Bayesian Optimization Overview

Marco Forgione

[1]IDSIA Dalle Molle Institute for Artificial Intelligence SUPSI-USI, Lugano, Switzerland

December 10, 2019

Motivations

Obtaining the predictive model for MPC is costly and time-consuming.

Typically, models are obtained through Physical modeling or Identification

- Requires domain knowledge and/or ad-hoc identification experiments
- A trade-off emerges between accuracy and complexity

It is often difficult to decide *a priori* how accurate the predictive model should be in order to achieve satisfactory closed-loop performance.

In this work:

- We introduce a data-driven framework aimed at finding the best model for MPC from calibration experiments
- We specialize this framework for a hierarchical MPC architecture often encountered in industrial applications

Motivations

Obtaining the predictive model for MPC is costly and time-consuming.

Typically, models are obtained through Physical modeling or Identification

- Requires domain knowledge and/or ad-hoc identification experiments
- A trade-off emerges between accuracy and complexity

It is often difficult to decide *a priori* how accurate the predictive model should be in order to achieve satisfactory closed-loop performance.

In this work:

- We introduce a data-driven framework aimed at finding the best model for MPC from calibration experiments
- We specialize this framework for a hierarchical MPC architecture often encountered in industrial applications

Motivations

Obtaining the predictive model for MPC is costly and time-consuming.

Typically, models are obtained through Physical modeling or Identification

- Requires domain knowledge and/or ad-hoc identification experiments
- A trade-off emerges between accuracy and complexity

It is often difficult to decide *a priori* how accurate the predictive model should be in order to achieve satisfactory closed-loop performance.

In this work:

- We introduce a data-driven framework aimed at finding the best model for MPC from calibration experiments
- We specialize this framework for a hierarchical MPC architecture often encountered in industrial applications

Bayesian Optimization

Overview

One of the best *off-the-shelf* global optimization algorithms

- Iteratively updates a stochastic surrogate model of the unknown $J(\theta)$ via Bayesian inference. Typically, a Gaussian Process (GP)
- Balances exploitation and exploration by optimizing an acquisition function $A(\theta)$ instead of the surrogate model directly
- The acquisition function $A(\theta)$ favors points with estimated good performance $\rightarrow$ exploitation and/or high variance $\rightarrow$ exploration
- The acquisition function $A(\theta)$ is (relatively) cheap to evaluate. It is a mathematical object!

Bayesian Optimization

Overview

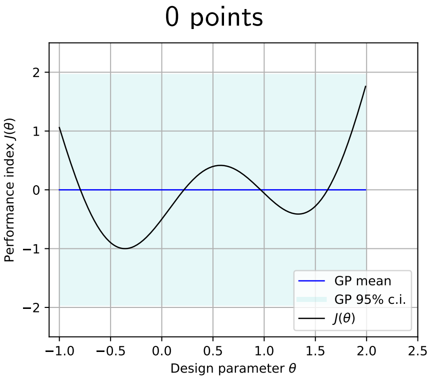
One of the best *off-the-shelf* global optimization algorithms

- Iteratively updates a stochastic surrogate model of the unknown $J(\theta)$ via Bayesian inference. Typically, a Gaussian Process (GP)
- Balances exploitation and exploration by optimizing an acquisition function $A(\theta)$ instead of the surrogate model directly
- The acquisition function $A(\theta)$ favors points with estimated good performance $\rightarrow$ exploitation and/or high variance $\rightarrow$ exploration
- The acquisition function $A(\theta)$ is (relatively) cheap to evaluate. It is a mathematical object!

Bayesian Optimization

Gaussian Process

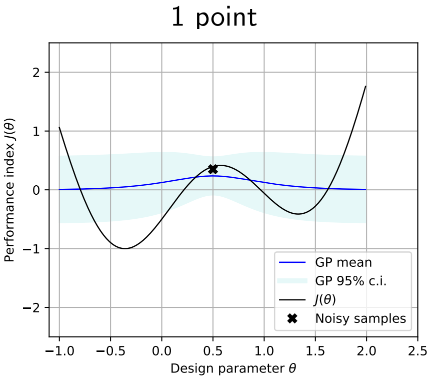
- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\text{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



Bayesian Optimization

Gaussian Process

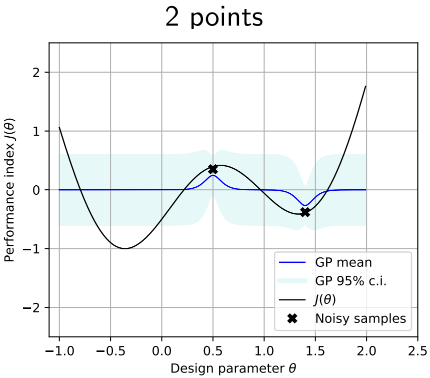
- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\mathrm{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



Bayesian Optimization

Gaussian Process

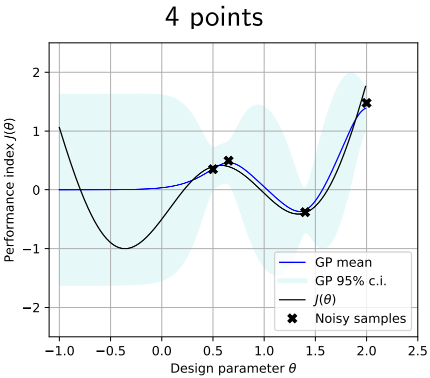
- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\mathrm{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



Bayesian Optimization

Gaussian Process

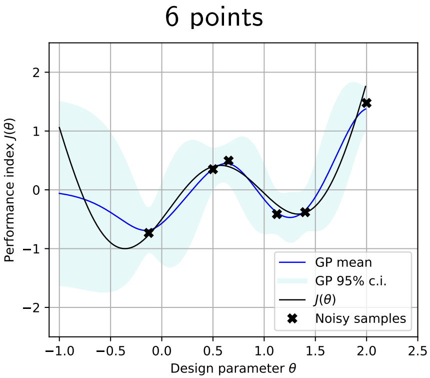
- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\text{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



Bayesian Optimization

Gaussian Process

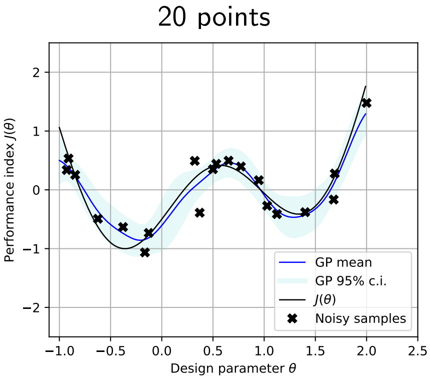
- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\mathrm{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



Bayesian Optimization

Gaussian Process

- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\mathrm{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



Bayesian Optimization

Acquisition function

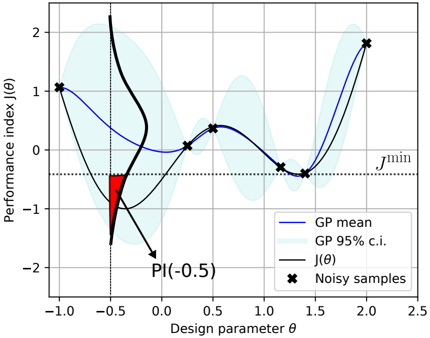
The GP provides the probability distribution of $J(\theta)$ for each parameter θ.
This probability is used to define an acquisition function, *e.g.*,

Probability of Improvement

$$A(\theta) = \mathrm{PI}(\theta) = p(J(\theta) \leq J^{\min})$$

Expected improvement

$$A(\theta) = \mathrm{EI}(\theta) = \mathbb{E}[\max\left(0, J^{\min} - J(\theta)\right)]$$



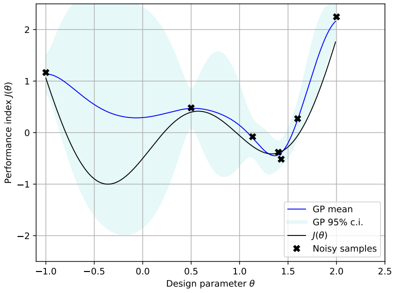
Bayesian Optimization

Overview

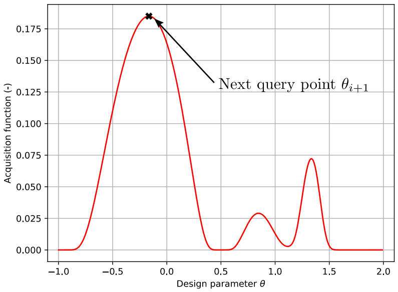
Steps of BO: for $i = 1, 2, \ldots i_{\max}$

1. **Execute** experiment with θ_i, measure $J_i = J(\theta_i) + e_i$
2. **Update** the GP model $\theta \to J(\theta)$ with (θ_i, J_i)
3. **Construct** acquisition function $A(\theta)$
4. **Maximize** $A(\theta)$ to obtain next query point θ_{i+1}

GP at iteration i



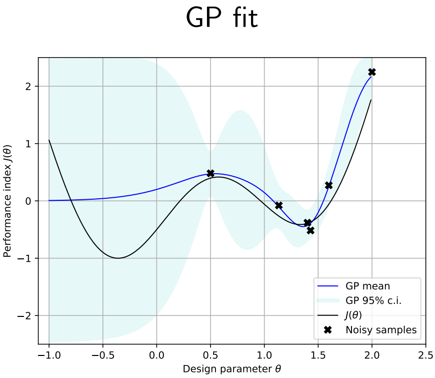
$A(\theta)$ at iteration i

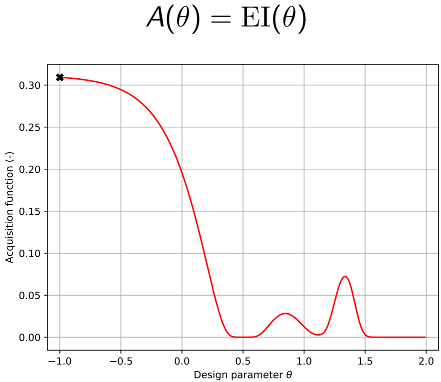


Bayesian Optimization

Example

iteration 6

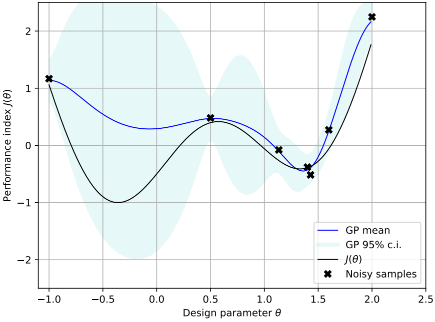




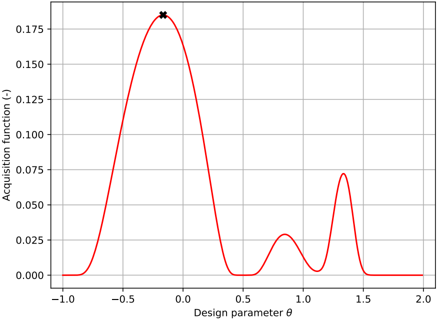
Bayesian Optimization
Example

iteration 7

GP fit

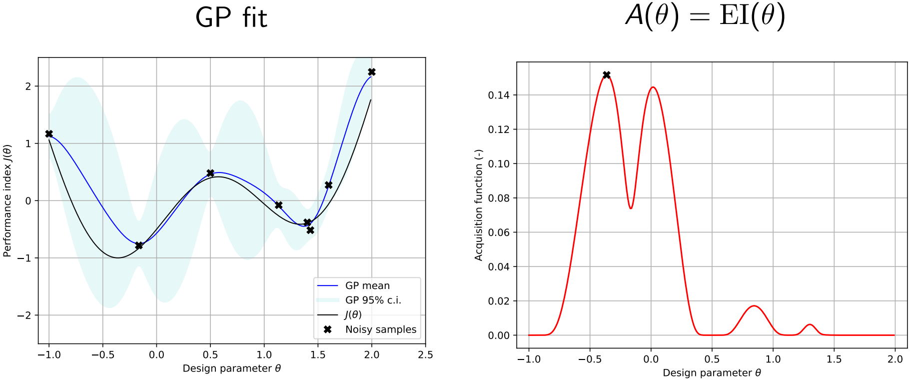


$A(\theta) = \mathrm{EI}(\theta)$



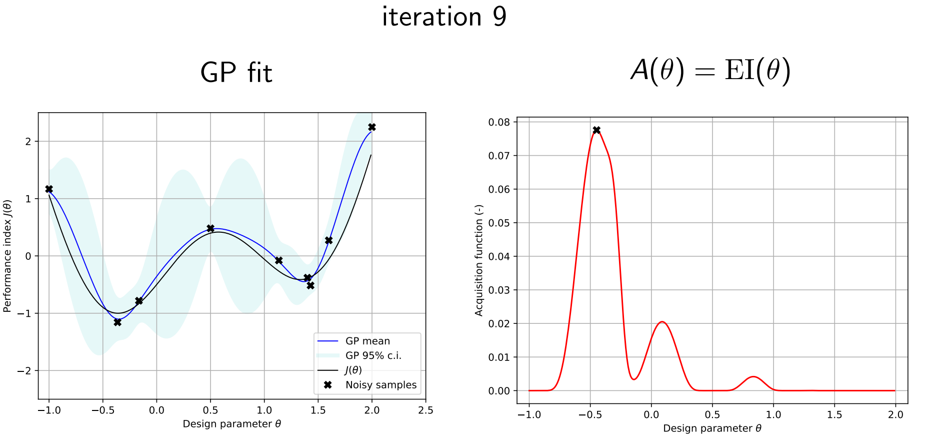
Bayesian Optimization
Example

iteration 8



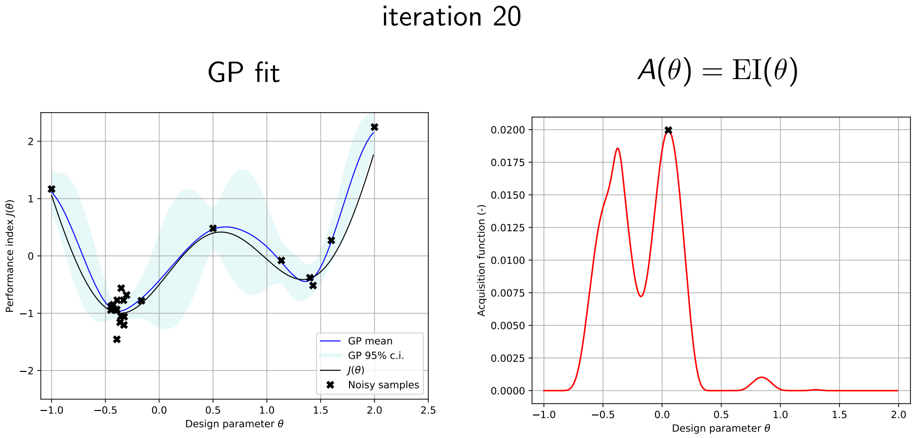
Bayesian Optimization

Example



Bayesian Optimization

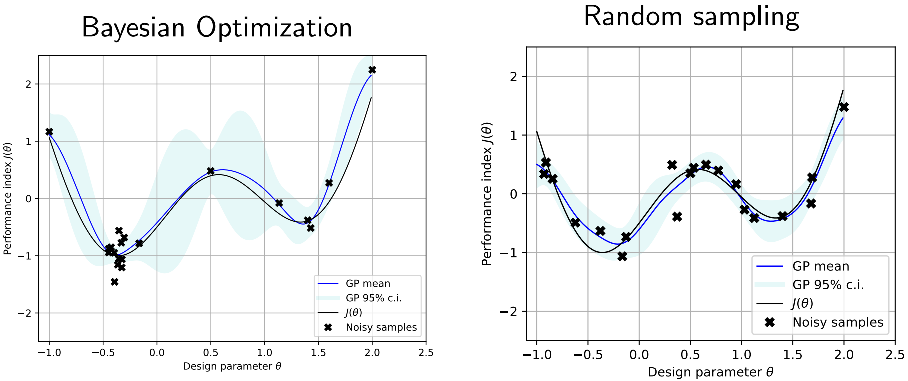
Example



Bayesian Optimization

Example

iteration 10



Thank you.
Questions?