Efficient and Simple Gaussian Process Supported Stochastic Model Predictive Control for Bioreactors using HILO-MPC

Bruno Morabito * **Johannes Pohlodek** * **Lena Kranert** *
Sebastián Espinel-Ríos *,** **Rolf Findeisen** ***

* *Laboratory for System Theory and Automatic Control, Otto von Guericke University Magdeburg, Germany*
** *Analysis and Redesign of Biological Networks, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany*
*** *Control and Cyber-Physical Systems Laboratory, TU Darmstadt, Germany*
(e-mail: rolf.findeisen@iat.tu-darmstadt.de)

Abstract: Model-based control of biotechnological processes is, in general, challenging. Often the processes are complex, nonlinear, and uncertain. Hence modeling tends to be complex and is often inaccurate. For this reason, non-model-based control strategies developed via flask, bench-scale, or pilot plant experiments are often applied in the biotechnology industry. Model-based control and optimization techniques can increase processes' performance and automation level, thereby decreasing costs and guaranteeing the desired specifications. These rely on a model of the process to make predictions and optimize the inputs to the plant. To improve the quality of the models, it is often helpful to use combined first principle and data-driven models together in a hybrid modeling approach which increases the model prediction capabilities. The residual uncertainty of the hybrid model should be taken into account in the control level to satisfy the process specifications and constraints. This paper proposes to use a stochastic model predictive control scheme that exploits a hybrid, Gaussian processes–based model. We outline the effectiveness of the stochastic model-based approach in combination with a suitable Kalman filter for state estimation considering an example biotechnological process. Furthermore, we underline that appropriate tools exist that allow the simple application of such methods even for the novice user. To do so, we use an open-source Python package — HILO-MPC, which allows the simple yet efficient formulation and solution of machine learning–supported optimal control and estimation problems.

Keywords: Predictive control, toolbox, machine learning, Gaussian process, uncertain process, biotechnology, stochastic model predictive control, HILO-MPC

1. INTRODUCTION

While in many fields, such as chemical engineering and autonomous driving, automation has significantly advanced, in biotechnology and especially biopharma, production processes, e.g., fermentations, are still mainly controlled with semi-empirical open-loop control policies (Mears et al., 2017; Roman and Olaru, 2018; Mitra and Murthy, 2021). A key operational objective in bio-engineering is to be able to ensure consistent productivity and product quality, possibly in a fully automated fashion (Luo et al., 2021). Furthermore, modern feedback control strategies are included within the guidelines of the quality-by-design concept encouraged by regulatory agencies (Rathore et al., 2021). Moreover, there is a strong motivation in the bioprocess community towards adopting the standards of Industry 4.0 and smart manufacturing, including aspects like online monitoring, control, and optimization (Sokolov et al., 2021). Three important points that need to be addressed to facilitate the use of these methods in industry are: Automated model identification/adaptation routines, uncertainty prediction methods, and simple software tools, to make these methods accessible to personnel with little or no control background.

Advanced model-based optimization and closed-loop control strategies such as model predictive control (MPC) Rawlings et al. (2017); Findeisen and Allgöwer (2002); Allgöwer et al. (2004) are one way to increase performance and reduce the cost of operations. It could also be used to avoid deviations in critical product quality attributes. Mathematical models are used to investigate systematically, develop economic benefits, and optimize biotechnological processes, such as the production of bioplastic (Koller et al., 2006; Duvigneau et al., 2021), lutein (Zhang et al., 2019) or the genetically enhanced synthesis of β-galactosidase (Tholudur and Ramirez, 1996), among others. However, due to noisy measurements, incomplete experimental data sets, oversimplifications or the lack of understanding of the underlying biological system, models of such biological processes tend to be inaccurate to a certain extent. Furthermore, it is often impossible to measure all variables necessary for closed-loop control online. This inaccuracy leads to poor system predictions, hence poor control performance.

⋆ BM and JP contributed equally to this paper. BM, SER and RF are affiliated with the International Max Planck Research School (IMPRS) for Advanced Methods in Process and Systems Engineering, Magdeburg. We acknowledge funding of the DIGIPOL project funded in the EU-ERDF scheme.

On the other hand, over the last decades, there has been an increasing interest in hybrid models, i.e., models containing first-principles and machine learning components (Oliveira, 2004; Von Stosch et al., 2014). Hybrid modeling believed to be a pragmatic approach that can fit within the quality-by-design paradigm and the FDA's Process Analytical Technology initiative (Sokolov et al., 2021). Hybrid models have the advantage of being more interpretable than pure machine learning models and requiring much fewer data to train. Moreover, they can still capture dynamics that are hard to capture with pure first principle models. A successful machine learning approach used for hybrid models is Gaussian processes (GPs) (Williams and Rasmussen, 2006). GPs have the advantage of providing the uncertainty on the prediction that can be used, e.g., to guarantee (with a certain probability) process specifications in the context of process optimization and control, which can be defined as process constraints.

In this paper, a hybrid model using Gaussian processes is used in a stochastic model predictive controller. Stochastic MPC is an MPC approach that considers uncertain processes with stochastic uncertainty (Heirung et al., 2018; Mesbah, 2018). Stochastic MPC can be more robust to uncertainty, guarantee constraint satisfaction (in probabilistic terms) and have a *dual-control* effect (cf. Mesbah (2018) and references therein). Nevertheless, the propagation of the uncertainty into the future and the constraints can be hard to handle. The propagation of even a simple probability distribution such as a Gaussian distribution through a nonlinear system results in a non-Gaussian distribution that is usually hard to describe.

Recently, in the field of bioreactor control, several stochastic MPC were proposed. They rely on Monte Carlo sampling (Bradford et al., 2021; Mowbray et al., 2021) or polynomial chaos expansion (Bradford et al., 2019b,a) for propagating the uncertainty. The previous methods can offer a good description of the uncertainty evolution, but they are complex and might be difficult to implement, especially for practitioners not familiar with stochastic control. In Hewing et al. (2020) a simpler and more tractable approximated stochastic problem based on the assumption that the future state evolution maintains a Gaussian distribution is used for an autonomous racing example. In this paper, we expand this approach also considering state estimation using an unscented Kalman filter. The approach is simpler than the Monte Carlo and polynomial chaos method previously mentioned, but it might nevertheless require some knowledge of stochastic MPC for its correct implementation. Therefore, we show how to use the toolbox HILO-MPC (Pohlodek et al., 2022) to define and solve this stochastic MPC problem with minimal effort. This facilitates the applicability of advanced control methods for researchers and users without deep knowledge of stochastic control approaches. To showcase the toolbox, we apply the method in the case of a bioreactor producing a foreign protein by the recombinant *Saccharomyces cerevisiae* SEY2102, and we show some preliminary results.

1.1 Notation

The notation $\mathcal{N}(\mu, K)$ indicates a Gaussian distributed random variable with mean μ and covariance matrix K, with $\mathcal{GP}(\mu(x), K)$ we indicate a Gaussian process with mean function $\mu(x)$ and covariance matrix K. We indicate the i, j element of a matrix A with $[A]_{i,j}$. The $\ominus$ indicates the Pontryagin difference i.e. $A \ominus B = \{a | a + b \in A, \ \forall b \in B\}$, with $p(x)$ we indicate the probability density of x and with $p(x|a)$ the probability density conditioned to a. $\Pr(e)$ is the probability of an event e. $\|x\|_A^2 = x^\intercal A x$. The symbol $\dagger$ indicates the Moore-Penrose inverse of a matrix.

2. PROBLEM FORMULATION

We will consider discrete-time nonlinear hybrid models of the form

$$x_{k+1} = f(x_k, u_k) + B\left(g(x_k, u_k) + w_k\right), \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$ is the state and input, respectively. The function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is known while $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ describes the unknown effect on the system dynamics and will be learned from data using Gaussian processes. This additive hybrid model structure is commonly used and aims at modifying the known function to represent better the data. The vector $w \in \mathbb{R}^{n_d}$ is process noise, assumed to be Gaussian distributed with mean zero i.e. $w_k \sim \mathcal{N}(0, \Sigma^w)$ with diagonal variance matrix $\Sigma^w = \text{diag}([\sigma_1^2, ..., \sigma_{n_d}^2])$. The matrix $B \in \mathbb{R}^{n_x \times n_d}$ is known.

2.1 Gaussian process regressor

We shortly review the main concepts behind a Gaussian process (GP), for more details refer to Williams and Rasmussen (2006). Gaussian processes are stochastic data-based models that can be used for regression and classification. Compared to other machine learning models, such as neural networks, they have the advantage of naturally providing a measure of uncertainty on the prediction. Furthermore, they embed previous knowledge by choosing an appropriate kernel functions that, for example, ensure smoothness or periodicity of the solution that can be dictated by first-principles. Here we are interested in GP regressors, i.e. that infer a continuous function from available data. Let $\mathfrak{g} : \mathbb{R}^{n_\mathfrak{x}} \to \mathbb{R}$ be a function of an input vector $\mathfrak{x}$. The measured output is $\mathfrak{y} = \mathfrak{g}(\mathfrak{x}) + \mathfrak{v}$, where $\mathfrak{v}$ is assumed to be Gaussian distributed noise $\mathfrak{v} \sim \mathcal{N}(0, \sigma)$. Gaussian processes assume that the unknown function $\mathfrak{g}$ is Gaussian distributed, i.e.

$$\mathfrak{g}(\mathfrak{x}) \sim \mathcal{N}(m(\mathfrak{x}), k(\mathfrak{x}, \mathfrak{x})), \tag{2}$$

where $m : \mathbb{R}^{n_\mathfrak{x}} \to \mathbb{R}$ is the mean function and $m : \mathbb{R}^{n_\mathfrak{x}} \times \mathbb{R}^{n_\mathfrak{x}} \to \mathbb{R}$ is the covariance function or *kernel*. The training takes place by fitting the *hyperparameters* that are contained in the mean and kernel function. Let ϕ be the vector of hyperparameters and

$$D = \{\mathfrak{X} = [\mathfrak{x}_1, ..., \mathfrak{x}_{n_D}] \in \mathbb{R}^{n_D \times n_x}, \mathfrak{Y} = [\mathfrak{y}_1, ..., \mathfrak{y}_{n_D}] \in \mathbb{R}^{n_D \times 1}\},$$

the data set where n_D is the number of measurements. Then the training consists of maximizing the log-marginal likelihood $\phi^* = \arg\max_\phi \ln p(\mathfrak{Y}|\mathfrak{X}, \phi)$, where

$$\begin{aligned} \ln p(\mathfrak{Y}|\mathfrak{X}, \phi) = -\frac{1}{2}\Big(& \ln(|K + \sigma^2 I|) \\ & + (\mathfrak{m} - \mathfrak{Y}))^\intercal \ln(K + \sigma^2 I)^{-1}(\mathfrak{m} - \mathfrak{Y})) \\ & + \frac{n_D}{2}\ln(2\pi)\Big), \end{aligned}$$

where n_D is the data set size and K is the variance matrix where $[K]_{i,j} = k(\mathfrak{x}_i, \mathfrak{x}_j)$ and $\mathfrak{m} = [m(\mathfrak{x}_1), ..., m(\mathfrak{x}_{n_D})]$. The prediction is then calculated at a query point $\mathfrak{x}^*$ by building the joint distribution as

$$\begin{pmatrix} \mathfrak{y} \\ \mathfrak{g}^* \end{pmatrix} = \mathcal{GP}\left(\begin{pmatrix} \mathfrak{m} \\ m(\mathfrak{x}) \end{pmatrix}, \begin{pmatrix} K + \sigma^2 I & \mathfrak{k} \\ \mathfrak{k}^\intercal & k(\mathfrak{x}^*, \mathfrak{x}^*) \end{pmatrix}\right),$$

where $\mathfrak{k} = [k(\mathfrak{x}_1, \mathfrak{x}^*), ..., k(\mathfrak{x}_{n_D}, \mathfrak{x}^*)]^\mathsf{T}$. The predicted mean and variance is finally given by the conditional posterior distribution $p(\mathfrak{g}^*|\mathfrak{X}, \mathfrak{Y}) = \mathcal{N}(\mu(\mathfrak{x}^*), \Sigma(\mathfrak{x}^*))$, where

$$\mu(\mathfrak{x}^*) = m(\mathfrak{x}^*) + \mathfrak{k}^\mathsf{T}(K + \sigma^2 I)^{-1}(\mathfrak{Y} - m(\mathfrak{X})), \quad (3)$$

$$\Sigma(\mathfrak{x}^*) = k(\mathfrak{x}^*, \mathfrak{x}^*) - \mathfrak{k}^\mathsf{T}(K + \sigma^2 I)^{-1}\mathfrak{k}. \quad (4)$$

In our case, the measurements $\mathfrak{Y}$ are given for all $j \in [1, ..., n_D - 1]$

$$y_j = g(x_i, u_i) + w_j = B^\dagger(x_{j+1} - f(x_j, u_j)).$$

Note that in (2) $\mathfrak{g}$ is assumed to be scalar valued since multiple outputs are nontrivial to handle in the GP framework. Hence, commonly a GP regressor is trained for every output dimension, hence obtaining n_d regressors. Note that $\mathfrak{x} = [x, u]$. The resulting multivariate GP will be

$$d(x, u) \sim \mathcal{N}(\mu^d(x, u), \Sigma^d(x, u)), \quad (5)$$

where mean and variance are build by concatenating the mean and variances for all n_d GP, i.e. $\mu^d = [\mu_1, ..., \mu_{n_d}]$ and $\Sigma^d = \text{diag}\left([\Sigma_1^d, ..., \Sigma_1^d]\right)$.

2.2 Stochastic Model Predictive Control

With the learned GP regressor $d : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_d}$ we can now formulate the stochastic MPC as follows

$$\max_{\Pi(x)} \quad \mathbb{E}\left(\sum_{k=0}^{N-1} l(x_k, u_k) + e(x_N)\right), \quad (6a)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k) + B\left(d(x_k, u_k) + w\right), \quad (6b)$$

$$u_k = \pi(x_k), \quad x_0 = x(k), \quad (6c)$$

$$\Pr\left(x_{k+1} \in \mathcal{X}\right) \geq p^x, \quad (6d)$$

$$\Pr\left(u_k \in \mathcal{U}\right) \geq p^u. \quad (6e)$$

Problem (6) is an infinite-dimensional problem and generally computationally intractable. In the following, we will approximate this problem in a tractable deterministic problem as in Hewing et al. (2020). Here only the main idea is given. For details, refer to the mentioned paper. First, we define the *ancillary control* to consider a feedback control action into the future that keeps the predicted uncertainty small

$$\pi_i(x_i) = \mu_i^u + K_i(x_i - \mu_i^x), \quad (7)$$

where K_i is a gain matrix, μ_i^x the mean of the state,μ_i^u the mean of the input for time i. The optimization variable are μ_i^u, $\forall i \in [0, N-1]$. Finding K_i is not trivial. A linear-quadratic regulator can be used, for example, by linearizing the system around some points of an (approximated) trajectory of the system. For small nonlinearities, a constant K can also be chosen.

2.3 Approximation of the Stochastic Problem

We approximate the state, input and disturbance as jointly Gaussian distributed

$$\begin{pmatrix} x_i \\ u_i \\ d_i + w_i \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} \mu_i^x \\ \mu_i^u \\ \mu_i^d \end{pmatrix}, \begin{pmatrix} \Sigma_i^x & \Sigma_i^{x,u} & \Sigma_i^{x,d} \\ (\Sigma_i^{x,u})^\mathsf{T} & \Sigma_i^u & \Sigma_i^{u,d} \\ (\Sigma_i^{x,d})^\mathsf{T} & (\Sigma_i^{u,d})^\mathsf{T} & \Sigma_i^d + \Sigma_i^w \end{pmatrix}\right),$$

where $\Sigma_i^u = K_i \Sigma K_i^\mathsf{T}$ and $\Sigma_i^{x,u} = \Sigma_i^x K_i^\mathsf{T}$. Next, the nonlinear known model of the system is linearized around the mean, which leads to the following update equations

$$\mu_{i+1}^x = f(\mu_i^x, \mu_i^u) + B\mu_i^d,$$

$$\Sigma_{i+1}^x = \left[\nabla f(\mu_i^x, \mu_i^u), B\right] \Sigma_i \left[\nabla f(\mu_i^x, \mu_i^u), B\right]^\mathsf{T}.$$

Now we are left with defining $\mu_i^d, \Sigma_i^d, \Sigma_i^{x,d}$ and $\Sigma_i^{u,d}$. There are different methods that can be used (see Hewing et al. (2020) and references therein), in our case the Taylor approximation is used. Hence we obtain

$$\mu_i^d = \mu^d(\mu_i^\mathfrak{x}),$$

$$\begin{pmatrix} \Sigma_i^{\mathfrak{x},d} \\ \Sigma_i^d \end{pmatrix} = \begin{pmatrix} \Sigma_i^\mathfrak{x}(\nabla \mu^d(\mu_i^\mathfrak{x}))^\mathsf{T} \\ \Sigma^d(\mu_i^\mathfrak{x}) + \nabla \mu^d(\mu_i^\mathfrak{x}) \Sigma_i^\mathfrak{x} (\nabla \mu^d(\mu_i^\mathfrak{x}))^\mathsf{T} \end{pmatrix},$$

where

$$\mu_i^\mathfrak{x} = \begin{pmatrix} \mu_i^x \\ \mu_i^u \end{pmatrix}, \quad \Sigma_i^\mathfrak{x} = \begin{pmatrix} \Sigma_i^x & \Sigma_i^{x,u} \\ (\Sigma_i^{x,u})^\mathsf{T} & \Sigma_i^u \end{pmatrix}.$$

Now that variance, covariance matrices and means are defined, we proceed with definition of the chance constraints.

Constraints Tightening The classic approach is reformulate the constraints in terms of the state and input mean μ^x and μ^u and then tighten the constraints based on the error between real state or input an its respective mean. Hence, the constraints become

$$\mu_i^x \in \bar{\mathcal{X}}, \quad \bar{\mathcal{X}} = \mathcal{X} \ominus \mathcal{R}^x,$$

$$\mu_i^u \in \bar{\mathcal{U}}, \quad \bar{\mathcal{U}} = \mathcal{U} \ominus \mathcal{R}^u,$$

where $\mathcal{R}^x$ and $\mathcal{R}^u$ are the probabilistic i-step reachable set with probability p, that bounds the state and input error, i.e., $\Pr(e_i^j \in \mathcal{R}^j | e_0 = 0) \geq p^j, \quad j \in \{x, u\}$. This ensure that if $\mu_i^x \in \bar{\mathcal{X}}$ then $\Pr(x \in \mathcal{X}) \geq p^x$ (same applies for the input). The calculation of the reachable set and the tightening of the the constraint can be computationally expensive in general. Nevertheless, for some important classes of constraints, and thanks to the approximation of joint Gaussian distribution of states and inputs, it can be done efficiently. In biotech in particular, constraints are often half-space constraints, i.e. $h^\mathsf{T} x \leq b$ (for example the controller has to respect a given maximum of minimum concentration of metabolites). In this case the constraints can be efficiently tightened online as follows $h^\mathsf{T} x \leq b - \phi^{-1}(p^x)\sqrt{h^\mathsf{T} \Sigma_i^x h}$ where ϕ^{-1} is the quantile function of a Gaussian distribution and p^x is the chosen probability. Note that the constraints are a function of the state variance matrix Σ^x.

Objective function We consider the following quadratic cost terms

$$l(x_i, u_i) = \|x_i - x_i^r\|_Q^2 + \|u_i - u_i^r\|_R^2, \quad (8)$$

in our case, we will use the expected values, which read

$$\begin{aligned} \mathbb{E}(l(x_i, u_i)) =& \|\mu_i^x - x_i^r\|_Q^2 + \text{tr}\left(Q\Sigma_i^x\right) + \\ & \|\mu_i^u - u_i^r\|_R^2 + \text{tr}\left(R\Sigma_i^u\right), \\ \mathbb{E}(e(x_N)) =& \|\mu_N^x - x_N^r\|_E^2 + \text{tr}\left(E\Sigma_N^x\right). \end{aligned}$$

Now we have all the ingredients to define an approximated deterministic version of the stochastic problem (6).

2.4 Approximated Stochastic MPC

The approximated MPC problem is given by

$$\max_{\mu^u} \quad \mathbb{E}\left(\sum_{k=0}^{N-1} l(x_k, u_k) + e(x_N)\right) \quad (9a)$$

$$\text{s.t.} \quad \mu_{k+1}^x = f(\mu_k^x, \mu_k^u) + B\mu_k^d, \quad (9b)$$

$$\Sigma_{k+1}^x = \left[\nabla f(\mu_i^x, \mu_i^u), B\right] \Sigma_i \left[\nabla f(\mu_i^x, \mu_i^u), B\right]^\mathsf{T}, \quad (9c)$$

$$\mu_{k+1}^x \in \bar{\mathcal{X}}(\Sigma_{k+1}^x), \ \ \mu_k^u \in \bar{\mathcal{U}}(\Sigma_k^x), \quad (9d)$$

$$\mu_0^x = x(k), \ \ \Sigma_0^x = \Sigma^x(k), \quad (9e)$$

where $\mu^u = [\mu_0^u, ..., \mu_{N-1}^u]$ is the sequence of control inputs. As usual in MPC, the first element μ_0^u is applied to the

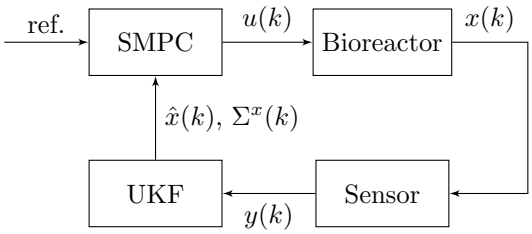

Fig. 1. Block diagram of the control strategy. SMPC: stochastic MPC.

plant and Problem (9) is solved again. In bioreactors, usually not all the states can be measured and need to be estimated. Furthermore, the estimate is uncertain and this uncertainty should be taken into account. For this the current state $x(k)$ and current variance $\Sigma^x(k)$ is estimated using an unscented Kalman filter (UKF) (see (Wan and Van Der Merwe, 2000) for the details). The model used by the UKF is the following

$$x_{k+1} = f(x_k, u_k) + \mu^x(x_k, u_k) + w_k, \tag{10}$$
$$y_k = h(x_k) + v_k, \tag{11}$$

where y_k is the measurement, $h(x_k)$ the measurement function and $\mu^x(x_k, u_k)$ is the mean of the GP. Zero-mean Gaussian distributed measurement and process noise are assumed. Note that in the model used in the UKF, only the mean of the GP is used.

3. CASE STUDY: PROTEIN PRODUCTION

As a case study, we use the model of a continuous reactor for the production of foreign protein using genetically modified *Saccharomyces cerevisiae SEY2102* (Tholudur and Ramirez, 1996) adapted for a continuous process. The model is given by

$$\dot{X} = \mu X - DX, \tag{12a}$$
$$\dot{S} = -7.5\mu X - D(S - S_f) \tag{12b}$$
$$\dot{P}_\mathrm{t} = R_\mathrm{fp} X - DP_\mathrm{t}, \tag{12c}$$
$$\dot{P}_\mathrm{m} = \phi(P_\mathrm{t} - P_\mathrm{m}) - DP_\mathrm{m}, \tag{12d}$$

where X is the biomass in g/l, S the substrate in g/l and P_t is the total protein amount in unit culture volume basis and P_m the target protein. The reactions rates used for the simulated plant are: $\mu = \frac{21.87s}{(s+0.4)(s+62.5)}$, $R_\mathrm{fp} = \frac{se^{-5s}}{s+0.1}$, $\phi = \frac{4.75\mu}{0.12+\mu}$. To simulate model mismatch we add structural errors in R_fp and ϕ as follows

$$R_\mathrm{fp} = \frac{se^{-5s}}{s^2 + 0.1}, \quad \phi = \frac{4.75\mu}{(0.12+\mu)(0.08+\mu)} \tag{13}$$

It is assumed that only X, S and P_t are measured online while P_m is not measured. The goal is reaching a reference known steady-state conditions of $x_\mathrm{ref} = [2.73, 0.08, 13.62, 13.12]$ and $u_\mathrm{ref} = 0.06$. A feedback gain matrix K (cf. (7)) was found by using an LQR controller with the linearized model around the steady-state conditions and kept constant for all the prediction horizon. The concentration of P_m cannot be greater than 14 hence an upper bound of 14 is added to the optimization problem for this variable. Gaussian distributed random noise with a variance of 0.01 is added to the measured states X, S and P_t. Here, without loss of generality, we train the GP before the process begins, hence the training is offline. The data points are generated by running five batches with a MPC that uses the wrong model and tries to reach the previous mentioned references. The GPs are trained with 47 data points.

Implementation using HILO-MPC: HILO-MPC can be used to easily set up and solve problem (9). Here, we show an example of code for our case study. Note that for brevity the plant model is not reported and the values of the variables are omitted. For the complete code refer to the HILO-MPC website [1] .

```
from hilo_mpc import SMPC, GPArray, Kernel, UKF,
↪  SimpleControlLoop
from hilo_mpc.library.models import ecoli_plant_model,
↪  ecoli_model

# Get the approximated model
model = ecoli_model()
model.discretize('rk4', inplace=True)

# Get the plant model
plant = ecoli_plant_model()

# Load the data
data = scipy.io.loadmat('data.mat')
Y_train = data['Y_train']
X_train = data['X_train']

# Train the GPs. Here, 'GPArray' describes the
# concatenated output of the separately trained
# single output GPs.
gps = GPArray(model.n_x)
for k, gp in enumerate(gps):
    kernel = Kernel.squared_exponential(variance=...,
    ↪  length_scales=...)
    gp.initialize(feature_names, label_names[k],
    ↪  kernel=kernel)

    y_train = Y_train[k, :]

    gp.set_training_data(X_train, y_train)
    gp.setup()
    gp.fit_model()

"Set up the UKF"
# Summing a GP to the model will automatically
# add the mean of the GP to the model.
model_ukf = model + B @ gps
ukf = UKF(model_ukf)
ukf.setup()  # Set up the UKF
ukf.R = ...
ukf.Q = ...
ukf.set_initial_guess(x0=..., P0=...)

"Set up the Stochastic MPC"
mpc = SMPC(model, gps, B)
mpc.horizon = ...

mpc.quad_stage_cost.add_states(names=..., weights=...,
↪  ref=...)
mpc.quad_terminal_cost.add_states(names=...,
↪  weights=..., ref=...)
mpc.set_box_constraints(x_lb=..., x_ub=..., u_lb=...,
↪  u_ub=...)
mpc.setup(options={'stoc_approx': 'joint_gaussian'})

"Run the simulation for 500 time steps"
scl = SimpleControlLoop(plant, mpc, ukf)
scl.run(500)  # Simulates the system
scl.plot()  # Shows the results
```

[1] www.ccps.tu-darmstadt.de/research_ccps/hilo_mpc

Results: Figure 2 and 3 show the results of the simulations. Figure 2 shows the real noise-free concentrations, the predicted means of the states and 2σ standard deviation on the predicted means. Note that, thanks to the feedback gain, the uncertainty does not increase with the prediction and the constraint is satisfied at list with probability of 95% (i.e. a standard deviation of 2σ). Figure 3 compares the results of the stochastic MPC and of the nominal MPC, i.e., using the wrong reactions rates in (13) without the GP correction. In the case of the nominal MPC the reference cannot be reached.

4. CONCLUSIONS AND FUTURE RESEARCH

In this paper we used a hybrid modeling approach that uses Gaussian processes in a stochastic MPC approach. This allowed the use of data to capture unknown system dynamics and furthermore consider model uncertainty. HILO-MPC was used to simulate a continuous bioreactor that produces foreign protein with a recombinant *S. cerevisiae* and some preliminary results are shown. Through its easy and minimal syntax the toolbox facilitates the use of machine learning supported optimal control problems. In future HILO-MPC releases, we plan to provide also other stochastic MPC approaches using, for example, Monte Carlo sampling or Polynomial Chaos Expansion to propagate the uncertainty. Furthermore, the study on the applicability of such methods for biotechnological processes will be further refined.

REFERENCES

Allgöwer, F., Findeisen, R., and Nagy, Z.K. (2004). Nonlinear model predictive control: From theory to application. *Journal of the Chinese Institute of Chemical Engineers*, 35(3), 299–315.

Bradford, E., Imsland, L., Reble, M., and del Rio-Chanona, E. (2021). Hybrid Gaussian process modeling applied to economic stochastic model predictive control of batch processes. *Lecture Notes in Control and Information Sciences*, 485, 191–218.

Bradford, E., Reble, M., Bouaswaig, A., and Imsland, L. (2019a). Economic stochastic nonlinear model predictive control of a semi-batch polymerization reaction. In *IFAC-PapersOnLine*, volume 52, 667–672.

Bradford, E., Reble, M., and Imsland, L. (2019b). Output feedback stochastic nonlinear model predictive control of a polymerization batch process. In *2019 18th European Control Conference, ECC 2019*, 3144–3151.

Duvigneau, S., Dürr, R., Kranert, L., Wilisch-Neumann, A., Carius, L., Findeisen, R., and Kienle, A. (2021). Hybrid cybernetic modeling of the microbial production of polyhydroxyalkanoates using two carbon sources. *Computer Aided Chemical Engineering*, 50, 1969–1974.

Findeisen, R. and Allgöwer, F. (2002). An Introduction to Nonlinear Model Predictive Control. 21st Benelux Meeting on Systems and Control, 119–141. CiteSeer.

Heirung, T.A.N., Paulson, J.A., O'Leary, J., and Mesbah, A. (2018). Stochastic model predictive control — how does it work? *Computers and Chemical Engineering*, 114, 158–170.

Hewing, L., Kabzan, J., and Zeilinger, M.N. (2020). Cautious Model Predictive Control Using Gaussian Process Regression. *IEEE Transactions on Control Systems Technology*, 28(6), 2736–2743.

Koller, M., Horvat, P., Hesse, P., Bona, R., Kutschera, C., Atlić, A., and Braunegg, G. (2006). Assessment of formal and low structured kinetic modeling of polyhydroxyalkanoate synthesis from complex substrates. *Bioprocess Biosyst Eng*, 29, 367–377.

Luo, Y., Kurian, V., and Ogunnaike, B.A. (2021). Bioprocess systems analysis, modeling, estimation, and control. *Current Opinion in Chemical Engineering*, 33, 100705.

Mears, L., Stocks, S.M., Albaek, M.O., Sin, G., and Gernaey, K.V. (2017). Mechanistic fermentation models for process design, monitoring, and control. *Trends in Biotechnology*, 35(10), 914–924.

Mesbah, A. (2018). Stochastic model predictive control with active uncertainty learning: A Survey on dual control. *Annual Reviews in Control*, 45, 107–117.

Mitra, S. and Murthy, G.S. (2021). Bioreactor control systems in the biopharmaceutical industry: a critical perspective. *Systems Microbiology and Biomanufacturing*, 1–22.

Mowbray, M., Petsagkourakis, P., Chanona, E.A.d.R., Smith, R., and Zhang, D. (2021). Safe chance constrained reinforcement learning for batch process control. *arXiv preprint arXiv:2104.11706*.

Oliveira, R. (2004). Combining first principles modelling and artificial neural networks: a general framework. *Comp. & Chem. Eng.*, 28(5), 755–766.

Pohlodek, J., Morabito, B., Schlauch, C., Zometa, P., and Findeisen, R. (2022). Flexible development and evaluation of machine-learning-supported optimal control and estimation methods via HILO-MPC. doi:10.48550/ARXIV.2203.13671.

Rathore, A.S., Mishra, S., Nikita, S., and Priyanka, P. (2021). Bioprocess control: current progress and future perspectives. *Life*, 11(6), 557.

Rawlings, J.B., Mayne, D.Q., and Diehl, M.M. (2017). *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, Santa Barbara, California.

Roman, M. and Olaru, S. (2018). Model-based design for biosystems. Control opportunities and discrete-time modelling challenges. *IFAC-PapersOnLine*, 51(30), 666–671.

Sokolov, M., von Stosch, M., Narayanan, H., Feidl, F., and Butté, A. (2021). Hybrid modeling — a key enabler towards realizing digital twins in biopharma? *Current Opinion in Chemical Engineering*, 34, 100715.

Tholudur, A. and Ramirez, W.F. (1996). Optimization of fed-batch bioreactors using neural network parameter function models. *Biotechnology Progress*, 12(3), 302–309.

Von Stosch, M., Oliveira, R., Peres, J., and de Azevedo, S.F. (2014). Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Comp. & Chem. Eng.*, 60, 86–101.

Wan, E.A. and Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, 153–158. Ieee.

Williams, C.K. and Rasmussen, C.E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Zhang, D., Del Rio-Chanona, E.A., Petsagkourakis, P., and Wagner, J. (2019). Hybrid physics-based and data-driven modeling for bioprocess online simulation and optimization. *Biotechnology and Bioengineering*, 116(11), 2919–2930.

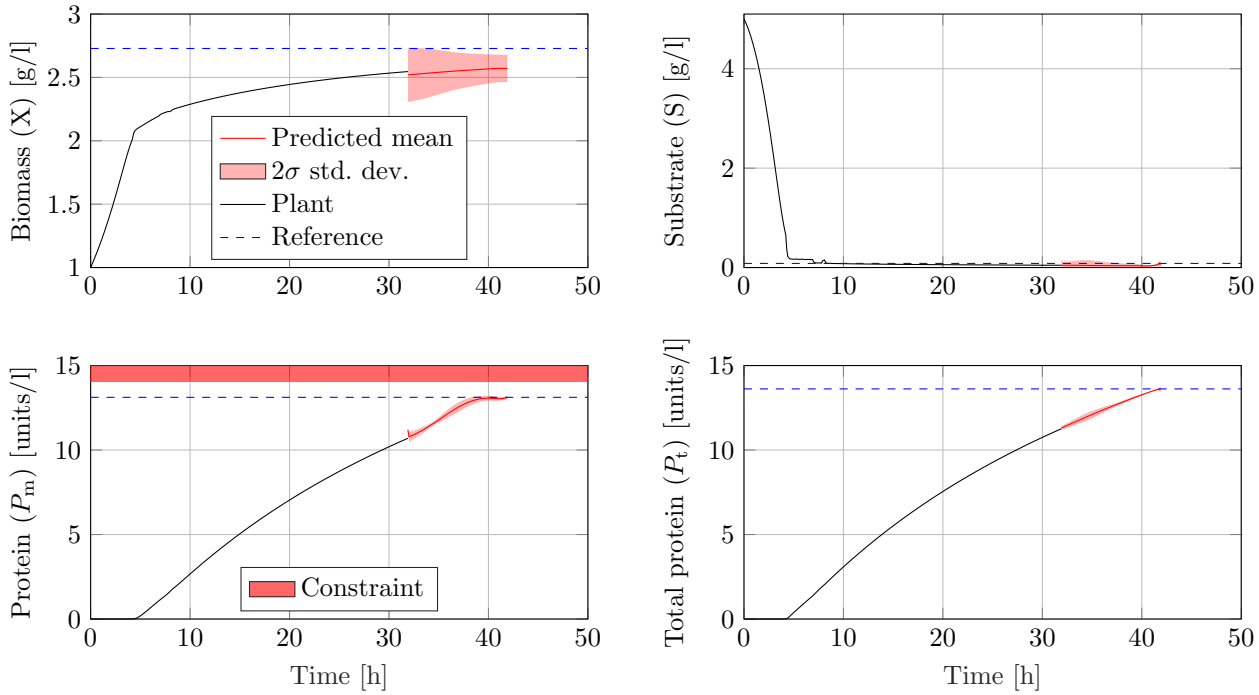

Fig. 2. Result of one iteration of the stochastic MPC. The black lines are the states of the real system and the red lines are the mean of each state predicted by the stochastic MPC. The red area shows the uncertainty of this prediction, where the bounds are defined as 2σ standard deviation from the mean.

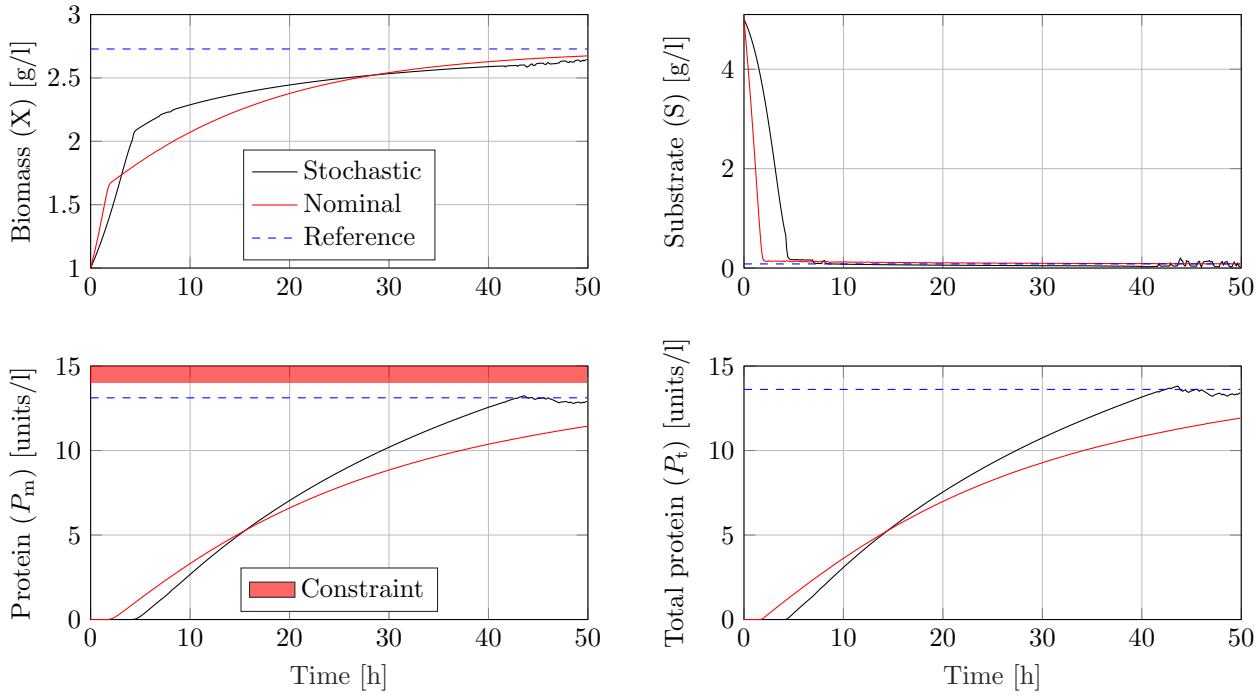

Fig. 3. Comparison between stochastic MPC and nominal MPC. The black lines are the results of the stochastic MPC, and the red lines of the nominal MPC.