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

Antoine Groudiev[1] Fabian Schramm[1,2] Éloïse Berthier[3] Justin Carpentier[1,2] Frederike Dümbgen[1,2]

[1]École Normale Supérieure, PSL Research University, Paris, France.
[2]Inria, Département d'informatique de l'ENS, CNRS, PSL Research University, Paris, France.
[3]U2IS, ENSTA, Institut Polytechnique de Paris, Palaiseau, France.

What is KernelSOS?

KernelSOS [3] is a **sampling-based zeroth-order optimization algorithm**, that solves the following problem:

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

It generalizes the **Sum-of-Squares (SOS) optimization framework** to cases where f is non-polynomial or non-parametric.

KernelSOS formulation

Using only function evaluations $f(\boldsymbol{x}_i)$ at sampled points $\boldsymbol{x}_i \in \Omega$, KernelSOS uses a kernel function $k(\boldsymbol{x}, \boldsymbol{y})$ to define a **surrogate function**, and minimizes it by solving a Semidefinite Program (SDP):

$$\min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) \qquad \text{Non-convex}$$

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

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

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

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

While in the original SOS framework, we enforce that $f-c$ is a sum of squared polynomials, KernelSOS enforces that $f-c$ is a **quadratic form over a Hilbert space** $\mathcal{H}$ defined by the kernel k.

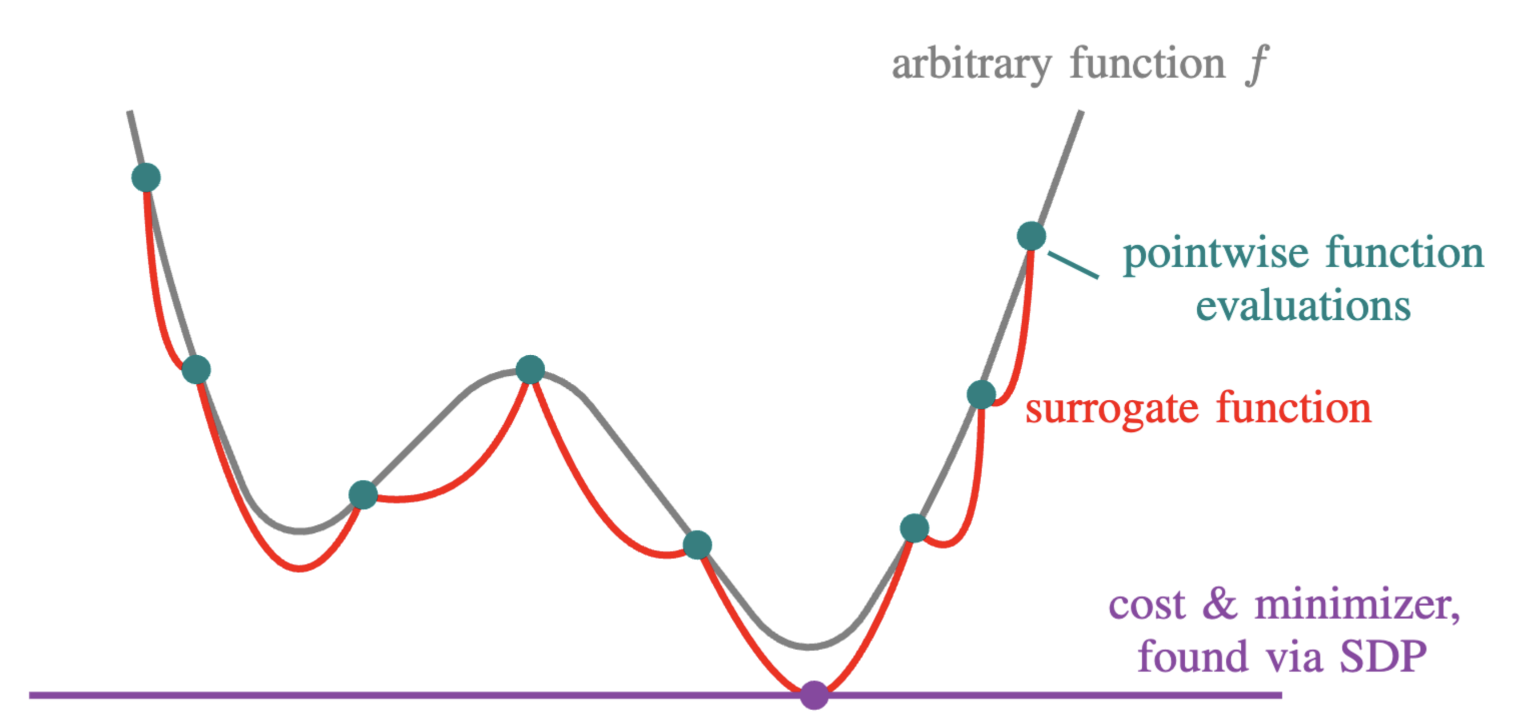


Figure 1. Visualization of the KernelSOS algorithm on a univariate function. The algorithm minimizes a kernel-defined surrogate function (red) based on samples of the original function (green).

Range-only localization

First, we consider a classic state estimation problem, range-only localization:

$$\min_{\boldsymbol{x}} \underbrace{\sum_{i=1}^{m} \frac{1}{\sigma_i^2}(d_i - \|\boldsymbol{x} - \boldsymbol{a}_i\|_2)^2}_{=: f_{\text{RO-non-sq}}(\boldsymbol{x})}, \tag{1}$$

where $\boldsymbol{a}_i$ are known positions of landmarks, d_i are measured distances to the landmarks, and σ_i are the measurement noise levels.

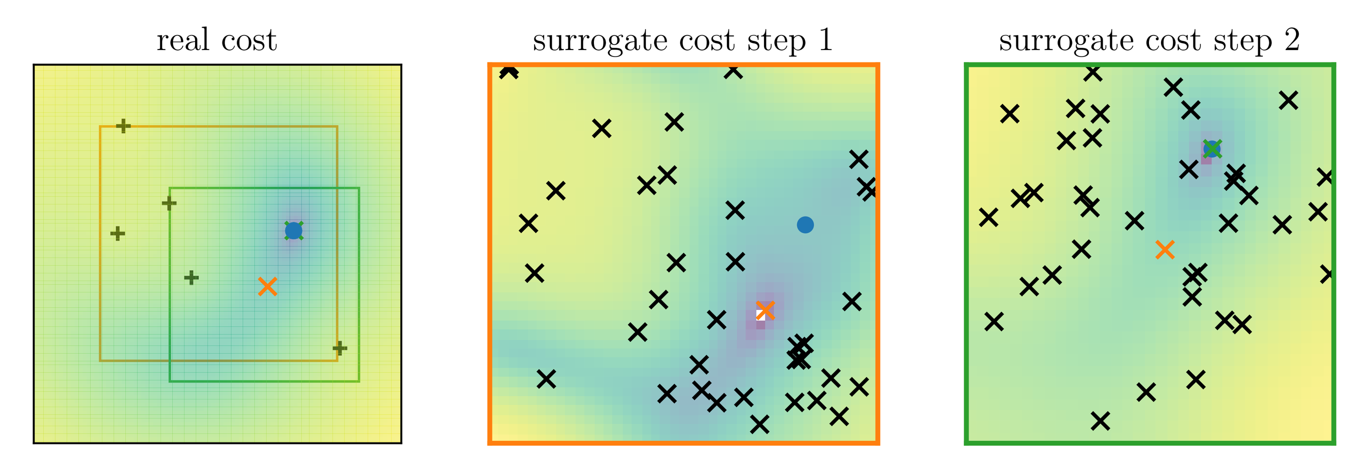


Figure 2. Illustration of the surrogate cost and the benefit of restarts. Left: real cost; middle: surrogate cost found by the first KernelSOS step; right: surrogate function found by the first restart (second step). Black crosses represent the known landmarks, and black x-marks represent the used samples.

Trajectory optimization

Considering the following trajectory optimization problem:

$$\underbrace{\min_{\boldsymbol{u}_{1:T}} \|\boldsymbol{x}_{T+1}(\boldsymbol{u}_{1:T})\|^2 + \rho\sum_{t=1}^{T}\|\boldsymbol{u}_t\|^2}_{=: f_{\text{TO}}(\boldsymbol{u}_{1:T}\,|\,\boldsymbol{x}_{\text{start}})} \quad \text{s.t.} \quad \boldsymbol{x}_{t+1} = g(\boldsymbol{x}_t, \boldsymbol{u}_t),\ \boldsymbol{x}_1 = \boldsymbol{x}_{\text{start}},$$

KernelSOS can be used to optimize trajectories in a **sampling-based manner**, simply by sampling the cost function f_{TO}, which features many local minima.

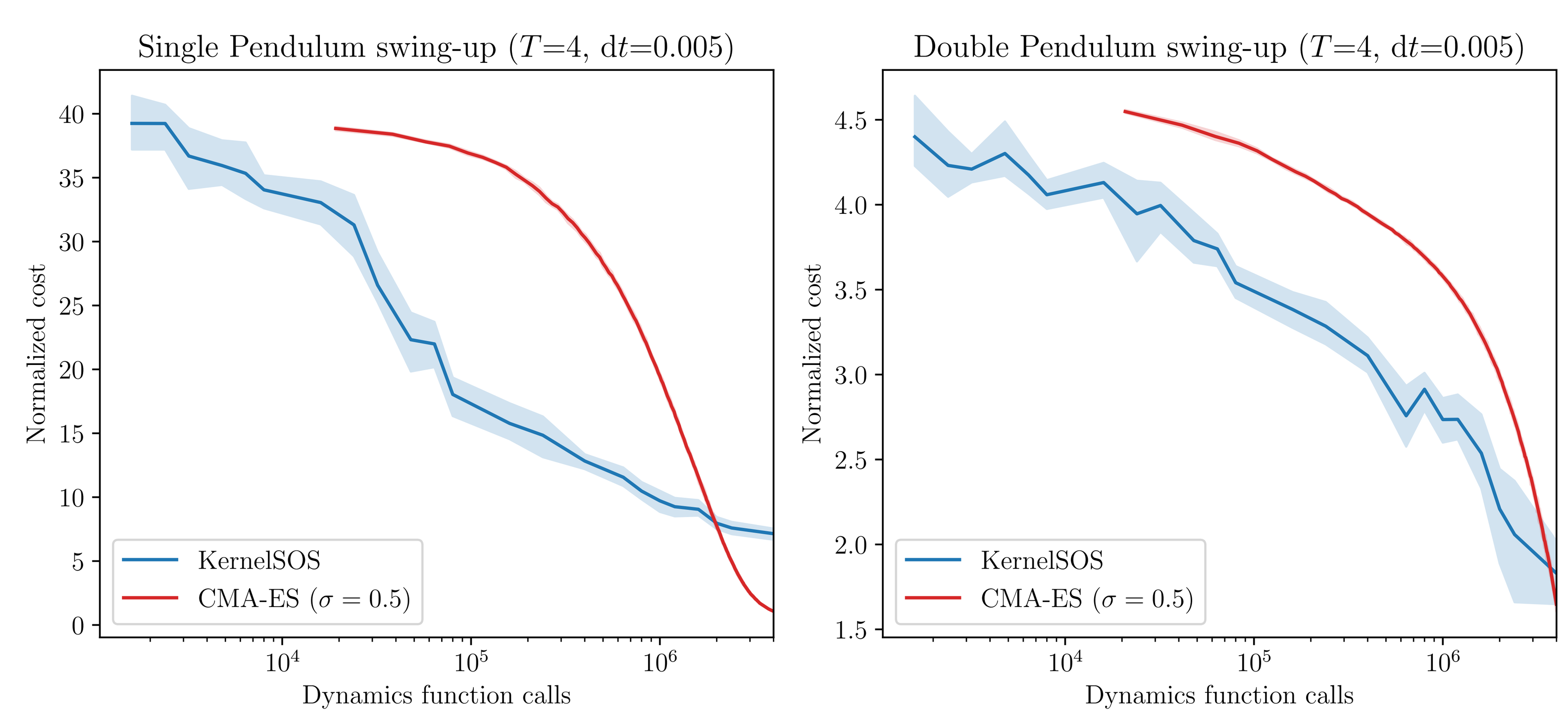


Figure 3. Comparison of KernelSOS and CMA-ES on trajectory optimization problems.

KernelSOS outperforms CMA-ES when the sample density is low. KernelSOS can be seen as a **globally-aware optimization method**, while CMA-ES acts in a more **local manner**.

Initializing local solvers with KernelSOS

We use KernelSOS to initialize a local solver, FDDP [2]. By leveraging the **local accuracy of FDDP** while taking advantage of the **global exploration capabilities of KernelSOS**, lower-cost solutions are found (left). The improved initialization from KernelSOS leads to a lower number of iterations for FDDP to converge, leading to a **similar total runtime** (right).

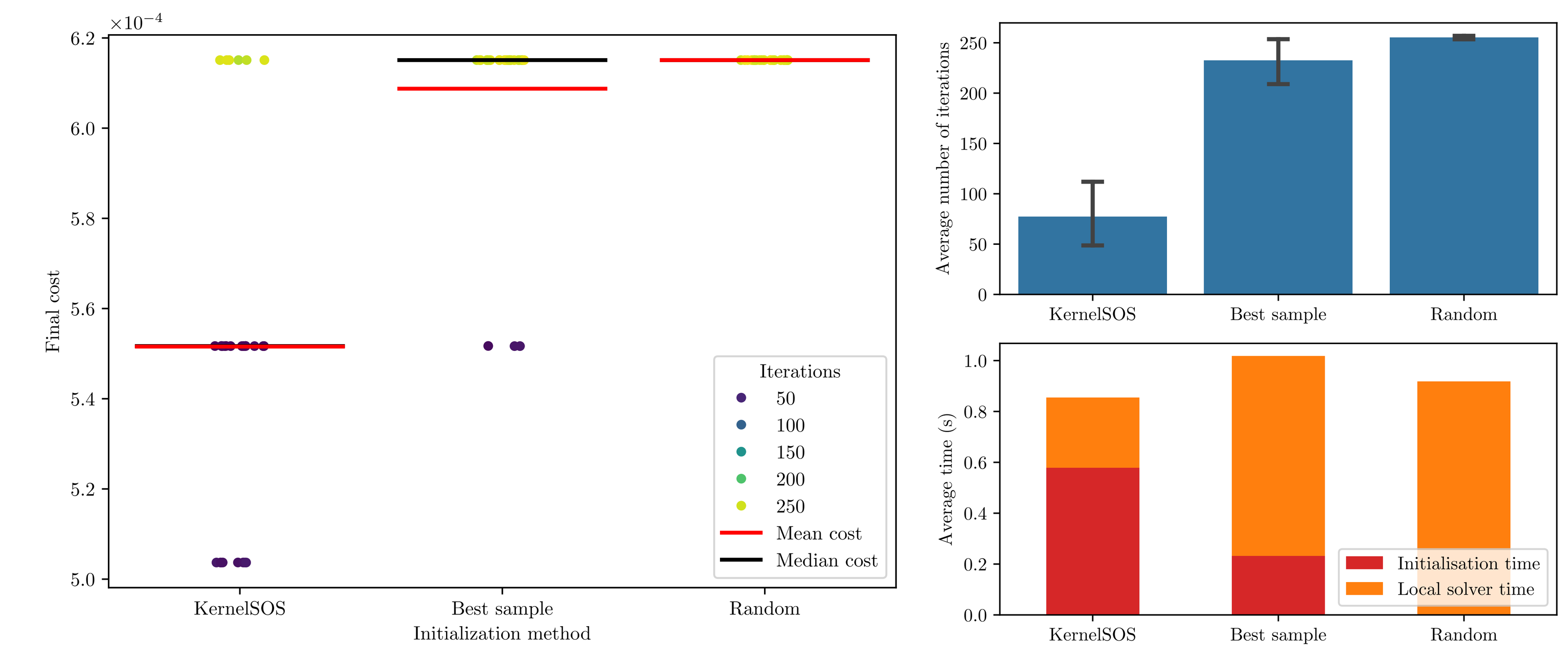
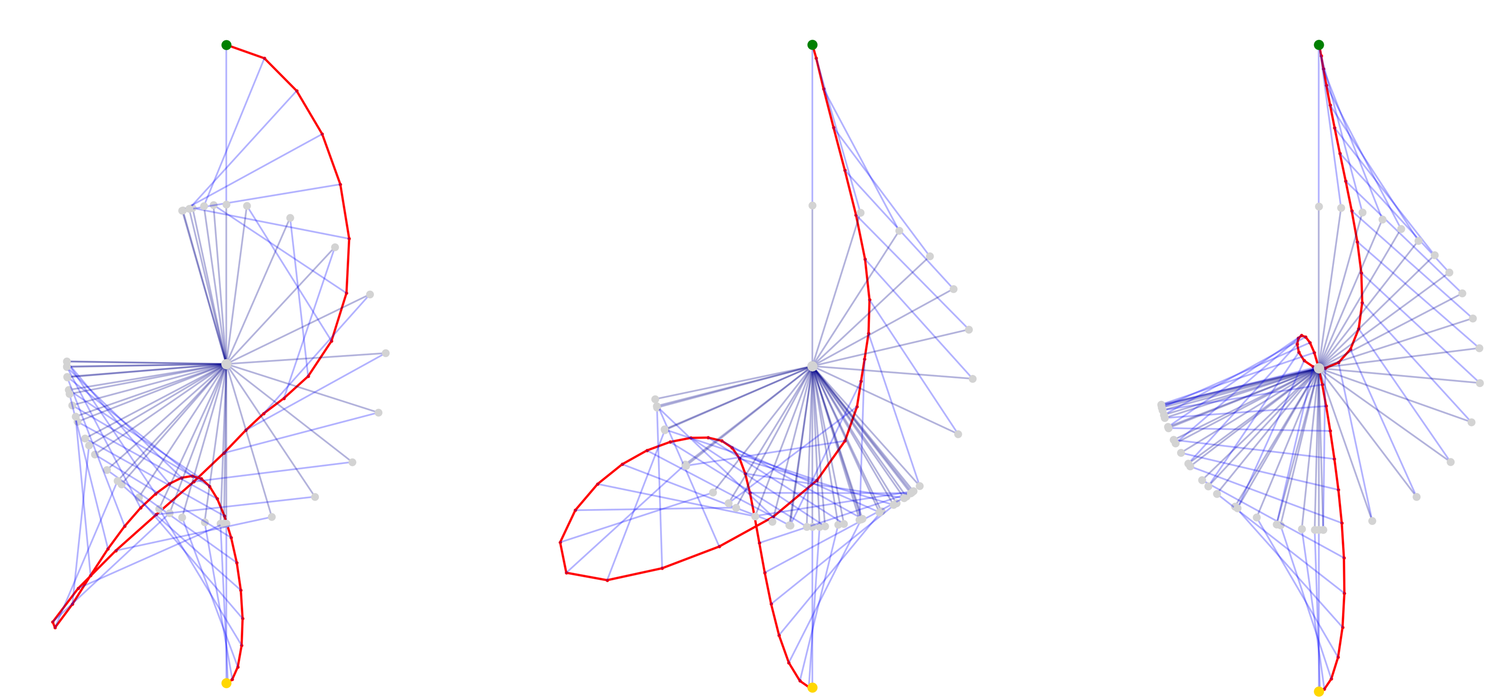


Figure 4. Comparison of the performance of FDDP depending on the initialization.

Discovering improved trajectories

KernelSOS can discover trajectories that are not findable by FDDP alone, as it explores the space more globally.



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

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

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

References

[1] Antoine Groudiev, Fabian Schramm, Éloïse Berthier, Justin Carpentier, and Frederike Dümbgen. KernelSOS for global sampling-based optimal control and estimation via semidefinite programming. 2025. In preparation.

[2] Wilson Jallet, Antoine Bambade, Sarah El Kazdadi, Carpentier Justin, and Mansard Nicolas. Aligator, `https://github.com/Simple-Robotics/aligator`.

[3] Alessandro Rudi, Ulysse Marteau-Ferey, and Francis Bach. Finding global minima via kernel approximations. *Mathematical Programming*, 209(1):703–784, 2025.

antoine.groudiev@ens.psl.eu