning**, for instance using Sobolev norm estimation
- Comparison to standard SOS, using **polynomial kernels**
- **Improve warm starting**, taking inspiration from Bayesian Optimization [5]
- Use of **first-order information**

References I

[1] Amir Beck, Petre Stoica, and Jian Li. "Exact and Approximate Solutions of Source Localization Problems". In: *IEEE Transactions on Signal Processing* 56.5 (2008), pp. 1770–1778.

[2] Diego Cifuentes and Pablo A. Parrilo. "Sampling Algebraic Varieties for Sum of Squares Programs". In: *SIAM Journal on Optimization* 27.4 (2017). http://arxiv.org/abs/1511.06751, pp. 2381–2404.

[3] Frederike Dümbgen, Connor Holmes, and Timothy D Barfoot. "Safe and smooth: Certified continuous-time range-only localization". In: *IEEE Robotics and Automation Letters* 8.2 (2022), pp. 1117–1124.

[4] Frederike Dümbgen et al. "Toward globally optimal state estimation using automatically tightened semidefinite relaxations". In: *IEEE Transactions on Robotics* (2024).

References II

[5] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

[6] Trevor Halsted and Mac Schwager. "The Riemannian Elevator for Certifiable Distance-based Localization". In: *Preprint* (2022).

[7] Nikolaus Hansen. *The CMA Evolution Strategy: A Tutorial*. http://arxiv.org/abs/1604.00772. 2023.

[8] Sarah Müller, Alexander von Rohr, and Sebastian Trimpe. "Local policy search with Bayesian optimization". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 20708–20720.

[9] Alessandro Rudi, Ulysse Marteau-Ferey, and Francis Bach. "Finding global minima via kernel approximations". In: *Mathematical Programming* (2024), pp. 1–82.