Experimental validation of modifier adaptation and Gaussian processes for real time optimisation ⋆

Evren M. Turan *, **Sofie Lia** *, **José Matias** **, **Johannes Jäschke** *.

* *Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (e-mail: evren.m.turan@ntnu.no, johannes.jaschke@ntnu.no).*
** *Department of Chemical Engineering, McMaster University, Hamilton, Canada (e-mail: assumpcj@mcmaster.ca).*

Abstract: Plant-model mismatch is a major challenge in the implementation of real time optimisation approaches. Various methods have been proposed to correct for this mismatch based on online data, however there is still little work in the experimental validation and comparison of methods. This work focuses on the experimental implementation and validation of some recently proposed methods to account for plant-model mismatch in a real time optimisation context: output modifier adaption, Gaussian processes, and Gaussian process output modifier adaptation.
These methods are implemented on an experimental rig, designed to emulate gas-lifted oil wells drawing from the same reservoir. All the methods are, on average, able to improve upon the baseline performance, with Gaussian process output modifier adaptation showing the best performance. A major challenge was to tune the methods to be robust against the process noise.

Keywords: Real time optimisation, Modifier adaptation, Gaussian processes, Experimental validation

1. INTRODUCTION

Real time optimisation (RTO) seeks to enhance the performance of a process by finding operating points that maximise some economic objective while satisfying constraints, typically related to quality and safety. Most RTO methods use online measurements and a process model to solve an optimisation problem at each iteration to find these operating points. In the traditional approach (Chen and Joseph, 1987) a rigorous steady state process model is developed, and at each iteration the parameters of the model are updated based on steady state measurements to compute the new operating point. Many variations of this approach have been proposed, mainly differing in how the online measurements are used.

Despite its potential, RTO is not widely used in industry. Significant challenges that prevent widespread RTO adoption are 1) the need to identify and wait for steady state, and 2) plant-model mismatch. Plant-model mismatch refers to the existence of some discrepancy between the model and plant predictions. If certain model-adequacy conditions are met, then with appropriate adjustment of model parameters the optimum computed by the model will match the plant optimum. Unfortunately, these conditions are difficult to verify for realistic systems (Forbes et al., 1994). A group of methods for overcoming these difficulties rely on augmenting the model using online measurements, e.g. Integrated System Optimization and Parameter Estimation (ISOPE) (Roberts, 1995), and modifier adaptation (MA) (Marchetti et al., 2009). This work focuses on the experimental implementation and validation of recently proposed variants of modifier adaptation on an experimental lab rig.

Modifier adaptation aims to estimate a correction term from online data and include this in the optimisation problem to overcome plant-model mismatch (Marchetti et al., 2009). Under mild assumptions this approach allows convergence to a point satisfying the Karush-Kuhn-Tucker (KKT) conditions of the plant despite model mismatch. In the most variants of modifier adaptation, these corrections are linear terms that are either added to the objective and constraint functions (the "classic" approach) or to the model equations (output modifier adaptation) (Marchetti et al., 2009).

Unfortunately, modifier adaptation requires an accurate estimate of the process gradients which can be extremely challenging in practice due to process noise. Many variants of modifier adaptation have been proposed (Marchetti et al., 2016), most of which aim at improving the method's robustness to noise and/or efficiency (Marchetti et al., 2010; François and Bonvin, 2014; de Avila Ferreira et al., 2018; Marchetti et al., 2016).

A recently proposed approach is to use Gaussian processes to model the plant model mismatch, and either include the Gaussian processes in the RTO optimisation or to use them

⋆ This work was supported by the Norwegian Research Council through the AutoPRO project.

to calculate the modifiers used in a modifier adaptation approach (de Avila Ferreira et al., 2018; del Rio Chanona et al., 2019, 2021). Gaussian processes were proposed as they are a well-established, non-parametric modelling approach that inherently account for noise, while providing an estimate of uncertainty in their predictions.

Despite the extensive theoretical and *in-silico* results there is very little work on the experimental implementation of modifier adaptation and its variants. Here, we implement and compare the performance of three methods, 1) output modifier adaption, 2) Gaussian processes embedded in real time optimisation, and 3) Gaussian process output modifier adaptation, on an experimental lab rig emulating a gas-lifted oil well system. To the authors' knowledge, the Gaussian process approaches (de Avila Ferreira et al., 2018; del Rio Chanona et al., 2019, 2021) have not been previously implemented on real, experimental systems.

2. BACKGROUND

2.1 Real time optimisation (RTO)

In an RTO approach we would like to solve the following steady state optimisation problem at each iteration:

$$\min_{u\in\mathcal{U}} \ \phi(u, x_p(u)) \tag{1a}$$
$$s.t. \ g(u, x_p(u)) \leq 0 \tag{1b}$$

where ϕ is the objective function, usually related to process economics, $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ are the decision or input variables, and $x_p(u)$ are the steady state plant states corresponding to input u. Typically the plant mapping, $f_p : \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, is not known and instead some model, $f : \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, is used in the related optimisation problem:

$$\min_{u\in\mathcal{U}} \ \phi(u, f(u)) \tag{2a}$$
$$s.t. \ g(u, f(u)) \leq 0 \tag{2b}$$

In general the model will not be able to capture the plant behaviour exactly, e.g. due to modelling simplifications. Thus, a solution of problem (2) may not correspond to a solution of (1). Instead it may even be infeasible for the true process, with some violation of a safety critical constraint. Under some hard to verify model adequacy conditions the model parameters can be updated to allow for convergence to the plant optimum (Chen and Joseph, 1987; Forbes et al., 1994). This is not true in general, because it requires the optimality conditions of the model to coincide with that of the plant.

2.2 Modifier adaptation

In a modifier adaptation strategy zeroth (ϵ) and first order (λ) correction terms are introduced to the optimisation problem such that the optimisation problem's necessary conditions of optimality match the plant optimum (Marchetti et al., 2009). Note that in contrast to traditional RTO (Chen and Joseph, 1987) the model parameters are not updated at each iteration. In the standard modifier adaptation approach these corrections are applied directly to the cost functions and constraints (Marchetti et al., 2009),

$$\begin{aligned}\phi_{MA}(u, f(u)) &= \phi(u, f(u)) + \epsilon_k^{\phi} + \lambda_k^{\phi\prime}(u - u_k)\\ g_{MA}(u, f(u)) &= g(u, f(u)) + \epsilon_k^{g} + \lambda_k^{g\prime}(u - u_k)\end{aligned} \tag{MA}$$

but, they can also be applied to the outputs of the model (Marchetti et al., 2009),

$$f_{MA}(u) = f(u) + \epsilon_k + \lambda_k'(u - u_k) \tag{MAy}$$

where the subscript k refers to the kth iteration of the modifier adaptation algorithm. The latter approach is termed output modifier adaptation (MAy) and will be considered in this work. In MAy the modifiers represent a linear correction to the process model, with the modifiers calculated from the plant states by:

$$x_p(u_k) = f(u_k) + \epsilon_k \tag{3a}$$
$$\lambda_k = \frac{\mathrm{d}x_p(u_k)}{\mathrm{d}u} - \frac{\mathrm{d}f(u_k)}{\mathrm{d}u} \tag{3b}$$

Note that $x_p(u)$ are the *noise-free plant states*, not measurements.

A commonly recommended approach is to filter the modifiers and/or the input variables (Marchetti et al., 2016):

$$\epsilon_k = (I - K_\epsilon)\epsilon_{k-1} + K_\epsilon \left(x_p(u_k) - f(u_k)\right) \tag{4a}$$
$$\lambda_k = (I - K_\lambda)\lambda_{k-1} + K_\lambda \left(\frac{\mathrm{d}x_p(u_k)}{\mathrm{d}u} - \frac{\mathrm{d}f(u_k)}{\mathrm{d}u}\right) \tag{4b}$$
$$u_k = (I - K_u)u_{k-1} + K_u u_k^* \tag{4c}$$

where u_k^* are the inputs found by solving the optimisation problem, u_k are the actual inputs used, and $K_\bullet$ are scalar filter values or filter matrices, typically defined as $K_\bullet = \mathrm{diag}(k_1, \ldots, k_{n_\bullet})$, $k_i \in (0, 1]$. By filtering one indirectly incorporates prior data into the algorithm which allows for greater robustness to noisy measurements, as each iteration cannot change the modifiers/set point as significantly. Another potential approach is to use a trust-region strategy instead of a filtering strategy (del Rio Chanona et al., 2019, 2021).

2.3 Obtaining plant derivatives

A key challenge in the successful implementation of modifier adaptation schemes is the need to estimate $\frac{\mathrm{d}x_p(u)}{\mathrm{d}u}$, as typically only noisy measurements of x_p are available. Let the plant measurement equation be:

$$y_p(u) = x_p(u) + \eta(u) \tag{5}$$

where η models the noise (stochasticity) component of the system. A possible approach to approximate $\frac{\mathrm{d}x_p(u)}{\mathrm{d}u}$ is to use a finite difference approximation, e.g. the first order forward finite difference, assuming $n_u = n_x = 1$ is:

$$\frac{\mathrm{d}x_p(u)}{\mathrm{d}u} \approx \frac{y_p(u + h) - y_p(u)}{h}, \qquad h > 0 \tag{6}$$

where h is a step size in the input. This approach has two main disadvantages: 1) one requires at least n_x perturbations of the plant around the operating point, which can be extremely costly and time-consuming, 2) this approach can be extremely sensitive to noise.

This sensitivity to noise can be shown by decomposing the error of the finite difference approximation (6):

$$\left|\frac{\mathrm{d}x_p(u)}{\mathrm{d}u} - \frac{y_p(u+h)-y_p(u)}{h}\right| \leq \left|\frac{\mathrm{d}x_p(u)}{\mathrm{d}u} - \frac{x_p(u+h)-x_p(u)}{h}\right| + \left|\frac{\eta(u+h)-\eta(u)}{h}\right| \quad (7)$$

The first term is the truncation error which decreases with h , while the second term is due to solely to the noise component, and has a variance which *increases* with decreasing h. Note also that for most reasonable choices of h, the variance of the noise term will be larger than the variance of $\eta(u)$. Furthermore, when $\frac{\mathrm{d}x_p(u)}{\mathrm{d}u}$ is small relative to the noise (e.g. near an unconstrained optima), the noise term will dominate the error. Thus, noise is a serious challenge in estimating $\frac{\mathrm{d}x_p(u)}{\mathrm{d}u}$.

An important note is that in the approach above, measurements from previous iterations are not incorporated in the calculations of the modifiers. Instead, previous measurements are only used indirectly in the filter equations. Thus, any large instance of noise can significantly impact the modifiers at an iteration, leading to calculating an unsuitable set point, unless aggressive filter values are chosen. Various authors have proposed adaptions that make use of previous measurements, including dual MA methods (Marchetti et al., 2010), fitting a model to the mismatch data (Gao et al., 2016; de Avila Ferreira et al., 2018), and the use of stochastic optimisation approaches (Rivas, 2021).

2.4 Gaussian processes

Gaussian processes were proposed to be used to create a model of the plant-model mismatch, as typically show good performance in low-data regimes, and by design account for the influence of noise and uncertainty (de Avila Ferreira et al., 2018). A brief description of the pertinent points of Gaussian processes is below, for a detailed introduction see Rasmussen and Williams (2005) or Gramacy (2020).

Gaussian processes are a non-parametric supervised learning approach for multiple-input single-output functions, widely used in machine learning and surrogate optimisation (Rasmussen and Williams, 2005; Gramacy, 2020). A Gaussian process is defined by a mean function, $m(u)$ (often a zero mean function) and a covariance or kernel function, $k(u, u')$, which describe a joint Gaussian distribution of a finite set of random variables (Rasmussen and Williams, 2005).

The kernel function identifies the closeness of points, and is a user choice when designing the Gaussian process. The characteristics and performance of Gaussian processes are heavily dependent on the choice of kernel functions (Rasmussen and Williams, 2005). A common choice of kernel function, for infinitely differentiable approximations, is the squared exponential kernel:

$$k(u_i, u_j) = \sigma_k \exp\left\{-\frac{(u_i - u_j)^2}{l^2}\right\} \quad (8)$$

where l is the length scale, and σ_k is the signal variance. Note that if the inputs are far from each other the function will be small (low correlation), while if they are close to each other the function will be larger (highly correlated). So, if u_i were a known data point, then u_j will be influenced by it if it is "close enough". This degree of closeness is strongly determined by l and σ_k which are typically found by maximum likelihood or maximum a posteriori estimation.

Consider some real valued function $f(u) : \mathbb{R}^n \to \mathbb{R}$, and the task of predicting the output at a vector of points, U_*, given noisy observations, Y_d, of the function at a vector of points, U_d. Additionally, assume that these noisy observations are due to additive independent identically distributed (i.i.d.) Gaussian noise with variance σ_n. Then by definition, the joint distribution of the Gaussian process describing this is:

$$\begin{bmatrix} Y_d \\ Y_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(U_d) \\ m(U_*) \end{bmatrix}, \begin{bmatrix} K_{dd} + \sigma_n I & K_{d*} \\ K_{*d} & K_{**} \end{bmatrix}\right) \quad (9)$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2, K_{ij} is a matrix with elements $k(U_i, U_j)$, and I is the identity. Note that σ_n, the noise variance, enters the joint distribution as an additive term to the variance of the training points. The distribution of the predicted outputs, $p(Y_*)$, conditioned on the data (i.e. the predictive posterior distribution) is given by:

$$p(Y_*|U_d, Y_d, U_*) = \mathcal{N}(\mu, \Sigma) \quad (10a)$$

$$\mu(U_*) = m(U_*) + K_{d*}[K_{dd} + \sigma_n I]^{-1}(Y_d - m(U_d)) \quad (10b)$$

$$\Sigma(U_*) = K_{**} - K_{*d}[K_{dd} + \sigma_n I]^{-1} K_{d*} \quad (10c)$$

where $\mu(U_*)$ represents the expected value at the points (U_*), and Σ describing the associated uncertainty for the predicted μ's. To gain some insight to equations eq: GP-predictive-equation, note that if there is no noise $\sigma_n = 0$, and the prediction points match the observations points, $U_d = U_*$, then $\mu = Y_d$ with $\Sigma = 0$, i.e. the output prediction will be the training data. Having $\sigma_n \neq 0$ means that the GP will instead return a "filtered" version of the input, as it assigns some of the input-output variation as due to noise.

2.5 Gaussian processes and real time optimisation

Gaussian processes can be used in a real time optimisation scheme, where $\mu(u)$ is used to describe the plant-model mismatch, either by 1) correcting the cost functions and constraints,

$$\phi_{GP}(u) = \phi(u) + \mu(u), \quad \text{(GP-RTO-1)}$$

2) correcting the model,

$$f_{GP,i}(u) = f_i(u) + \mu_i(u), \quad \text{(GP-RTO-2)}$$

or 3) to estimate the first order modifiers

$$f_{GP,i}(u) = f_i(u) + \mu_i(u_k) + \frac{\mathrm{d}\mu_i(u_k)}{\mathrm{d}u}(u - u_k). \quad \text{(GP-MAy)}$$

As GPs have scalar outputs, in GP-RTO-2 and GP-MAy one needs to define a separate GP for every plant output, which is shown by the i subscript in the equations. Note that GP-MAy, has a linear correction term, while the other formulations have a non-linear correction. The linear correction only attempts to correct the model locally, while the non-linear correction attempts to correct the model over the whole input space. This also means that the corrected model may fail to meet model adequacy conditions, in which case it may not yield converging iterates. Furthermore, this linear correction uses $GP_i(u_k)$ to correct the model bias instead of the measured bias

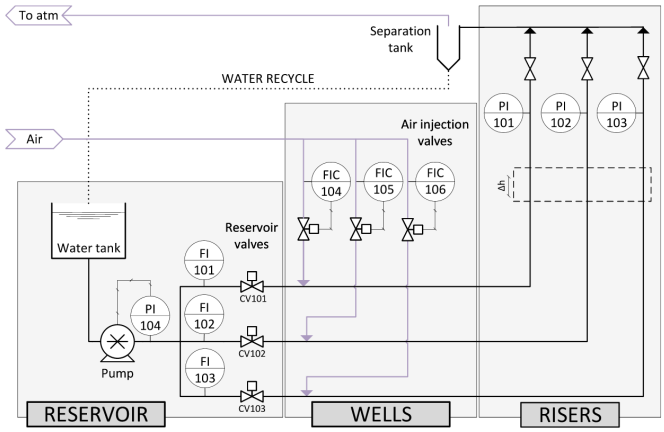

Fig. 1. Schematic of the gas lift lab rig, reproduced from Matias et al. (2022).

$y_p(u_k) - f(u_k)$, as the GP gives an estimate of the true plant state $x_p(u_k)$.

Gaussian processes were first introduced in the context of RTO under model mismatch in the GP-RTO-1 form (de Avila Ferreira et al., 2018). Later authors considered constructing trust regions from the GP variance estimate (del Rio Chanona et al., 2019), the use of acquisition functions in the optimisation (del Rio Chanona et al., 2021), and distributed optimisation variant of this scheme (Andersson et al., 2020).

3. CASE STUDY

In this work we examine the performance of MAy, GP-MAy, GP-RTO-2 on a gas lift lab rig, originally introduced and described in detail in Matias et al. (2022). A brief description of the system follows, however we refer to this work for further details.

3.1 Experimental rig

A schematic of the experimental set up is shown in Figure 1. The rig is designed to emulate the production of oil from three wells, with different production characteristics, that draw from the same reservoir. In the streams from each "well" gas can be injected to increase the production rate. The change in production rate is because increasing the amount of gas, decreasing the bulk fluid density, and hence the hydrostatic pressure due to the vertical riser. The pressurised reservoir is represented by the tank and pump system, which is controlled to have constant outlet pressure. To emulate the different production characteristics of the wells the valves CV101-CV103 are given different openings. For example, the decreasing of a well's productivity with time due to well depletion could be represented by decreasing the valve opening. Finally, the liquid flows after the riser are recycled back to the reservoir tank to form a closed loop. In the experimental set up we use water and air as the liquid and gas respectively, instead of the oil and natural gas used with an offshore system. Although these fluids are different, we are still able to see that gas-lift effect, and hence are able to study this phenomenon, and its role in resource allocation for production optimisation.

Table 1. Valve openings during experiment.

	Scenario		
Description	1	2	3
Time Span [s]	0-1200	1200-1800	1800-2400
CV-101	0.8	0.3	0.3
CV-102	0.4	0.4	0.4
CV-103	0.6	0.6	0.25

We consider real time optimisation of the experimental rig with the gas injection in the wells as the input variables, u (sensors FIC104-FIC106 in Figure 1, L/min at standard conditions for temperature and pressure), the liquid flow rates as the states, Q (sensors FIC101-FIC103, L/min), and the valve openings as measured disturbances, v_w (CV101-103). The objective is to maximise a weighted sum of liquid flow rates subject to total gas handling constraint, and bounds on the gas injection.:

$$\max_u \; 20Q_1(u_1, v_1) + 10Q_2(u_2, v_2) + 30Q_3(u_3, v_3) \quad (11a)$$

$$\sum_{w=1}^{3} u_w \leq 7.5\,\text{L/min} \quad (11b)$$

$$1\,\text{L/min} \leq u_w \leq 5\,\text{L/min} \qquad w = 1,2,3 \quad (11c)$$

$$0 \leq v_w \leq 1 \qquad w = 1,2,3 \quad (11d)$$

The weightings in the objective function reflect the operational situations where the wells produce different hydrocarbons, such that they require different processing and contribute differently to the economics. Likewise, the constraint on total gas lift is a common industrial constraint based on operational limitations.

A mechanistic model of the experimental rig has been developed (Matias et al., 2022), however to consider operation with significant plant-model mismatch we use a simple quadratic model to describe the flow rate from each well:

$$\hat{Q}_w = \theta_{w,1} + \theta_{w,2}u_w + \theta_{w,3}v_w + \theta_{w,4}u_w^2 + \theta_{w,5}v_w^2 \quad (12)$$

where w indexes the well, $\hat{Q}_w$ is the *predicted* liquid flow rate, and θ_w is a vector of model parameters with values shown in Table 2. Note that in the notation from section 2, $\hat{Q}_w$ is the model prediction of the x variable.

As in a real production system the gas lift is used as a manipulated variable. However, this cannot be manipulated directly. Instead, flow controllers are used to adjust valves to maintain the gas lift at the calculated set point (labelled air injection valves in Figure 1). This means that in addition to noise in the outputs, there is also noise in the process inputs. Instead of including noise in the inputs in the GP model, we instead use a time-averaged process input as the measured input variable, noting also that the noise in the inputs is significantly less than that of the flow rate measurements.

To compare the experimental performance of the MAy, GP-MAy, GP-RTO-2 methods we also consider using the quadratic model without any adaptations as a baseline method. During the experimental runs we change the choke valve openings to give three operating scenarios, as shown in Table 1. This is to model changes in the reservoir characteristics that may occur during operation. Note that this is treated as a measured disturbance for all the methods.

3.2 Implementation details

The approaches are implemented with the use of LabVIEW, MATLAB (R2020b), the GPML toolbox (v4.2) (Rasmussen and Nickisch, 2016), CasADi v3.5.5 (Andersson et al., 2019), and IPOPT (Wächter and Biegler, 2006).

As all the methods require the plant to reach steady state, we perform each change in the inputs u at intervals of 90 seconds. This is significantly longer than the settling time of the system, and ensures that we are only providing steady state data when calculating the mismatch. In addition to reduce the influence of noise we provide the methods the average measurement of the gas lift and corresponding flow rates over the last 45 seconds. This was done as it was found that not taking the average severely impacts the performance of all the methods.

To find suitable parameters for the methods, we performed several tests and chose the best combination of parameters for each method. For the filter parameters in the MAy approach, equations 4a-c, we use scalar filters $K_u = 0.8$, $K_\lambda = 0.3$, and $K_\epsilon = 0.7$. The aggressive filter values for the first order modifiers is due to the significant impact of noise on the finite difference gradient approximation, for which we use $h = 0.2$. In GP-RTO-2 method we only filter the inputs ($K_u = 0.6$) as we rely on the GP to reject the plant noise. With GP-MAy, we do not filter u or the modifiers, and instead restrict the maximum change in the inputs at each iteration to be 0.75.

For both of the GP approaches we consider corrections to the quadratic model using zero-mean Gaussian processes with squared exponential covariance functions. The covariance functions have separate length scales for u_w and v_w. The hyperparameters to estimate are the noise variance $\sigma_{n,w}$, and the length scales $l_{u,w}$, $l_{v,w}$. We do not optimise for $l_{v,w}$ and instead set these to $l_{v,w} = 0.01$. The reason for doing so is that each valve has at most two different valve openings during the run (see Table 1) which means that these length scales cannot be meaningfully optimised due to the lack of data. Instead in they are set to a relatively small value, which ensures that the GP will treat any change in the valve opening as a significant change to the system as the kernel functions will return very small values (see equation 8 and the discussion below).

Furthermore, to improve the robustness of the hyperparameter optimisation we perform maximum a posteriori estimation, with a Gaussian prior used for $l_{u,w}$, with mean and variance of 0.1. This is equivalent to regularising the optimisation by introducing a penalty proportional to $(l_{u,w} - 0.1)^2$. Without these considerations the hyperparameters are too sensitive to the data when the number of measurements are small, leading to repeated runs of the experiments giving very different results.

Lastly we consider the initialisation of the methods. All the approaches begin with uniform gas lift assignment and the constraint on total gas lift active, i.e. $u_1 = u_2 = u_3 = 2.5$. This is not an optimum allocation for plant or quadratic model. For the GP approaches there is first an initial probing period where data is gathered with gas lift allocations of 1.5 and 4.5 (see Figure 3), before starting the algorithms with uniform gas lift assignment.

Table 2. Parameters for quadratic model

Well	$\theta_{w,1}$	$\theta_{w,2}$	$\theta_{w,3}$	$\theta_{w,4}$	$\theta_{w,5}$
1	0.246	0.150	9.656	-0.008	-7.169
2	-0.411	0.689	30.153	-0.056	-21.956
3	-1.104	0.775	26.641	-0.080	-19.643

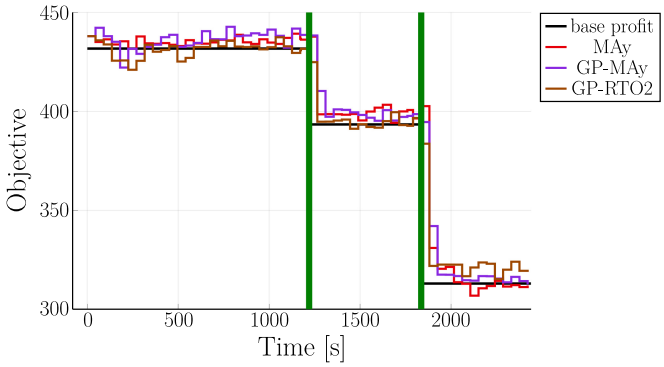

Fig. 2. Average objective value (45 seconds) of the methods compared the average profit achieved by using the quadratic model without modifications (base profit). Different scenarios are separated by vertical green lines.

4. EXPERIMENTAL RESULTS

The online performance of the different methods are shown in Figure 2, with average objective values in Table 3, and corresponding manipulated inputs in Figure 3. The average objective includes the initialisation of three data points for the GP methods, and a section without starting MAy (to allow for synchronisation of the step changes in the valves). We note that although the improvement in average objective values is small (Table 3), this is of similar magnitude to the expected gain for industrial implementations ($\sim 1 - 4\%$) (Foss et al., 2018).

Except for GP-RTO-2 in the scenario 1, the methods improve over the baseline performance for all the scenarios, as shown in Table 3, despite finding different inputs (Figure 3). Despite the difference in input values, all three methods prioritise gas lift to well 2 the least in sections 1 and 2. This is the expected behaviour as well 2 has the lowest weight in the cost function, and a low productivity due to the small valve opening in first two scenarios (Table 1). In section 3 MAy performs the worst, and prioritises Well 1 over the others. In contrast both GP methods perform better, although GP-RTO-2 has significant oscillations in the allocation of well 2 and 3. However, despite the oscillations GP-RTO-2 actually performs the best in Section 3 (clearly shown in Figure 2). Upon examining the changes in the fitted GPs from iteration to iteration, these oscillations appear to be due to the high level of noise hampering the identification of the true system states.

The best performing method on average across the three sections is GP-MAy. Furthermore, note that from a operations point of view the input profile of this method is good, with less perturbations than MAy, and seeming to settle to fixed values in each section, unlike GP-RTO-2. Based on these two points, the results suggest GP-MAy as the most promising method.

Table 3. Average objective value in each section and the across all sections.

	Section			Avg.
Method	1	2	3	
Base	432	393	313	379
MAy	435	402	320	386
GP-MAy	436	402	322	387
GP-RTO-2	431	397	325	384

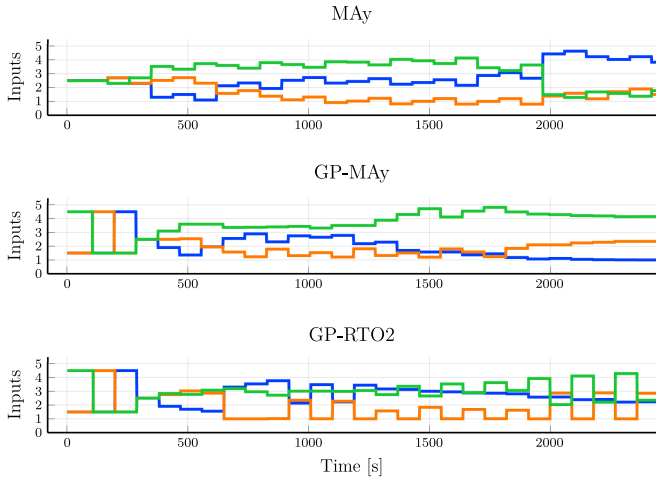

Fig. 3. Inputs used by the different algorithms. In all subplots the blue, red and green lines are for wells 1, 2, and 3 respectively. GP-MAy and GP-RTO2 include an initial probing period, see section 3.2.

5. CONCLUSIONS

We demonstrate the use of output modifier adaption and two methods of incorporating GPs in RTO on an experimental set up. All methods are able to improve upon the base case on average, despite significant system noise. GP-MAy gave the most consistent performance across the experiments (Table 3). For all methods, averages of the measurements had to be provided for reasonable performance, and filters or constraints had to be introduced to reduce the impact of noise. For reasonable performance all the methods required tuning of their parameters, including setting priors for the GP parameters, based on the system characteristics. Further work could include comparisons with other methods for RTO under model-mismatch, the use of stochastic optimisation methods, and the introductions of constraints on the system outputs.

REFERENCES

Andersson, J.A., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36.

Andersson, L.E., Bradford, E.C., and Imsland, L. (2020). Gaussian processes modifier adaptation with uncertain inputs for distributed learning and optimization of wind farms. *IFAC-PapersOnLine*, 53(2), 12626–12631.

Chen, C.Y. and Joseph, B. (1987). On-line optimization using a two-phase approach: An application study. *Industrial & engineering chemistry research*, 26(9), 1924–1930.

de Avila Ferreira, T., Shukla, H.A., Faulwasser, T., Jones, C.N., and Bonvin, D. (2018). Real-time optimization of uncertain process systems via modifier adaptation and gaussian processes. In *2018 European Control Conference (ECC)*, 465–470. IEEE.

del Rio Chanona, E.A., Graciano, J.A., Bradford, E., and Chachuat, B. (2019). Modifier-adaptation schemes employing gaussian processes and trust regions for real-time optimization. *IFAC-PapersOnLine*, 52(1), 52–57.

del Rio Chanona, E.A., Petsagkourakis, P., Bradford, E., Graciano, J.A., and Chachuat, B. (2021). Real-time optimization meets bayesian optimization and derivative-free optimization: A tale of modifier adaptation. *Computers & Chemical Engineering*, 147, 107249.

Forbes, J., Marlin, T., and MacGregor, J. (1994). Model adequacy requirements for optimizing plant operations. *Computers & chemical engineering*, 18(6), 497–510.

Foss, B., Knudsen, B.R., and Grimstad, B. (2018). Petroleum production optimization–a static or dynamic problem? *Computers & Chemical Engineering*, 114, 245–253.

François, G. and Bonvin, D. (2014). Use of transient measurements for the optimization of steady-state performance via modifier adaptation. *Industrial & Engineering Chemistry Research*, 53(13), 5148–5159.

Gao, W., Wenzel, S., and Engell, S. (2016). A reliable modifier-adaptation strategy for real-time optimization. *Computers & chemical engineering*, 91, 318–328.

Gramacy, R.B. (2020). *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. Chapman and Hall/CRC.

Marchetti, A., Chachuat, B., and Bonvin, D. (2010). A dual modifier-adaptation approach for real-time optimization. *Journal of Process Control*, 20(9), 1027–1037. doi:https://doi.org/10.1016/j.jprocont.2010.06.006. ADCHEM 2009 Special Issue.

Marchetti, A., Chachuat, B., and Bonvin, D. (2009). Modifier-adaptation methodology for real-time optimization. *Industrial & engineering chemistry research*, 48(13), 6022–6033.

Marchetti, A.G., François, G., Faulwasser, T., and Bonvin, D. (2016). Modifier adaptation for real-time optimization—methods and applications. *Processes*, 4(4), 55.

Matias, J., Oliveira, J.P., Le Roux, G.A., and Jäschke, J. (2022). Steady-state real-time optimization using transient measurements on an experimental rig. *Journal of Process Control*, 115, 181–196.

Rasmussen, C.E. and Nickisch, H. (2016). The gpml toolbox version 4.0. *Technical Documentation*.

Rasmussen, C.E. and Williams, C.K.I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press. doi: 10.7551/mitpress/3206.001.0001.

Rivas, R. (2021). *Contributions to real-time optimization of process systems under uncertainty*. Thesis, Technische Universität Dortmund.

Roberts, P. (1995). Coping with model-reality differences in industrial process optimisation—a review of integrated system optimisation and parameter estimation (ISOPE). *Computers in Industry*, 26(3), 281–290.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25–57.