Performance-oriented model learning for data-driven MPC design

Dario Piga[1], Marco Forgione[1], Simone Formentin[2], and Alberto Bemporad [3]

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

[2]Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy

[3]IMT School for Advanced Studies Lucca, Lucca, Italy

58th IEEE Conference on Decision and Control
Nice, France

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

In this work:

- We consider the model as a design parameter and tune it on calibration experiments to optimize a user-defined performance index
- We specialize this framework for a hierarchical MPC architecture often encountered in industrial applications
- Can be seen as an extension of Identification for Control

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

In this work:

- We consider the model as a design parameter and tune it on calibration experiments to optimize a user-defined performance index
- We specialize this framework for a hierarchical MPC architecture often encountered in industrial applications
- Can be seen as an extension of Identification for Control

Motivations

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

Typically, models are obtained through Physical modeling or Identification

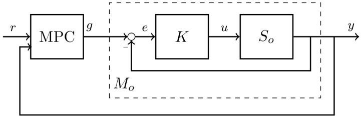
- Requires domain knowledge and/or ad-hoc identification experiments
- A trade-off emerges between accuracy and complexity

In this work:

- We consider the model as a design parameter and tune it on calibration experiments to optimize a user-defined performance index
- We specialize this framework for a hierarchical MPC architecture often encountered in industrial applications
- Can be seen as an extension of Identification for Control

Control architecture

We consider the Reference Governor architecture for system S_o



1. An inner controller K handles fast dynamics
2. An outer MPC takes care of constraints and performance specs

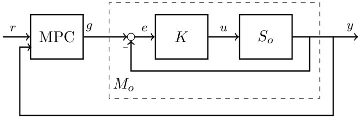
MPC requires a model M of the inner loop M_o. Existing approaches:

- Build model S for S_o, design $K \Rightarrow M = \texttt{feedback}(SK, I)$
- Direct identification of K targeting a reference model M (VRFT)

In our work, M and K are tuned simultaneously with a data-driven global optimization approach.

Control architecture

We consider the Reference Governor architecture for system S_o



1. An inner controller K handles fast dynamics
2. An outer MPC takes care of constraints and performance specs

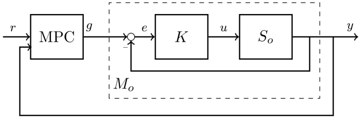
MPC requires a model M of the inner loop M_o. Existing approaches:

- Build model S for S_o, design $K \Rightarrow M = \texttt{feedback}(SK, I)$
- Direct identification of K targeting a reference model M (VRFT)

In our work, M and K are tuned simultaneously with a data-driven global optimization approach.

Control architecture

We consider the Reference Governor architecture for system S_o



1. An inner controller K handles fast dynamics
2. An outer MPC takes care of constraints and performance specs

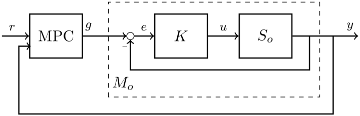
MPC requires a model M of the inner loop M_o. Existing approaches:

- Build model S for S_o, design $K \Rightarrow M = \texttt{feedback}(SK, I)$
- Direct identification of K targeting a reference model M (VRFT)

In our work, M and K are tuned simultaneously with a data-driven global optimization approach.

Control architecture

Inner Loop Controller



The inner controller K generates the system input u.
It is designed to handle fast dynamics

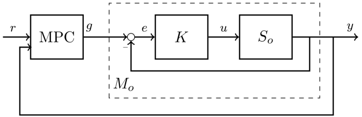
- Stabilize inner loop M
- Reject fast system disturbances

It is often as simple as a PID. . .

$$K(z, \theta) = \theta_P + \theta_I T_s \frac{1}{z-1} + \theta_D \frac{N_d}{1 + N_d T_s \frac{1}{z-1}}$$

Control architecture

Model Predictive Controller



The outer MPC generates the reference g for the inner loop M_o using a model $M(\theta) : g \to \left[\begin{smallmatrix} y \\ u \end{smallmatrix}\right]$

$$\xi_{t+1} = A_M \xi_t + B_M g_t$$
$$\begin{bmatrix} y_t \\ u_t \end{bmatrix} = C_M \xi_t + D_M g_t,$$

to handle constraints and enhance performance, according to

$$\min_{\{g_{t+k|t}\}_{k=1}^{N_p}} \sum_{k=1}^{N_p} \left\| y_{t+k|t} - r_{t+k} \right\|_{Q_y}^2 + \left\| u_{t+k|k} - u_{t+k-1|t} \right\|_{Q_{\Delta u}}^2$$

s.t. model equations, constraints on g, y, u, Δu

Performance-oriented tuning

Overview

To implement the performance-oriented tuning, we need to define

- Tunable design parameters of the inner controller K and of the inner loop model M collected in a design vector θ, with $\theta \in \Theta$.
- An experimental procedure to perform calibration experiments representative of the intended closed-loop operation
- A closed-loop performance index J defined in terms of measured input/outputs during the calibration experiment: $J = J(y_{1:T}, u_{1:T};\ \theta)$

MPC calibration is seen as a global optimization problem:

$$\theta^{\mathrm{opt}} = \arg\min_{\theta \in \Theta} J(y_{1:T}, u_{1:T};\ \theta)$$

each (noisy) function evaluation correspond to a calibration experiment.

Problem is tackled using efficient global optimization algorithms.

Performance-oriented tuning

Overview

To implement the performance-oriented tuning, we need to define

- Tunable design parameters of the inner controller K and of the inner loop model M collected in a design vector θ, with $\theta \in \Theta$.
- An experimental procedure to perform calibration experiments representative of the intended closed-loop operation
- A closed-loop performance index J defined in terms of measured input/outputs during the calibration experiment: $J = J(y_{1:T}, u_{1:T};\ \theta)$

MPC calibration is seen as a global optimization problem:

$$\theta^{\mathrm{opt}} = \arg\min_{\theta \in \Theta} J(y_{1:T}, u_{1:T};\ \theta)$$

each (noisy) function evaluation correspond to a calibration experiment.

Problem is tackled using efficient global optimization algorithms.

Performance-oriented tuning

Overview

To implement the performance-oriented tuning, we need to define

- Tunable design parameters of the inner controller K and of the inner loop model M collected in a design vector θ, with $\theta \in \Theta$.
- An experimental procedure to perform calibration experiments representative of the intended closed-loop operation
- A closed-loop performance index J defined in terms of measured input/outputs during the calibration experiment: $J = J(y_{1:T}, u_{1:T};\ \theta)$

MPC calibration is seen as a global optimization problem:

$$\theta^{\text{opt}} = \arg\min_{\theta \in \Theta} J(y_{1:T}, u_{1:T};\ \theta)$$

each (noisy) function evaluation correspond to a calibration experiment.

Problem is tackled using efficient global optimization algorithms.

Bayesian Optimization

Overview

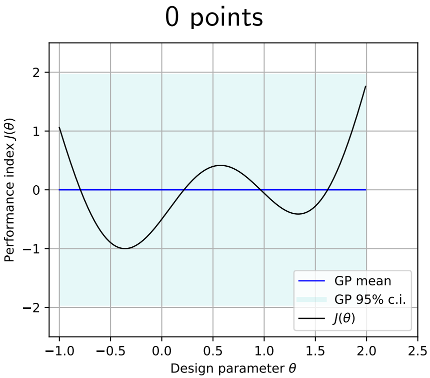
One of the best *off-the-shelf* global optimization algorithms

- 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

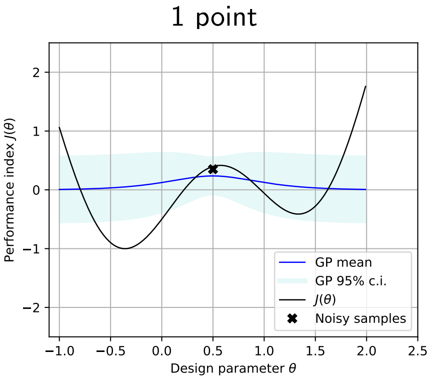
- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\text{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



Bayesian Optimization

Gaussian Process

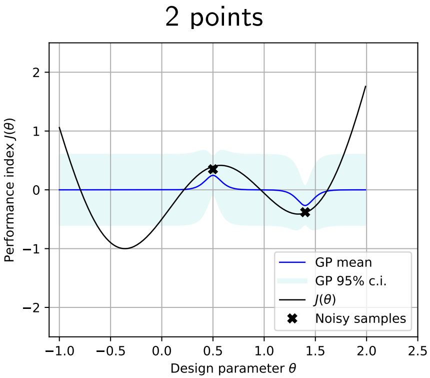
- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\text{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



Bayesian Optimization

Gaussian Process

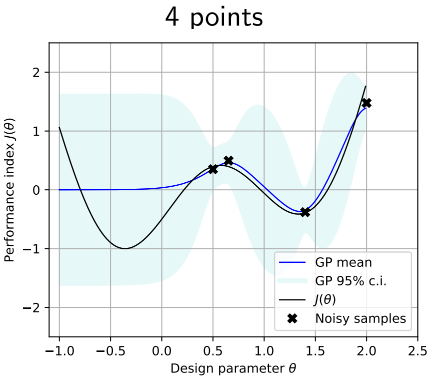
- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\mathrm{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



Bayesian Optimization

Gaussian Process

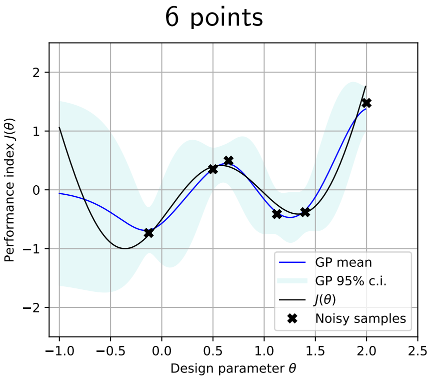
- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\text{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



Bayesian Optimization

Gaussian Process

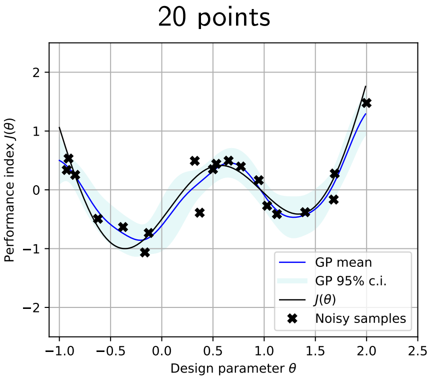
- The function $J(\theta)$ assumed Gaussian with prior mean $E[J(\theta)] = \mu(\theta)$ and covariance $\mathrm{cov}[J(\theta_1), J(\theta_2)] = \kappa(\theta_1, \theta_2)$.
- The posterior mean and covariance given a new observation (θ_i, J_i) is obtained in closed form



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

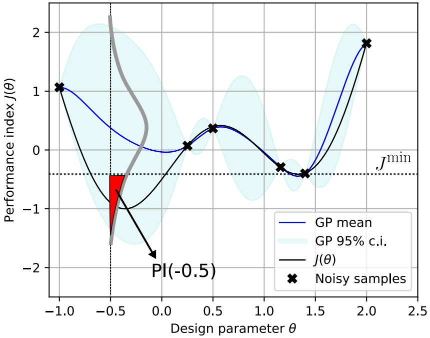
The GP provides the probability distribution of $J(\theta)$ for each parameter θ.
This probability is used to define an acquisition function, *e.g.*,

Probability of Improvement

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

Expected Improvement

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



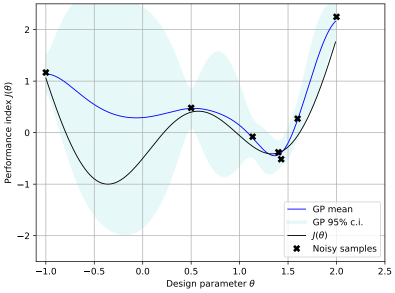
Bayesian Optimization

Overview

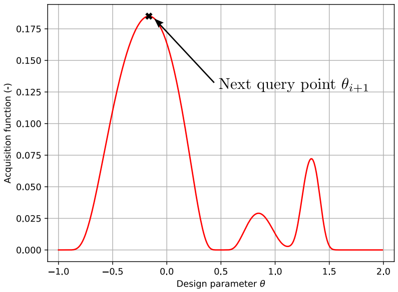
Steps of BO: for $i = 1, 2, \ldots i_{\max}$

1. **Execute** experiment with θ_i, measure $J_i = J(\theta_i) + e_i$
2. **Update** the GP model $\theta \rightarrow J(\theta)$ 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

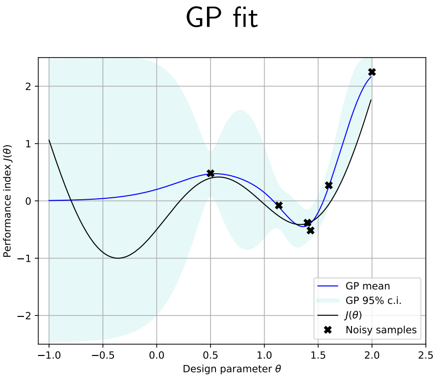


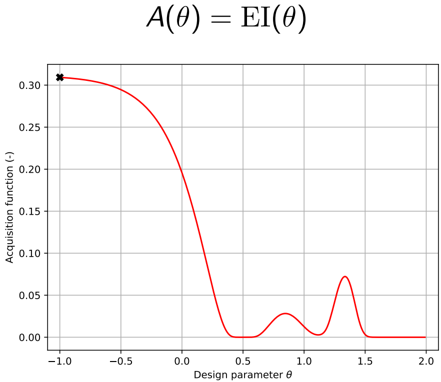
$A(\theta)$ at iteration i



Bayesian Optimization
Example

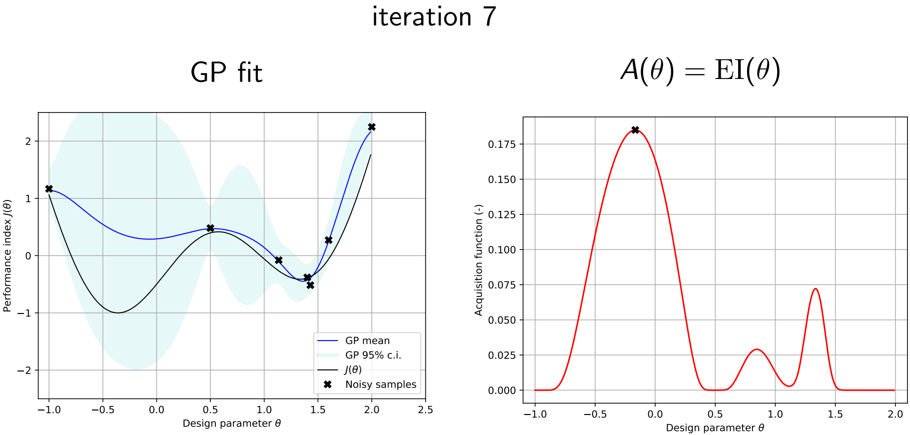
iteration 6





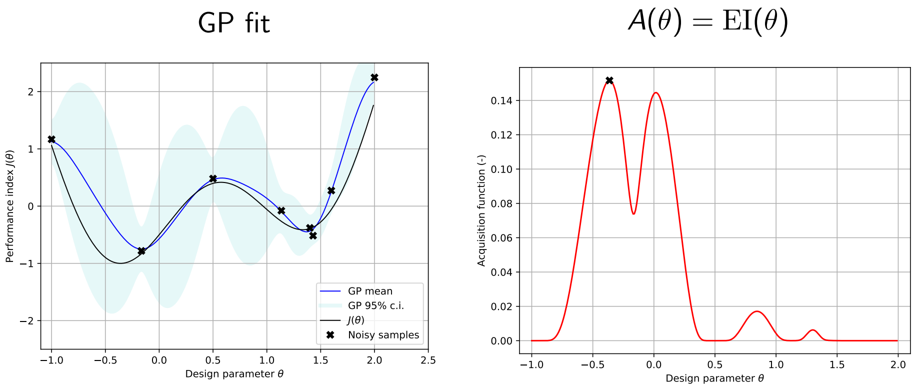
Bayesian Optimization

Example



Bayesian Optimization
Example

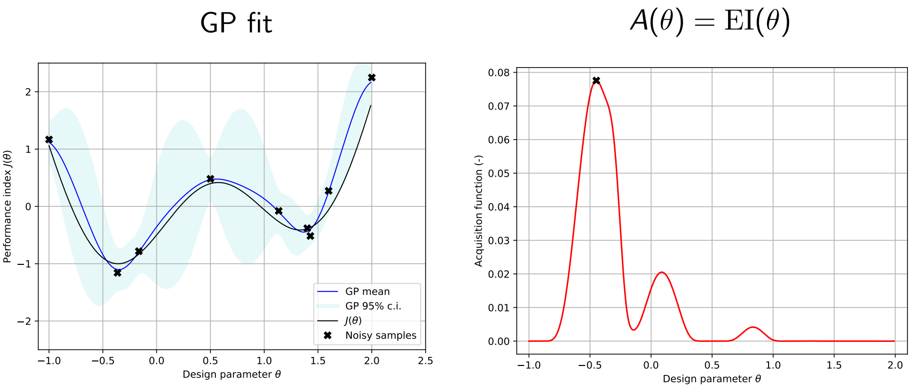
iteration 8



Bayesian Optimization

Example

iteration 9

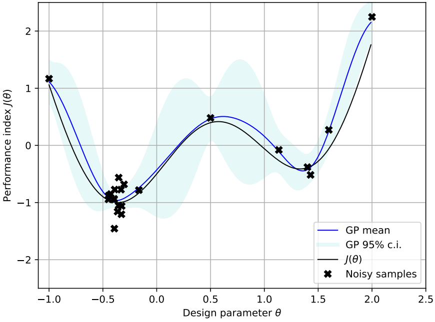


Bayesian Optimization

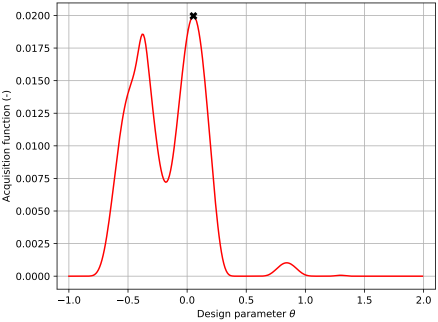
Example

iteration 20

GP fit

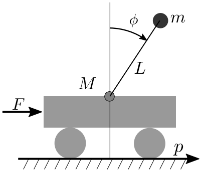


$A(\theta) = \mathrm{EI}(\theta)$



Simulation Example

Cart-pole system



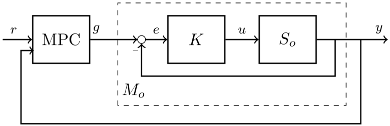
Objective: starting at $p_0 = 0$, $\phi_0 = 15^o$

1. stabilize pendulum in the upright unstable equilibrium $\phi = 0$
2. keep cart position p in $[-1 \;\; 1]$ m

$$J = \log\left[\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{10}|p_t| + \frac{9}{10}|\phi_t|\right)\right] + \sum_{t=1}^{T}\ell(|p_t| - 1)$$

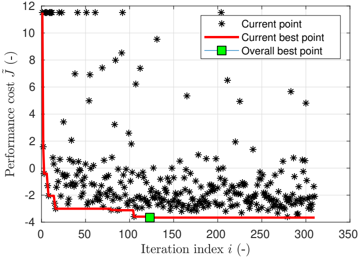
- State $x = [p \; \dot{p} \; \phi \; \dot{\phi}]^\top$
- Output $y = [p \; \phi]^\top$ corrupted by white measurement noise
- Input $u = F$ with fast additive disturbance (10 rad/sec)
- Control structure: inner PID on ϕ, outer MPC as Reference Governor

- Design parameters: PID gains, model M, prediction horizon N_p
- Calibration experiments of 10 s
- $T_s^{\mathrm{PID}} = 5\ \mathrm{ms}$, $T_s^{\mathrm{MPC}} = 50\ \mathrm{ms}$

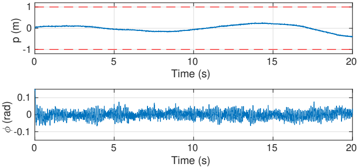


Simulation Example

Performance index vs. iteration



Optimal trajectory



- For increasing iteration i, more and more points have "low" cost
- Optimal trajectory satisfies constraints $p \in [-1 \;\; 1]$ m
- Achieved performance is better than our manual tuning

Conclusions

An experiment-driven MPC calibration approach based on global optimization

- Predictive model explicitly tuned for the performance index
- Applied to a hierarchical Reference Governor structure

Current/future works

- Application to robotic systems with PID+feedback linearization
- Tuning of MPC parameters such as cost-function weight matrices, observer gains, sampling time, solver accuracy for embedded MPC
- Analyze generalization properties with respect to objectives not considered in the calibration phase
- Find parametrized solution with respect to different reference trajectories

Conclusions

An experiment-driven MPC calibration approach based on global optimization

- Predictive model explicitly tuned for the performance index
- Applied to a hierarchical Reference Governor structure

Current/future works

- Application to robotic systems with PID+feedback linearization
- Tuning of MPC parameters such as cost-function weight matrices, observer gains, sampling time, solver accuracy for embedded MPC
- Analyze generalization properties with respect to objectives not considered in the calibration phase
- Find parametrized solution with respect to different reference trajectories

Thank you.
Questions?