Baysian Optimization in MPC

Frederic Dlugi

Universität zu Lübeck
frederic.dlugi@student.uni-luebeck.de

December 2, 2021

Abstract

When using MPC to control a vehicle it is often neccesary to fine tune vehicle parameter for the model to match reality. This is a time consuming process that often relies on trial and error or grid search of the parameter space. In this paper we evaluate the use of Bayesian Optimization to tackle this problem. We simulate vehicle dynamics of a simple bicycle model with two parameters. We try to find the real vehicle parameters by optimizing controller performance with different parameters using Bayesian Optimization.

I. Introduction

We start by introducing bayesian optimization and MPC on a high level. This is done to let you know what I understand when talking about these algorithms.

i. What is Bayesian Optimization?

Bayesian optimization is a strategy for global optimization of backbox functions. (Alg. 1) It therefore does not require the computation of gradients, but works best on continuous functions. It is best suited for optimizing functions, where each evaluation takes a long time. The number of input dimensions for bayesian optimization is typically less then 20. [1] Bayesian Optimization uses an aquisition function that operates on a gaussian process of the evaluations. In this paper we evaluate three different acquisition functions: Expected Improvement (Alg. 2), Probability of Improvement (Alg. 3) and Upper Confidence Bound (Alg. 4). We use the Matern ($\nu = 2.5$) kernel for all experiments. We also vary the α Parameter of the Gaussian Process to smooth out the cost landscape.

Algorithm 1 Bayesian Optimization

```
f ← black box function
initPos ← initial positions
evaluations ← f(initPos)
for i = 1 → N do
    α ← acquisitionFunction(evaluations)
    nextPos ← argmax(α)
    evaluations.append(f(nextPos))
result ← max(evaluations)
```

Algorithm 2 Expected Improvement

```
μ, σ ← gp.predict(x)
z ← (μ − y_max − xi)/σ
result ← (μ − y_max − xi) ∗ cdf(z) + σ ∗ pdf(z)
```

Algorithm 3 Probability of Improvement

```
μ, σ ← gp.predict(x)
z ← (μ − y_max − xi)/σ
result ← cdf(z)
```

Algorithm 4 Upper Confidence Bound

```
μ, σ ← gp.predict(x)
result ← μ + κ ∗ σ
```

ii. What is MPC?

MPC is an acronym for model predicive control. It replaces classical control algorithms that usually work in an offline manner with an optimizer, that solves the optimal control problem for a receding horizon (Figure 1). Each action taken is the first action of the optimal control plan. The plan gets updated continuously (Figure 2), this is called receding horizon approach.

Figure 1: *Blockdiagram of MPC algorithm*

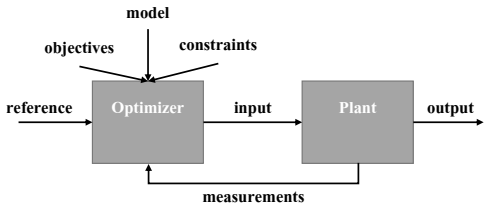
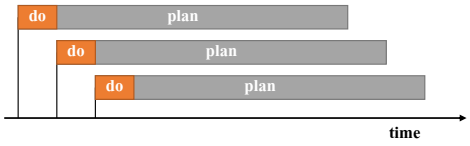


Figure 2: *Plan horizon of MPC algorithm*



iii. Kinematic Bicycle Model

We use a kinematic bicycle model defined by algorithm 5 and 6. We try to estimate the L_r and L_f in this paper.

Algorithm 5 Kinematic Bicycle Model

$v \leftarrow$ speed (input)
$\omega \leftarrow$ steering angle rate (input)
$x_c, y_c \leftarrow$ center position
$\theta \leftarrow$ absolute angle
$\delta \leftarrow$ steering angle
$\beta \leftarrow$ side slip factor
$L_r \leftarrow$ Distance center to rear wheel
$L_f \leftarrow$ Distance center to font wheel
$L \leftarrow L_r + L_f$

Algorithm 6 Update Bicycle Model

$\dot{x_c} \leftarrow v * \cos(\theta + \beta)$
$\dot{y_c} \leftarrow v * \sin(\theta + \beta)$
$\dot{\theta} \leftarrow v * \cos(\beta) * \tan(\delta)/L$
$\dot{\delta} \leftarrow \omega$
$\beta \leftarrow \text{atan2}(L_r * \tan(\delta), L)$

II. Experiments

Experiment Setup

References

[1] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.