GP-based Model Predictive Control*

Dimitrios Gkoutzos[1], Luzia Knödler[2] and Lucas Rath[3]

***Abstract*— Describe topic and relevance in a few sentences so that the reader is motivated to read the whole paper.**

I. INTRODUCTION

Model predictive control (MPC) is a popular control strategy which uses a dynamic plant model to obtain the control input that optimizes future reactions of the plant [4]. The performance of MPC depends highly on how well the model captures the dynamics of the plant [3]. But the identification of such an *a priori* model can be challenging and the dynamics of the plant could also change during the application [3], [7]. Therefore, a simple and fixed nominal model of the plant can be used in combination with a learned disturbance model. The disturbance model represents the error between the observed behaviour of the plant and the behaviour of the nominal model [7]. It can be modelled as a Gaussian Process (GP) regression which is a probabilistic, non-parametric model [4]. GPs have the advantage of characterizing the prediction uncertainities [4]. The mean estiamte of a GP can also be used to model the full dynamics of the plant and not only the model error. This approach was applied to a cart pole swing-up environment and an autonomous racing task in [8]. To reduce high computational costs they choose sparce spectrum GPs. Kocijan et al. [4] use an offline-identified GP model instead. Another alternative is the generation of local GPs (LGPs) where for each subspace of the GP input space different GPs are identified. While Nguyen-Tuong et al. [6] and Meier et al. [5] identify many LGPs, Ostafew et al. [7] compute one single LGP based on data within a sliding window. Other applications of GPs in a MPC framework are .

In this report we present the results of our project within the course "Statistical Learning and Stochastic Control". First, our literature research on GP-based MPC is summarized. Then a short introduction to the theory of MPC and GPs is given. Later, two examples which's implementation was part of the project are introduced in Section III and the results are discussed in Section IV.

We start by defining the used noation below.

Bold lowercase letters are used for vectors $\boldsymbol{x} \in \mathcal{R}^n$ and bold uppercase letters for matrices $\boldsymbol{X} \in \mathcal{R}^{n\times m}$.

*Project within the course Statistical Learning and Stochastic Control, University of Stuttgart, January 7, 2020.

[1]Dimitrios Gkoutzos is a student of the Master study programm Engineering Cybernetics, University of Stuttgart, `email???`

[2]Luzia Knödler is a student of the Master study program Engineering Cybernetics, University of Stuttgart, `luzia.knoedler@web.de`

[3]Lucas Rath is a student of the Master study program Engineering Cybernetics, University of Stuttgart, and of the Master study program Systems, Control and Mechatronics, Chalmers University of Technology, `lucasrm25@gmail.com`

Introduce topic and describe motivation and relevance of problem/topic.

In this paper we give an introduction to the results presented in paper(s) [2]. We present the main results, discuss ideas and illustrate the results with simulations.

Notation. Define notation.

II. BACKGROUND

In this section a revision of GPs for our application and necessary background information on MPC are presented. The revison of GPs follows Kabzan et al. [3] as well as Rasmussen and Williams [9].

As mentioned in the Introduction, a GP is used to identify the disturbance $\boldsymbol{d_{true}}$, which describes the error between the nominal plant dynamics f_{nom} and the true dynamics f_{true}. Thus, the true system equations are given by

$$\begin{aligned} \boldsymbol{x_{k+1}} &= f_{true}(\boldsymbol{x_k}, \boldsymbol{u_k}) \\ &= f_{nom}(\boldsymbol{x_k}, \boldsymbol{u_k}) + \boldsymbol{B_d}\big(\boldsymbol{d_{true}}(\boldsymbol{z_k}) + \boldsymbol{w}\big), \end{aligned} \tag{1}$$

with $\boldsymbol{z_k} = \boldsymbol{B_z}\boldsymbol{x_k}$ ($\boldsymbol{B_z} \in \mathcal{R}^{n_z\times n}$ and $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{\Sigma_w})$ the gaussian measurement noise, where $\boldsymbol{\Sigma_w} = diag(\sigma_1^2, \cdots, \sigma_{n_d}^2)$. The disturbance $\boldsymbol{d} = \boldsymbol{d_{true}}(\boldsymbol{z_k}) + \boldsymbol{w_k}$ is identified using input and output data pairs $(\boldsymbol{z_k}, \boldsymbol{y_k} = \boldsymbol{d_{true}}(\boldsymbol{z_k}) + \boldsymbol{w_k})$ which are saved in a dictionary $\mathcal{D}$ of length m

$$\begin{aligned} \mathcal{D} = \{&\boldsymbol{Z} = [\boldsymbol{z_1}, \cdots, \boldsymbol{z_m}] \in \mathbb{R}^{n_z\times m}, \\ &\boldsymbol{Y} = [\boldsymbol{y_1}, \cdots, \boldsymbol{y_m}] \in \mathbb{R}^{n_d\times m}\}. \end{aligned} \tag{2}$$

If $m > m_{max}$, the dictionary is updated by removing the oldest data pair.

Each output dimension $a \in \{1, \cdots, n_d\}$ is treated as a different GP with the kernel $\boldsymbol{k^a}$, which results in the posterior distribution mean $\boldsymbol{\mu^a}(\boldsymbol{z})$ and variance $\boldsymbol{\Sigma^a}(\boldsymbol{z})$

$$\boldsymbol{\mu^a}(\boldsymbol{z}) = \boldsymbol{k^a_{zZ}}(\boldsymbol{K^a_{ZZ}} + \boldsymbol{I}\boldsymbol{\sigma_a})^{-1}[\boldsymbol{Y}]_{.,i}, \tag{3}$$

$$\boldsymbol{\Sigma^a}(\boldsymbol{z}) = k^a_{zz} - \boldsymbol{k^a_{zZ}}(\boldsymbol{K^a_{ZZ}} + \boldsymbol{I}\boldsymbol{\sigma_a})^{-1}\boldsymbol{k^a_{Zz}} \tag{4}$$

in dimension a for a test point $\boldsymbol{z}$. The expressions k^a_{zz}, $\boldsymbol{k^a_{zZ}}$, $\boldsymbol{k^a_{Zz}}$ and $\boldsymbol{K^a_{ZZ}}$ are compact notations for $k^a(\boldsymbol{z}, \boldsymbol{z}) \in \mathbb{R}$, $[\boldsymbol{k^a_{zZ}}]_j = k^a(\boldsymbol{z}, z_j)$, $[\boldsymbol{k^a_{Zz}}]_j = k^a(z_j, \boldsymbol{z})$ and $[\boldsymbol{K^a_{ZZ}}]_{ij} = k^a(z_i, z_j)$, respectively. It holds $\boldsymbol{k^a_{Zz}} = (\boldsymbol{k^a_{zZ}})^T \in \mathbb{R}^m$.

For each output dimension a we make use of the squared exponentional kernel given by

$$k^a(\boldsymbol{z}, \bar{\boldsymbol{z}}) = \boldsymbol{\sigma_{f,a}}^2 \exp(-0.5(\boldsymbol{z} - \bar{\boldsymbol{z}})^T \boldsymbol{M}^{-1}(\boldsymbol{z} - \bar{\boldsymbol{z}})), \tag{5}$$

where $\boldsymbol{\sigma_{f,a}}^2$ and $\boldsymbol{M}$ are the squared output variance and the positive diagonal length scale matrix, respectively. The practical implementation of Gaussian process regression from page 19 of Rasmussen and Williams [9] which uses the Cholesky factorization to address the matrix inversion is applied. This results in the following algorithm

$$\boldsymbol{L^a} = \text{cholesky}(\boldsymbol{K}(\boldsymbol{Z},\boldsymbol{Z}) + \boldsymbol{\sigma_n}(\boldsymbol{a},\boldsymbol{a})^2 \boldsymbol{I}) \tag{6}$$

$$\boldsymbol{\alpha^a} = \boldsymbol{L}^T \backslash (\boldsymbol{L} \backslash \boldsymbol{Y}(:,a)), \tag{7}$$

where $[\boldsymbol{K}(\boldsymbol{Z},\boldsymbol{Z})]_{ij} = k(z_i, z_j)$ with $z_i, z_j \in \boldsymbol{Z}$. Thus, the predictive mean and variance are given by

$$\mu_y^a = \boldsymbol{K}(\boldsymbol{z},\boldsymbol{Z})^T \boldsymbol{\alpha^a} \tag{8}$$

$$\Sigma_y^a = k(\boldsymbol{z},\boldsymbol{z}) - \boldsymbol{v^a}^T \boldsymbol{v^a}, \tag{9}$$

with $\boldsymbol{v^a} = \boldsymbol{L^a}/\boldsymbol{K}(\boldsymbol{z},\boldsymbol{Z})$.
Combining the GPs of each dimension results in the multivariate GP approximation

$$\boldsymbol{d}(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\mu^d}(\boldsymbol{z}), \boldsymbol{\Sigma^d}(\boldsymbol{z})), \tag{10}$$

where the mean $\boldsymbol{\mu^d}(\boldsymbol{z}) \in \mathcal{R}^{n_d}$ and the variance $\boldsymbol{\Sigma^d}(\boldsymbol{z}) \in \mathcal{R}^{n_d \times n_d}$ are given by

$$\boldsymbol{\mu^d}(\boldsymbol{z}) = [\mu^1(\boldsymbol{z}), \cdots, \mu^{n_d}(\boldsymbol{z})]^T, \tag{11}$$

$$\boldsymbol{\Sigma^d}(\boldsymbol{z}) = \text{diag}([\Sigma^1(\boldsymbol{z}), \cdots, \Sigma^{n_d}(\boldsymbol{z})]). \tag{12}$$

The model considered for control including the identified disturbance d is given by

$$\boldsymbol{x_{k+1}} = f_{nom}(\boldsymbol{x_k}, \boldsymbol{u_k}) + \boldsymbol{B_d}\boldsymbol{d}(\boldsymbol{z_k}). \tag{13}$$

Model predictive control (MPC), receding horizon control or moving horizon control are all names for a control strategy which predicts the future dynamic behaviour within a finite prediction horizon and chooses the control input such that a perfomance functional is minimized [1]. Since the predicted behaviour is not equal to the system behaviour due to disturbances and model-plant mismatch, only the first input of the computed control input sequence is applied [1]. Using the new measurement one sampling time later, the procedure is repeated to find a new control sequence within the receding horizon.
Evaluating the GP model $\boldsymbol{d}(\boldsymbol{x_k}, \boldsymbol{u_k})$ results in a stochastic distribution, which leads to a stochastic distribution of the state $\boldsymbol{x}$. The distribution at each prediction step is assumed to be Gaussian with $\boldsymbol{x_k} \sim \mathcal{N}(\boldsymbol{\mu_k^x}, \boldsymbol{\Sigma_k^x})$. To evaluate the uncertainity over the prediction horizon N the mean and variance of $\boldsymbol{x}$ are propagated forward which results in

$$\boldsymbol{\mu_{k+1}^x} = f_{nom}(\boldsymbol{\mu_k^x}, \boldsymbol{u_k}) + \boldsymbol{B_d}\boldsymbol{\mu^d}(\boldsymbol{\mu_k^x}, \boldsymbol{u_k}) \tag{14}$$

$$\boldsymbol{\Sigma_{k+1}^x} = ????. \tag{15}$$

For more information on the propagation see ???? Appendix.

Thus, the MPC problem is given by

$$\min_{\tilde{u}} \Big(\sum_{k=0}^{N-1} f_o(t_k, \boldsymbol{\mu_k^x}, \boldsymbol{\Sigma_k^x}, \boldsymbol{\tilde{u}_k}, r(t)) \Big) + f_{end}(t_N, \boldsymbol{\mu_N^x}, \boldsymbol{\Sigma_N^x}, r(t_N)) \tag{16}$$

$$\boldsymbol{\mu_0^x} = \boldsymbol{x}(t) \tag{17}$$

$$\boldsymbol{\mu_{k+1}^x} = f_{nom}(\boldsymbol{\mu_k^x}, \boldsymbol{\tilde{u}_k}) + \boldsymbol{B_d}\boldsymbol{\mu_k^d} \tag{18}$$

$$\boldsymbol{\Sigma_{k+1}^x} = \nabla^T f_{nom}(\boldsymbol{\mu_k^x}, \boldsymbol{\tilde{u}_k}), \boldsymbol{\Sigma_{xdw}} \nabla f_{nom}(\boldsymbol{\mu_k^x}, \boldsymbol{\tilde{u}_k}), \tag{19}$$

$$\boldsymbol{\mu_k^x} \in \mathcal{X}, \tag{20}$$

$$\boldsymbol{u_k} \in \mathcal{U}, \tag{21}$$

where $\boldsymbol{\tilde{u}}(\dot{)} : [t, t+N] \rightarrow \mathcal{U}$ and $\tilde{u}_k$ refers to the k-th element of $\boldsymbol{\tilde{u}}$. The reference trajectory is given by $r(t)$. The input and state constraint sets $\mathcal{U}$ and $\mathcal{X}$ can be defined as box constraints according to Allgöwer et al. [1]

$$\mathcal{U} = \{u \in \mathcal{R}^p | \boldsymbol{u_{min}} \leq \boldsymbol{u} \leq \boldsymbol{u_{max}}\}, \tag{22}$$

$$\mathcal{X} = \{u \in \mathcal{R}^n | \boldsymbol{x_{min}} \leq \boldsymbol{x} \leq x_{max}\}. \tag{23}$$

According to the control problem, a cost function f_o and the final cost f_{end} are defined. The cost functions, which were used during this project, are further described for each example in Section III.

III. MAIN RESULTS

Two examples were considered during this project. An inverted pendulum and autonomous driving based on a single track model. Both examples are intruduced in this chapter.

A. Inverted Pendulum

The motion of the inverted pendulum is decribed by

$$(M_c + M_p)\ddot{x} + b\dot{x} + \frac{1}{2}M_p l \ddot{\theta}\cos\theta - \frac{1}{2}M_p l \dot{\theta}^2 \sin\theta = F \tag{24}$$

$$(I + M_p\Big(\frac{l}{2}\Big)^2)\ddot{\theta} - \frac{1}{2}M_p g l \sin\theta + M_p l \ddot{x}\cos\theta = 0. \tag{25}$$

A schematic drawing of the inverted pendulum including the parameters can be seen in Figure 1. The standard gravity on the surface of the earth g is set to be $g = 9.8$. The state space model with the states $[x, \dot{x}, \theta, \dot{\theta}]$ is given by

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \\ \dot{\theta} \\ \end{bmatrix} \tag{26}$$

Ideas, theorems, proofs and discussions .....

IV. EXAMPLES

Show and discuss simulation examples etc....

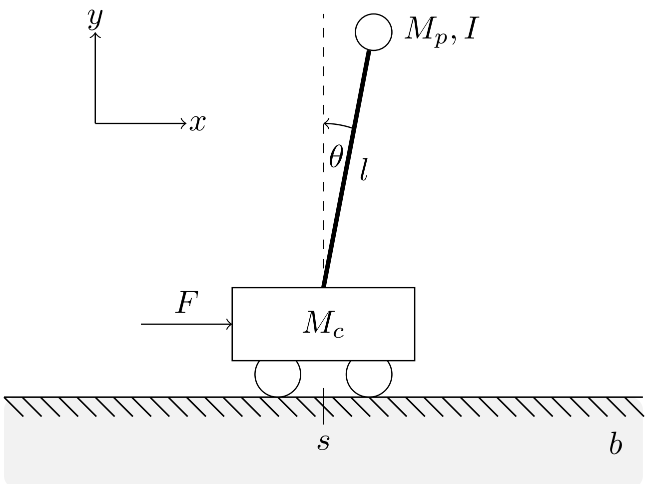

Fig. 1. A schematic drawing of the inverted pendulum on a carriage. The mass of the carriage and the pole are given by M_c and M_p, respectively. The pole is defined by its length l and its moment of intertia I. b is the friction coefficient between the carriage and the floor. The states of the state-space model are the carriage position s, its derivative $\dot{s}$, the pole angle with the vertical θ and its derivative $\dot{\theta}$. F is the applied force.

V. CONCLUSIONS

Summarize the main points (with more details than in the preceding introduction). The paper should not be between 4 and 8 pages.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

REFERENCES

[1] Frank Allgöwer, Rolf Findeisen, Christian Ebenbauer, et al. Nonlinear model predictive control.

[2] R. Brockett. The early days of geometric nonlinear control. *Automatica*, 50:2203–2224, 2014.

[3] Juraj Kabzan, Lukas Hewing, Alexander Liniger, and Melanie N Zeilinger. Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters*, 4(4):3363–3370, 2019.

[4] Juš Kocijan, Roderick Murray-Smith, Carl Edward Rasmussen, and Agathe Girard. Gaussian process model based predictive control. In *Proceedings of the 2004 American control conference*, volume 3, pages 2214–2219. IEEE, 2004.

[5] Franziska Meier, Philipp Hennig, and Stefan Schaal. Efficient bayesian local model learning for control. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2244–2249. IEEE, 2014.

[6] Duy Nguyen-Tuong, Jan R Peters, and Matthias Seeger. Local gaussian process regression for real time online model learning. In *Advances in Neural Information Processing Systems*, pages 1193–1200, 2009.

[7] Chris J Ostafew, Angela P Schoellig, and Timothy D Barfoot. Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4029–4036. IEEE, 2014.

[8] Benjamin Van Niekerk, Andreas Damianou, and Benjamin S Rosman. Online constrained model-based reinforcement learning. 2017.

[9] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.