



# Article A Multiple Solution Approach to Real-Time Optimization

Jack Speakman <sup>1,\*</sup> and Grégory François <sup>1,2</sup>

- <sup>1</sup> School of Engineering, The University of Edinburgh, Edinburgh EH9 3FB, UK
- <sup>2</sup> Institute of Systems Engineering, School of Engineering, HES-SO Valais-Wallis Rue de L'Industrie 23, 1950 Sion, Switzerland
- \* Correspondence: jack.speakman@ed.ac.uk

Abstract: Modifier Adaptation (MA) is a method of real-time optimization (RTO) which modifies a single model to match the first order properties of the plant. Known uncertainties in the parameters of this model are discarded in favor of real-time measurements, but they can be used to quantify the mismatch between the plant and model. Using multi-model methods increases the computation time, but can improve rate of convergence of the RTO scheme. This article proposes a framework, known as multiple solution modifier adaptation (MSMA), which produces several models which are all modified in the same way as standard MA, each producing a potential solution to be applied to the plant. From this framework, three recommended schemes are proposed on how to select the operating point to be applied to the plant: (1) Selecting the solution based off the modifiers; (2) Selecting the mean solution from convex models; (3) Selecting the closest solution to the current operating point. Each of these methods have different advantages, including limiting the increase in computational complexity and improving the model adequacy conditions of the scheme. These recommended schemes are shown on three different case studies of varying complexity with all three schemes showing improvements over standard MA.

**Keywords:** multi-model; multiple solution; modifier adaptation; real-time optimization; plant-model mismatch; MSMA

# 1. Introduction

Through advances in computing power and mathematical techniques, advanced optimization techniques have grown to be a vital part of process systems engineering. Optimization of a process allows for an appreciable increase in the efficiency, reduction in waste, guarantee of safety, improved product quality, and maximized profits. A well optimized process provides a competitive edge over the field, benefits the whole supply chain down to the consumer, and allows for a sustainable operation in a world of increasing resource scarcity.

The standard approach to finding the optimum of a process is to create a mathematical model, and apply optimization techniques to this model to determine the operating conditions which minimize (or maximize) an objective function. This operating point must also satisfy a set of conditions known as constraints, and an infeasible operating point violates the process constraints. A model cannot fully accurately predict the outcome of a real process, as no model is perfect. The presence of simplifications, noise and uncertainty of the process results in an unavoidable mismatch between the eventual realization of the process (known as the plant) and model.

If the degree of uncertainty of the model can be determined, it can be accounted for in the optimization, however the resulting operating point can never be guaranteed to be at the true optimum of the plant, and may still result in infeasible operation. In order for improvements to be made, additional information about the plant is required to allow for an improvement to the operation of the process, and potentially the ability to find the true optimum of the plant. This information comes in the form of measurements from sensors



Citation: Speakman, J.; François, G. A Multiple Solution Approach to Real-Time Optimization. *Processes* 2022, 10, 2207. https://doi.org/ 10.3390/pr10112207

Academic Editor: Jean-Pierre Corriou

Received: 30 September 2022 Accepted: 24 October 2022 Published: 26 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). in the plant, which allow for the states to be estimated. Utilizing these measurements in process optimization is the basis of the field known as real-time optimization (RTO). This information alone is sufficient to apply standard optimization techniques directly to the plant (e.g., gradient-based algorithms, Hooke-Jeeves algorithm, etc.) without the need of a model, by estimating the true value of the objective function and the constraint functions and directly modifying the operating conditions towards optimality. Using this unstructured approach often results in poor performance as the experiment cost (i.e., the cost to get an accurate measurement) and violating the constraints of the process during an experiment are not of great enough importance in standard computational optimization techniques.

The need for an approach which can find the optimum of a process with minimal risk in the experiments, and reduced total number of experiments is desired. One such approach is to use the measurements directly and avoid the need for an optimization scheme in a feedback style control scheme [1–3]. Another approach for which this can be achieved is by marrying the measurements taken directly from the plant with the uncertain model, and is a step in the right direction to formulating an approach which can rapidly converge towards the true plant optimum without violating the constraints of the plant. The most popular approach to RTO using a model structure is the two-step approach [4–6], which uses the measurements to estimate the value of the uncertain parameters of the model. Using these new parameters, a new operating point can be found which has an increased chance of approaching the true plant optimum and remaining feasible than using standard optimization techniques. This approach however has a detrimental flaw in that convergence is achieved when the measurements from the plant predict a model which has the current operating point as the optimum [7]. This convergence criteria does not always coincide with the optimal conditions of the plant.

One method of ensuring that an RTO scheme will only converge at an operating condition which satisfies the Karush–Kuhn–Tucker (KKT) conditions of the plant is to ensure that at the current operating point, the first order conditions of optimality of the model are equal to the measured values of the plant [8–10]. This is the basis behind a group of RTO schemes known as modifier adaptation (MA) [11]. If the model is modified such that the first order properties of the model equal the measured values of the plant, upon convergence the operating point must be the optimum of the modified model, which by definition is also the optimum of the plant. The simplest approach to ensuring first-order matching is through the addition of affine modifiers directly to the objective and constraint functions of the nominal model.

The guarantee of KKT matching upon convergence is a strong result, but comes with equally strong drawbacks. The primary drawback is that in order calculate the modifiers which allow the model to match the first-order properties of the plant, the value and the gradient of the plant objective and constraints are required at each iteration of the RTO scheme. Estimating the gradient of the plant numerically requires both additional experiments at operating conditions local to the current operating point, and very accurate estimates of the states as errors in the measurements are amplified in the gradient calculation. Therefore, many advances to the MA framework have come with the intention to reduce the impact of these issues, and offer methods which reduce the number of experiments required for the gradient estimation [12]. Such methods include using previous measurements for the gradient estimation [13,14], nested MA [15], directional MA [16–18], and using quadratic approximations [19,20].

Another direction of advancing MA comes in the form of using additional information from the measurements to allow for a more informed decision to be made in determining where improvements to the operating point can be made. Some examples of using additional information comes in the form of transient measurements between experiments [21–23], which can provide better state estimates and reduce the time required for each experiment to be performed, or using information of uncontrollable disturbances [24,25], but are measured and therefore can be accounted for. Another set of data which is not fully utilized is the set of previous measurement from the plant, which are typically discarded in favor of the most recent measurements. For this, the use of Gaussian processes can interpolate between the data points and with or without a model, this newly created fit can be used to predict the outcomes of the plant [26,27].

A typical approach to process modeling is during the conceptualization and design stages of a plant, a model is produced using measurements from a pilot plant of the process. The parameters of a mathematical model are adjusted to fit these measurements, but will not be able to fully accurately fit this data (without over fitting) due to the presence of noise and errors in the foundations of the model (from simplifications). The uncertainty in these parameters provide valuable insight into the mismatch between the plant and model (assuming the pilot plant is an accurate representation of the true plant). An operating point which is robust to this uncertainty can be found using multi-model-based optimization techniques such as worst-case analysis. Both the uncertainty in the parameters and the real-time measurements are both used to estimate the plant-model mismatch, therefore either one or the other is typically used, rather than both. Another disadvantage behind using a multi-model approach in an RTO setting is that solving the RTO solution is required to be fast and solving multi-model is typically more computationally intensive (therefore slower) compared to single-model approaches. Despite this, recently the authors to this article, and others have shown that using mutliple models to generate a single non-linear program (NLP) to be solved for the RTO approach can improve the rate and stability of convergence, and reduce the likelihood of breaking the constraints of the plant before convergence [28–30]. These methods are set-up to use the uncertainty in the model parameters to produce a single NLP to be solved, whereas the methods proposed in this article solve each model NLP independently to produce a set of candidate solution, which are then combined into a single operating point to be applied to the plant.

This article is structured as follows; firstly, the problem to be solved is mathematically defined, along with the standard MA framework. Next, in Section 3, the proposed multiple solution modifier adaptation framework is defined and the two primary research questions are posed. The following two sections offer answers to these research equations, firstly looking at how the multiple solutions can be generated in Section 4, then how they can be processed into a single RTO solution in Section 5. Finally, some of the most promising approaches in the framework are compared against one another on the benchmark case study of the Williams-Otto CSTR, a distillation column, and a semi-batch reactor.

#### 2. Problem Definition

#### 2.1. Model-Based Optimization

The objective of process optimization is to find a set of inputs, given by  $u \in \mathbb{R}^{n_u}$ , which minimize an objective function, given by  $\Phi \in \mathbb{R}$ , subject to some constraints, given by  $G \in \mathbb{R}^{n_g}$ . For the plant, this can be cast as the solution to the following non-linear program (NLP):

$$u_p^* \in \arg\min_{\boldsymbol{u}} \quad \Phi_p(\boldsymbol{u}),$$
  
s.t. 
$$G_p(\boldsymbol{u}) \leq \boldsymbol{0},$$
 (1)

where a subscript *p* refers to a property of the plant. Typically, a process with have a set of outputs, given by  $y \in \mathbb{R}^{n_y}$ , which are functions of the inputs. The objective and constraint functions are typically a function of both the inputs and outputs, such that,

$$\Phi(u) = \phi(u, y(u)), \tag{2}$$

$$G(u) = g(u, y(u)).$$
(3)

It is assumed that the objective and constraint functions are known functions w.r.t. the input and outputs. However, the exact mapping between the outputs and the inputs for the plant is considered to be unknown, therefore the plant NLP cannot be directly solved. The standard approach to overcoming this unknown is to produce a model of the process which predicts the outputs of the plant. One such type of model is known as a parameterized state-space model which is formed of two sets of equations,

$$\frac{\partial x}{\partial t} = F(x, u, \theta), \tag{4}$$

$$y = H(x, u, \theta), \tag{5}$$

where *x* represents the states, *t* represents time, and  $\theta$  represents the model parameters. For RTO, the primary interest is in the steady-state outputs, therefore the time derivative can be set to **0** and rearranged for the states (i.e.,  $x = f(u, \theta)$ ; if this cannot be mathematically rearranged, it can be solved numerically in the optimization of the NLP). Using this function for the states, a functional representation of the outputs can be achieved,

$$\mathbf{y}(\mathbf{u}, \boldsymbol{\theta}) = \mathbf{H}(f(\mathbf{u}, \boldsymbol{\theta}), \mathbf{u}, \boldsymbol{\theta}). \tag{6}$$

The model-based NLP can be formulated using this functional representation of the outputs,

$$u^{*} \in \arg\min_{u} \quad \phi(u, y(u, \theta)),$$
  
s.t. 
$$g(u, y(u, \theta)) \leq 0,$$
  
$$u \in \mathcal{U}.$$
 (7)

This NLP is solved using the nominal parameters,  $\theta = \theta_{nom}$ . The solution to the model-based NLP is only guaranteed to be at true plant optimum if there is no plant-model mismatch, i.e.,  $y(u, \theta_{nom}) = y_p(u)$ . However, in practice there is always some degree of plant-model mismatch present, therefore the model-based NLP will always produce a suboptimal solution (barring chance) and potentially even operate at an infeasible operating condition for the plant.

Using higher fidelity models can assist in ensuring the model-based optimum is closer to the true optimum, but this is a costly endeavor and cannot fully account for the discrepancy, especially if the plant is being operated in a different set-up to what was originally defined during the modeling stage. Another approach is to use the known uncertainty in the model to account for the mismatch, however this typically leads to a feasible, but highly suboptimal operating point [31–33].

## 2.2. Modifier Adaptation

The approach to rectifying this discrepancy between the plant and model of interest to this article is real-time optimization (RTO), specifically modifier adaptation (MA). Using measurements of the plant, a true insight into the state of the plant at the current operating point can be observed. MA uses this window into the plant to determine the first-order properties of the plant, and using these properties, a linear term is added to the model characteristic functions to force the modified models to match the first order conditions of optimality observed of the plant. The modified NLP used by the standard MA scheme is as follows,

$$u_{k}^{*} \in \arg\min_{u} \quad \Phi_{k}(u),$$
  
s.t. 
$$G_{k}(u) \leq 0,$$
  
$$u \in \mathcal{U},$$
 (8)

where *k* refers to the iteration number, and  $\Phi_k$  and  $G_k$  are the modified cost and constraints, respectively, which are defined as,

$$\Phi_k(\boldsymbol{u}) = \phi(\boldsymbol{u}, \boldsymbol{y}(\boldsymbol{u}, \boldsymbol{\theta}_{nom})) + \varepsilon_k^{\phi} + (\boldsymbol{u} - \boldsymbol{u}_k)^{\mathsf{T}} \boldsymbol{\lambda}_k^{\phi}, \qquad (9)$$

$$G_k(\boldsymbol{u}) = \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{y}(\boldsymbol{u}, \boldsymbol{\theta}_{nom})) + \boldsymbol{\varepsilon}_k^g + (\boldsymbol{u} - \boldsymbol{u}_k)^\mathsf{T} \boldsymbol{\lambda}_{k'}^g$$
(10)

where  $\varepsilon$  and  $\lambda$  represent the zeroth and first order modifiers, respectively.  $\varepsilon_k^{\varphi}$  can be neglected from the formulation of the modified objective function as it does not directly influence the NLP solution. The other modifiers are defined as the difference between the observed values of the plant, and the estimated values of the model at the current operating point,

$$\lambda_{k}^{\phi} = \nabla_{u} \Phi_{p}(\boldsymbol{u}) \Big|_{\boldsymbol{u}_{k}} - \nabla_{u} \Phi(\boldsymbol{u}, \boldsymbol{\theta}_{nom}) \Big|_{\boldsymbol{u}_{k}'}$$
(11)

$$\lambda_k^g = \nabla_u G_p(u) \Big|_{u_k} - \nabla_u G(u, \theta_{nom}) \Big|_{u_k'}$$
(12)

$$\boldsymbol{\varepsilon}_{k}^{g} = \boldsymbol{G}_{p}(\boldsymbol{u}_{k}) - \boldsymbol{G}(\boldsymbol{u}_{k}, \boldsymbol{\theta}_{nom}). \tag{13}$$

where  $\nabla_u$  is the gradient operator. The main advantage of MA lies in its mathematically proven ability to guarantee KKT matching upon convergence [11]. Said differently, the linear correction terms of the model expressions of the cost and constraints functions of Equations (9) and (10) are sufficient to reconcile the 1st order conditions of optimality of the model and of the plant upon convergence. It follows that Problem (1) for the plant and Problem (8) for the modified model share the same solutions upon convergence, despite the fact that the model and plant are different. An adequate model for MA is one which satisfies the second order conditions of optimality at the plant optimum, as if this is not met then the plant optimum cannot be the solution to Problem (8) when the current operating point is at the plant optimum, and convergence cannot occur. As the modifiers do not affect the second order properties, the model must meet these, which are that the model has a positive definite Hessian of the Lagrangian at the plant optimum (i.e., a minimum can be observed).

# 2.3. Model Uncertainty in MA

During the modeling stage of a process, the parameters of the model are defined from observations of experiments. These parameter calculations are condensed into a nominal value—the set of parameters which have the highest confidence to accurately predict the plant outcome—and an uncertainty in the set of nominal parameters. Standard MA techniques neglect the uncertainty in these nominal parameters in favor of using the measurements of the plant to account for the uncertainty. However, there is merit to be taken from this known uncertainty in the parameters before experiments are taken from the plant.

One such approach which takes advantage of the known uncertainty in the model parameters in MA is an approach which utilizes the Lipschitz constants of the process [34]. Lipschitz constants are defined as the maximum gradient of the characteristic functions, which can be used to define a region of guaranteed feasibility around the current operating point. These constants for the plant cannot be directly measured, however an upper bound can be used from the model to estimate this region. By using the uncertainty in the model parameters a more conservative estimate can be obtained for these constants [30].

Another approach is to ensure that each iteration satisfies the modified constraints for not only the nominal parameters, but for each set of parameters in the uncertainty set [29]. Alternatively, a single model can be generated which upper-bounds the second derivative of all the models in the uncertainty set [28]. All these approaches to using the known model uncertainty aim to use this uncertainty to create a single NLP to be solved by the RTO scheme.

This article is interested in maintaining the individual NLPs for each set of parameters which could accurately predict the plant, and using the set of solutions to create a fully informed decision on the next set of operating conditions.

# 3. Multiple Solution MA

The general principle behind multiple solution modifier adaptation (MSMA) is to generate a set of potential operating points, and use this set operating points to find the 'best' operating point for the next iteration of the RTO scheme. Naturally from this, two research questions arise:

- 1. Solution Production: How can the set of potential solutions be generated?
- 2. Solution Processing: How can the set of potential solutions be combined into a single operating point to be applied to the plant?

The use of multiple models to produce multiple solutions can be summarized by the following NLP,

$$u_{k,\pi}^* \in \arg\min_{u} \quad \Phi_{k,\pi}(u),$$
  
s.t. 
$$G_{k,\pi}(u) \leq 0,$$
  
$$u \in \mathcal{U},$$
 (14)

$$\boldsymbol{u}_{k+1} = f_{MS}(\boldsymbol{u}_k, \boldsymbol{u}_{k,1}^*, \boldsymbol{u}_{k,2}^*, \dots, \boldsymbol{u}_{k,\pi}^*),$$
(15)

where  $\pi$  refers to the model number. Problem (14) is the NLP used to calculate the individual solutions which make up the set of potential solutions, and Equation (15) refers to the solution processing, which uses the current operating point and the set of potential operating points to find the next RTO iterate. The functions used in the NLP used to calculate the potential operating points follow the same principle as standard MA, by introducing modifiers which rectify the current value and gradient such that they match the estimated values of the plant at the current operating point. This is achieved in the same manner, through modifiers which are added to the model optimization functions.

$$\Phi_{k,\pi}(\boldsymbol{u}) = \Phi_{\pi}(\boldsymbol{u}) + (\boldsymbol{u} - \boldsymbol{u}_k)^{\mathsf{T}} \boldsymbol{\lambda}^{\phi}_{k,\pi} + \boldsymbol{\varepsilon}^{\phi}_{k,\pi'}$$
(16)

$$G_{k,\pi}(\boldsymbol{u}) = G_{\pi}(\boldsymbol{u}) + (\boldsymbol{u} - \boldsymbol{u}_k)^{\mathsf{T}} \boldsymbol{\lambda}_{k,\pi}^g + \boldsymbol{\varepsilon}_{k,\pi'}^g$$
(17)

where  $\Phi_{\pi}$  and  $G_{\pi}$  are the choice of model used to calculate the solutions. The modifiers are dependent on the model used,

$$\boldsymbol{\varepsilon}_{k,\pi}^{\varphi} = \boldsymbol{\Phi}_{p}(\boldsymbol{u}_{k}) - \boldsymbol{\Phi}_{\pi}(\boldsymbol{u}_{k}), \tag{18}$$

$$\boldsymbol{\varepsilon}_{k,\pi}^{g} = \boldsymbol{G}_{p}(\boldsymbol{u}_{k}) - \boldsymbol{G}_{\pi}(\boldsymbol{u}_{k}), \tag{19}$$

$$\lambda_{k,\pi}^{\phi} = \nabla_{u} \Phi_{p}(\boldsymbol{u}) \Big|_{\boldsymbol{u}_{k}} - \nabla_{u} \Phi_{\pi}(\boldsymbol{u}) \Big|_{\boldsymbol{u}_{k}'}$$
(20)

$$\lambda_{k,\pi}^g = \nabla_u G_p(\boldsymbol{u}) \Big|_{\boldsymbol{u}_k} - \nabla_u G_\pi(\boldsymbol{u}) \Big|_{\boldsymbol{u}_k}.$$
(21)

Importantly, the information used to calculate these modifiers requires the same estimates used by standard MA from the plant, therefore additional experiments are not required. The main advantage of modifier adaptation is the guarantee of KKT matching upon convergence, therefore it is desirable that the solution processing method maintains this property. As is seen in Section 5, depending on the method used, the resulting RTO solution may not be equal to any of the individual proposed solutions generated by the models. This can result in premature convergence which does not satisfy the KKT conditions of the plant. An example of this may be that two models suggest operating points in opposite directions, and the resulting processed solution may suggest the current operating point, even though the estimated first order properties of the plant suggest that

the current operating point does not satisfy the KKT conditions of the plant. A sufficient condition to avoid this is as follows,

**Condition 1.** *Convergence, defined as,* 

$$\lim_{k \to \infty} u_{k+1} = u_k, \tag{22}$$

only occurs if the current operating point is equal to one or more of the potential solutions generated by the models.

With the above condition defined, the following can be proposed on the KKT matching of the plant.

**Theorem 1** (Plant KKT matching). If using an MSMA scheme which has converged to a point,  $u_{\infty} = \lim_{k \to \infty} u_k$ , Condition 1 is met, and perfect estimates of the gradient and value of the plant are known at the current operating point. Then this converged point is a KKT point of the plant.

**Proof of Theorem 1.** As stated in Theorem 1 and Condition 1, the converged operating point is also a converged point of one of the corresponding models. Thus, as each model matches the gradient of the plant at the current operating point, this must also be a KKT point of the plant, as with standard MA.

With the overarching mechanisms of the proposed MSMA approach clearly outlined, with a sufficient condition for plant KKT matching upon convergence. In the following sections, the two research questions posed earlier are fleshed out and several methods are investigated.

# 4. Producing the Proposed Solutions

As shown in the previous section, the MSMA approach requires the functions  $\Phi_{\pi}(u)$  and  $G_{\pi}(u)$  to be defined. This section will investigate three different approaches to generating these functions.

#### 4.1. Using Known Model Uncertainties

The first approach considers generates the NLPs using the known uncertainty in the model. This uncertainty can either be structural, where several potential structures for the model are known and each can be used to formulate a separate NLP; or through parametric uncertainty, where the model parameters are known to have uncertainties in their precise values due to simplifications, noise or other sources; or a combination of structural and parametric uncertainties. The NLP functions can be formulated as follows,

$$\Phi_{\pi}(\boldsymbol{u}) = \Phi(\boldsymbol{u}, \boldsymbol{\theta}_{\pi}), \tag{23}$$

$$G_{\pi}(u) = G(u, \theta_{\pi}), \quad \theta_{\pi} \in \vartheta,$$
 (24)

where  $\vartheta$  is a set of parameters which contains the structural and parametric uncertainties in the model. As each of these models are corrected such that the value and gradient matches the estimated value of the plant (and therefore match each other) at the current operating point, this approach assumes that by changing the parameters of the model that second or higher order changes to the model are seen. If changes in the chosen parameters only result in linear changes to the model, the resulting modified models will be identical, and the advantages of a multi-solution approach would be lost.

Additionally, using this approach results in many NLPs of similar complexity to the nominal model being solved for each RTO iteration. Depending on the fidelity of these models, this can be a resource heavy task which cannot always be appropriate in practice, especially if there are many sets of parameters. Therefore, there is a natural tradeoff between the number of models and the benefit gained, with each additional model providing diminishing returns. As the modifier adaptation method uses global modifiers to rectify the local properties, the usefulness of using a high-fidelity model is up for debate, and lower fidelity models which have parameters which can shed light on the plant-model mismatch may be more appropriate for this method.

#### 4.2. Using Convexified Models

The second approach to generating the set of solutions considered looks into using convexified models. Convex models offer benefits not only within RTO but also general optimization, where they are generally easier to solve, reducing the computational intensity for each iteration. There are several approaches which have been applied to RTO regarding the use of convex models. Here, a set of convex NLPs is formulated using the models generated from the previous approach.

This approach to convexification is to use each of the models formulated from the known uncertainties, and fit a quadratic function w.r.t. the inputs to the objective and constraint functions [35]. This results in a set of quadratic models used in the NLP,

$$\Phi_{\pi}(\boldsymbol{u}) = \boldsymbol{u}^{\mathsf{T}} \boldsymbol{Q}_{\pi}^{\Phi} \boldsymbol{u} + \boldsymbol{L}_{\pi}^{\Phi} \boldsymbol{u} + \boldsymbol{C}_{\pi}^{\Phi},$$
<sup>(25)</sup>

$$G_{\pi,j}(\boldsymbol{u}) = \boldsymbol{u}^{\mathsf{T}} \boldsymbol{Q}_{\pi}^{\mathsf{G}_{j}} \boldsymbol{u} + \boldsymbol{L}_{\pi}^{\mathsf{G}_{j}} \boldsymbol{u} + \boldsymbol{C}_{\pi}^{\mathsf{G}_{j}}, \quad \forall j \in [1, \dots, n_{g}],$$
(26)

where *Q*, *L*, and *C*, are the quadratic, linear and constant terms for each function, respectively. These terms can be found through a least squares regression across a set of inputs points for each set of parameters in the parameter set, i.e.

$$\left[\mathbf{Q}_{\pi}^{\Phi}, \mathbf{L}_{\pi}^{\Phi}, \mathbf{C}_{\pi}^{\Phi}\right] = \min_{\mathbf{Q}, \mathbf{L}, \mathbf{C}} \sum_{\mathbf{u} \in \mathcal{U}} \left| \left| P_{2}(\mathbf{u}) - \Phi(\mathbf{u}, \boldsymbol{\theta}_{\pi}) \right| \right|^{2},$$
(27)

$$\left[\mathbf{Q}_{\pi}^{G_{j}}, \mathbf{L}_{\pi}^{G_{j}}, \mathbf{C}_{\pi}^{G_{j}}\right] = \min_{\mathbf{Q}, \mathbf{L}, \mathbf{C}} \sum_{\boldsymbol{u} \in \mathcal{U}} \left\| P_{2}(\boldsymbol{u}) - G_{j}(\boldsymbol{u}, \boldsymbol{\theta}_{\pi}) \right\|^{2},$$
(28)

where  $P_2(u)$  is a second order polynomial w.r.t. the inputs,

$$P_2(\boldsymbol{u}) = \boldsymbol{u}^{\mathsf{T}} \boldsymbol{Q} \boldsymbol{u} + \boldsymbol{L} \boldsymbol{u} + \boldsymbol{C}. \tag{29}$$

As convexity is desired, a condition is placed on the eigenvalues of Q such that they are positive. Using quadratic approximations vastly simplifies the multiple NLPs to be solved at each iteration, significantly reducing the computational cost of generating the set of potential operating points. This comes are the cost of losing artifacts from the original model which might explain important phenomena of the process. As the modifiers which are added by the MA scheme already correct the local properties and ruin the global predictive power of the modified model, replacing the detailed model with a quadratic approximation will not have catastrophic implications.

#### 4.3. Using Location Based Models

The previous methods have relied on a set of known uncertainties being readily available, and that this set provides valuable insight into the plant-model mismatch. This is not always the case. This last method offers an alternative approach to generating a set of NLPs to solve, formulated using a single model of the plant.

Using a quadratic fit on the model was utilized in the previous approach, however this fit was carried out over the whole input space, producing a function which provides a general overview of the global properties of the model, but may remove useful information on the local properties of the plant. An alternative is to split the input space into several regions and fit a separate quadratic function to each region. These regions can either be from a uniform grid, or generated through an analysis of the model to formulate regions which follow a pseudo-quadratic nature, thus the resulting fit will be more likely to maintain the local properties of the plant. Each MSMA model has as associated input region,

$$\mathcal{U}_{\pi} \in \mathcal{U} \tag{30}$$

Then using the same quadratic equations as before (Equations (25) and (26)), but with quadratic constants solved using the sub-regions and minimized against the single available model (as shown here using the nominal parameters),

$$\left[\boldsymbol{Q}_{\pi}^{\Phi}, \boldsymbol{L}_{\pi}^{\Phi}, \boldsymbol{C}_{\pi}^{\Phi}\right] = \min_{\boldsymbol{Q}, \boldsymbol{L}, \boldsymbol{C}} \sum_{\boldsymbol{u} \in \mathcal{U}_{\pi}} \left| \left| P_{2}(\boldsymbol{u}) - \Phi(\boldsymbol{u}, \boldsymbol{\theta}_{nom}) \right| \right|^{2},$$
(31)

$$\left[\mathbf{Q}_{\pi}^{G_{j}}, \mathbf{L}_{\pi}^{G_{j}}, \mathbf{C}_{\pi}^{G_{j}}\right] = \min_{\mathbf{Q}, \mathbf{L}, \mathbf{C}} \sum_{\mathbf{u} \in \mathcal{U}_{\pi}} \left\| P_{2}(\mathbf{u}) - G_{j}(\mathbf{u}, \boldsymbol{\theta}_{nom}) \right\|^{2}.$$
(32)

Using either method of using uniform or non-uniform grids, allowing these regions to overlap allows for several fitted models to represent each point in the input space. The proximity of these models can be used to formulate weights used in the processing of solutions, and fittings of regions which are far from the current operating point can be neglected entirely, resulting in fewer NLPs to solve at each iteration. Using uniform grids, the number of regions grows to the power of the number of inputs, therefore quickly becomes too many models to reasonably handle. Therefore, using a non-uniform approach to generating these regions may be necessary.

#### 4.4. Solution Production Illustrative Example

This example illustrates the three methods of generating multiple models used by MSMA. The methods produce the multiple solutions using models derived from a model of the plant. The model function used to illustrate these methods is a single variable constraint, taken from [29].

$$G(u, \theta) = (u-2)^2 - 3 - \theta_1 sin(\theta_2 u), \quad u \in [0, 4].$$
(33)

Where necessary, the value and uncertainty of the parameters are known and given as a closed set,

$$\boldsymbol{\vartheta} := \left\{ \boldsymbol{\theta} \in \mathbb{R}^2 \mid \begin{bmatrix} 1.7\\ 2.4 \end{bmatrix} \le \boldsymbol{\theta} \le \begin{bmatrix} 2.3\\ 2.6 \end{bmatrix} \right\}$$
(34)

where the nominal parameters are defined as,

$$\boldsymbol{\theta}_{nom} = \begin{bmatrix} 2\\ 2.5 \end{bmatrix}. \tag{35}$$

For the purposes of illustrating the solution producing approaches, 5 sets of model parameters are chosen which include the 4 extrema of the parameter set, and the nominal model. Figure 1 shows the three methods of generating the models used to generate the potential solutions. Figure 1a shows the set of models produced by using 5 sets of parameters from the known uncertainty. Figure 1b shows a set of 5 quadratic models, produced by fitting a quadratic function to each of the models from the set of known uncertainties in the parameters.

Finally, Figure 1c shows a set of quadratic models produced from a single set of parameters (here the nominal parameters), but each using a limited set of inputs. Where the original model curvature is convex (i.e., a postive definite Hessian), the fitted quadratic provides a good approximation of the model; and where the original model curvature is concave (i.e., a negative definite Hessian), the fitted quadratic is forced to be convex.



**Figure 1.** *G* vs. *u* for the sets of models for each of the three sets of solution production methods. The blue line represents the nominal model, and the gray line represents the set of models.

### 5. Processing the Proposed Solutions

The selection of the processing method strongly relates to the success of the method, with different objectives being achieved (or attempted to be achieved) by each. Some methods focus on improving the feasibility of iterates, improving rate and chance of convergence, or appropriate model selection.

# 5.1. Selection Based on Modifiers

One such method which has been proposed previously on dealing with multiple models is to use the relative magnitude of the modifiers for each model to select the most appropriate model. This approach has been suggested for handling different structures which provide more accurate approximations of the plant based on the location in the operating space [36,37]. Alternatively, it has also been suggested as a method of handling degradation in the plant which changes the underlying mechanisms of the plant, and thus changes the most appropriate model for use in the RTO scheme [38,39]. The standard approach to model selection is to define an overall modifier based on a weighted sum of the individual modifiers,

$$\Psi_{k,\pi} = w_{\varepsilon}^{\phi} \varepsilon_{k,\pi}^{\phi} + w_{\lambda}^{\phi} \lambda_{k,\pi}^{\phi} + w_{\varepsilon}^{g} \varepsilon_{k,\pi}^{g} + w_{\lambda}^{g} \lambda_{k,\pi}^{g}$$
(36)

where w refers to the weights. Then the model is selected based on the minimum overall modifier as follows,

$$f_{MS}(u_k, u_{k,\pi}^*) = (I - K)u_k + Ku_{k,\pi_k}^*,$$
(37)

$$\pi_k \in \arg\min_{\pi} \Psi_{k,\pi}.$$
(38)

By choosing the model based on the modifiers, the model can be selected before all solutions are solved, thus removing one of the primary concerns of MSMA, of solving many NLPs per iterate. As the final RTO solution is inherently a solution to one of the models which matches the estimates of the plant, then Condition 1 is met and KKT matching is guaranteed upon convergence if perfect estimates of the plant are available.

As a result of choosing the most appropriate NLP before solving for multiple solution, the information from the other models and solutions is discarded. The choice of the weights is highly influential to the success of the approach. The weights serve to bring all the modifiers to a similar order of magnitude, such that a single constraint or the objective does not dominate the overall modifier term. Additionally, they offer the ability to have certain constraints or the objective more influence over the choice of model. In the end, the choice of the weights are highly process specific and depend on the decisions of the operator. Alternatively, the magnitudes of each modifier can be ranked, and the sum of the ranks can be used to choose the model.

## 5.2. Selection Based on Central Tendencies

One of the simplest methods of combining the multiple solutions is to combine the proposed solutions from all the approaches based on the central tendencies. Three central tendencies are considered in this work, arithmetic mean, geometric median, and most likely. All of these allow for the whole set of data to be considered in the choice of the next operating point.

## 5.2.1. Arithmetic Mean

Similarly to the median, using the arithmetic mean, either all of the points can be considered equally, or through the use of weightings, certain operating points can be favored, potentially improving the convergence rate and feasibility of the iterates. The selection function using the arithmetic mean is as follows,

$$f_{MS}(\boldsymbol{u}_{k}, \boldsymbol{u}_{k,\pi}^{*}) = (\boldsymbol{I} - \boldsymbol{K})\boldsymbol{u}_{k} + \boldsymbol{K}\sum_{i=1}^{n_{\pi}} w_{i}\boldsymbol{u}_{k,i}^{*}$$
(39)

where  $w_i$  are weights which sum to 1. As with the median, using the mean can produce points which are not from the original set of potential operating points, thus Condition 1 may not be satisfied upon convergence. This can be overcome using a property of convex models,

**Proposition 1** (KKT matching using the convex models). Using any solution processing method which selects a point within the convex hull of the potential solutions along with a solution production method using convex NLPs in the MSMA scheme given by (14) and (15). If perfect estimates of the value of the plant constraints, and gradient of the plant objective and constraints are available, upon convergence the converged point is a KKT point of the plant.

**Proof of Proposition 1.** Consider the current operating point  $u_k$ . If this operating point is infeasible for the plant, i.e.,  $\exists j_v \in \{1, ..., n_g\}$  such that  $G_{j_v, p}(u_k) > 0$ . The modified convex NLP can be written as the sum of an affine term (which their equal to the estimates of the plant), and a curvature term,

$$G_{j_{v},k}(\boldsymbol{u}) = G_{j_{v},p}(\boldsymbol{u}_{k}) + \nabla_{\boldsymbol{u}}G_{j_{v},p}\Big|_{\boldsymbol{u}_{k}}(\boldsymbol{u}-\boldsymbol{u}_{k})^{\mathsf{T}} + [\text{Curvature term}].$$
(40)

The solution to the NLP using this modified constraint must have a constraint less than or equal to 0 (to be feasible), and the curvature term is positive as the base NLP is convex. Therefore, each solution in the set of potential operating points generated from the convex NLPs must be within the semi-infinite open region defined by the gradient of the violated constraint,

$$\nabla_{\boldsymbol{u}} G_{j_{\boldsymbol{v}},\boldsymbol{p}} \Big|_{\boldsymbol{u}} (\boldsymbol{u} - \boldsymbol{u}_k)^{\mathsf{T}} < 0.$$

$$\tag{41}$$

Therefore, the current operating point is outside of the convex hull of the potential solutions and cannot be the next operating point, and convergence cannot occur at an infeasible point of the plant.

Next, if this operating point is feasible for the plant, using the same logic as for the constraints, the set of potential operating points generated from the convex NLPs must either be within the semi-infinite open region defined by the gradient of the objective function,

$$\nabla_{\boldsymbol{u}}\phi_{\boldsymbol{p}}\Big|_{\boldsymbol{u}_{k}}(\boldsymbol{u}-\boldsymbol{u}_{k})^{\mathsf{T}}<0. \tag{42}$$

or at the current operating point. If one convex NLP suggests the current operating point as the solution to the MA problem given by Equations (14) and (15), then as all models, by definition, have the same first order properties as one another at the current operating

points, all potential operating points must be at the current operating point. Therefore, either the set of operating points must either be within the semi-infinite open region defined above, and convergence to  $u_k$  will not occur at the current operating point, or all potential points are at the current operating point, and Condition 1 is met, thus from Theorem 1 the converged operating point is a KKT point of the plant.  $\Box$ 

As the mean of a set is within the convex hull of the set, Proposition 1 is valid using the solution processing method given by Equation (39) if the solutions are produced using a set of convex models.

## 5.2.2. Geometric Median

Using the geometric median finds an operating point which balances the directions to each of the potential operating points in the set generated by the models, thus naturally handling outliers. The geometric median can be found by solving the following,

$$f_{MS}(\boldsymbol{u}_k, \boldsymbol{u}_{k,\pi}^*) = (\boldsymbol{I} - \boldsymbol{K})\boldsymbol{u}_k + \boldsymbol{K}\boldsymbol{u}_k^*$$
(43)

$$\boldsymbol{u}_{k}^{*} \in \arg\min_{\boldsymbol{u}} \sum_{i=1}^{n_{\pi}} w_{i} ||\boldsymbol{u} - \boldsymbol{u}_{k,i}||.$$

$$(44)$$

where  $w_i$  are weights which can optionally be used to favor certain models/solutions. Using the median can produce points which are not from the original set of potential operating points, thus upon convergence Condition 1 may not be satisfied. If convex models are used, Proposition 1 is satisfied, but this can also be rectified by limiting the minimization to the domain of the potential operating points, thus finding the potential solution which the best approximates the median, i.e.,

$$u_{k}^{*} \in \arg\min_{u \in u_{k,\pi}^{*}} \sum_{i=1}^{n_{\pi}} w_{i} ||u - u_{k,i}||.$$
(45)

Using the limited median satisfies Condition 1, and KKT matching is guaranteed upon convergence if perfect estimates of the plant are available.

# 5.2.3. Most Likely

Another approach to selecting the most appropriate solution is through the use of the most likely operating point via a distribution or a mixture distribution. By fitting a distribution to the set of model solutions, typically one for each structure forming a mixture distribution, and using this to select the next operating point. When using a single model which follows a close to normal distribution of points, the resulting solution will be identical to using averages. However, if multiple structures (and multiple distributions) are used, then the most likely point will not be between the two distributions but within one of them.

For a mixture distribution, for example one for each structure of model available, if each structure suggests a distinct set of potential operating points, then the most likely point from the overall distribution will depend on the variance, with the individual distribution with the least variance being favored. This is not necessarily the best choice from the set of points, as the most appropriate structure, and thus the better operating point, may have more variance in the suggested points.

The choice of distribution depends on the set of operating points obtained, but as convergence is approached, the set of points will approach a singular point, therefore the chosen distribution should be able to accurately handle this case, such as Gaussian (normal) or skew-Gaussian (skew-normal) distributions. The simplest case is using a multivariate

Gaussian mixture model and the probability for a given set of operating conditions from this distribution is given by the following,

$$\boldsymbol{U} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{D} w_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \qquad (46)$$

where *D* is the number of distributions, *w* is a weighting factor,  $\mu$  is the distribution mean, and  $\Sigma$  is the distribution covariance matrix for distribution *i*.

The sum of the weights must equal 1 for the probabilities to remain meaningful, i.e., between 0 and 1, but as is seen in the next equation, the relative magnitudes are more important.  $f_{U}$  refers to the overall PDF of the mixture distribution, and can be used to find the mode of the distribution. The final operating point is filtered from the point of maximum likelihood,

$$f_{MS}(u_k, u_{k,\pi}^*) = (I - K)u_k + Ku_k^*,$$
(47)

$$\boldsymbol{u}_k^* \in \arg\max_{\boldsymbol{u}} f_{\boldsymbol{U}}(\boldsymbol{u}). \tag{48}$$

Alternatively, Condition 1 can be enforced if the maximum is limited to the set of suggested operating points, i.e.,

$$\boldsymbol{u}_{k}^{*} \in \arg \max_{\boldsymbol{u} \in \boldsymbol{u}_{k,\pi}^{*}} f_{\boldsymbol{U}}(\boldsymbol{u}).$$
(49)

## 5.3. Selection Based on Closeness

The final method to combining the solutions is considering the distance between the current operating point and the potential operating points. Taking larger steps in the input space is risky, as the only known of the plant is the current operating point. Therefore, a potential choice from the set of solutions is the closest to the current operating point. This is defined by using a normalized distance,

$$f_{MS}(\boldsymbol{u}_k, \boldsymbol{u}_{k,\pi}^*) = (\boldsymbol{I} - \boldsymbol{K})\boldsymbol{u}_k + \boldsymbol{K}\boldsymbol{u}_k^*, \tag{50}$$

$$\boldsymbol{u}_{k}^{*} \in \arg\min_{\boldsymbol{u} \in \boldsymbol{u}_{k,\pi}^{*}} ||\boldsymbol{u} - \boldsymbol{u}_{k}||.$$
(51)

As the closest operating point is inherently one of the potential operating points suggested by the models, Condition 1 is satisfied, and KKT matching is guaranteed upon convergence if perfect estimates of the plant are available. Alternatively to selecting the closest solution, if convex NLPs are used, the closest operating point within the convex hull of the set of potential solutions can be used.

# 5.4. Solution Processing Illustrative Example

A two variable,  $u = [u_1, u_2]$ , example has been developed to illustates the approaches above of combining a set of potential operating points into a single point for use by the RTO scheme. To allow full comparison of the solution processing, all methods are illustrated using the solutions produced using the parametric uncertainty, and all weights for the modifiers, solutions and distributions are discarded, i.e., equal weightings. In this example, structural plant-model mismatch is introduced via the non-linear function  $\zeta(u_1)$ , which is applied to the first input. The system cost and constraints are given via parametric functions,

$$\Phi(\boldsymbol{u},\boldsymbol{\theta}) = 4 - \theta_1 \zeta(\boldsymbol{u}_1) - \boldsymbol{u}_2, \tag{52}$$

$$G(\boldsymbol{u},\boldsymbol{\theta}) = \theta_2(\zeta(u_1) - 0.9u_2)^2 + u_2 - 1.8.$$
(53)

where

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \tag{54}$$

are the parameters. The model has two different potential structures of  $\zeta(u_1)$  as given:

(Plant) 
$$\zeta(u_1) = e^{u_1} + e^{-u_1} - 1,$$
 (55)

(Model 1) 
$$\zeta(u_1) = u_1^2 + 1,$$
 (56)

(Model 2) 
$$\zeta(u_1) = \sqrt{3u_1^2 + 1}.$$
 (57)

The plant is accurately described using the following parameters,

$$\boldsymbol{\theta}_p = \begin{bmatrix} 1\\ 1.5 \end{bmatrix}. \tag{58}$$

The current operating point,  $u_k$ , is defined to be at the point [0.5, 1.5]. Figure 2 illustrates the current plant objective and constraint functions with the true plant optimum and current operating point.



**Figure 2.** Example 2 plant NLP. The red cross represents the true optimum, the red line represents  $G_p = 0$ , and the grey contour represents the plant objective function.

The set of parameters used to formulate the models is formed of the choice of either structure 1 or 2 for  $\zeta$ , and a value of  $\theta$  taken from a normal distribution. The choice of these parameters is based off a comparison to the true plant constraint and objective to account for some of the structural mismatch in  $\zeta(u_1)$ ,

$$\vartheta_1 \sim \mathcal{N}\left( \begin{bmatrix} 1\\ 1.8 \end{bmatrix}, \begin{bmatrix} 0.2 & 0\\ 0 & 0.2 \end{bmatrix} \right),$$
(59)

$$\vartheta_2 \sim \mathcal{N}\left( \begin{bmatrix} 0.9\\ 1.8 \end{bmatrix}, \begin{bmatrix} 0.5 & 0\\ 0 & 0.3 \end{bmatrix} \right)$$
(60)

where  $\vartheta_1$  and  $\vartheta_2$  represent the set of parameters for model structures 1 and 2, respectively. The set of points used by the solution processing function is generated by taking 30 sets of parameters for each of the two model structures from the above distribution, generating 60 potential operating points.

Figure 3 shows the solution from using each of the three different approaches discussed in the previous section. Firstly, selection based on the modifiers suggests a point which is relatively close to the most likely point. Next, selection based on using either the mean or the median suggests a point which is also relatively close to the plant optimum, but is not guaranteed to be a point suggested by any of the models in the set of potential operating points. Selection based on closest is the only method which produces a point which is feasible for the plant (before filtering), and also suggests a potential operating point from the set of models, thus guaranteed to satisfy Condition 1.





Finally, selection based on most likely utilizes a fitted distribution. This fitted distribution is formed of a mixture Gaussian distribution, formed of two Gaussian distributions, one fitted to each of the model structures. The resulting most-likely solution is at the point where both distributions overlap. As this problem is non-linear, despite the error in the parameters being characterized as being normally distributed, the resulting set of solutions are not. Clearly there is a skewness to the data, with a bunching of points, with a long tail of other solutions. As a standard Gaussian distribution has no skewness, this tail will draw the distribution from the true density of the solutions, therefore the most-likely point obtained from the mixture Gaussian fit is a poor representation of the true most-likely point, and is one of the key issues with dealing with distributions and non-linear functions.

# 5.5. Recommended Schemes

A wide range of options have been suggested for the production and processing of the solutions in the MSMA framework. In this penultimate section, some combinations which work well together are highlighted with some comments on their effectiveness.

1. Selection based on modifiers using known model uncertainties. The first suggested approach uses the modifiers to determine the NLP to use for the RTO scheme before the solutions are solved. By using the known uncertainties in the model parameters, the choice of NLP (and effectively the choice of model parameters) becomes a rudimentary parameter estimator. The prime advantage of this approach is the reduced computation required per RTO iteration, with only the additional calculation of the modifiers of the set of models over standard MA. This comes at the expense of removing the information contained in the set of potential operating points (as they are discarded before being solved).

2. Selection based on central tendencies using quadratic approximations. The next suggested approach uses the central tendencies of the set of solutions generated by either region-based or uncertainty based models. The use of convex quadratic models allows for the solutions to be found fast, and the guarantee of KKT matching upon convergence is maintained by the approach.

3. Selection based on closeness using known model uncertainties. The final suggested approach is to take advantage of the closest operating point. Using the closest point is considered the safest option suggested and can be combined with using the model uncertainties to find the true optimum of the plant. This approach is the most computationally demanding of the suggested approaches, but can be supplemented with quadratic approximations or lower fidelity models to reduce the computation times, or by refining the set of models to reduce the number of potential solutions.

#### 5.6. Analysis of Schemes

The primary limitation to using MSMA approaches (other than using the modifiers to find the most appropriate model) is that in order to find the set of potential operating points, many model NLPs must be solved at each iteration of the RTO scheme. Therefore, if the model is of high-fidelity, this is a costly and time-consuming task which may increase the total RTO convergence time if it takes significantly longer than the convergence time of the plant.

The MA approach adds affine modifiers to the model which results in an NLP which accurately predicts the local aspects of the plant, but can result in poor predictions of global picture of the plant. Therefore, using high-fidelity models in the MA scheme may produce artifacts in the optimization functions, such as local minima, which are not of significance when looking for the true plant optimum, and may slow down the convergence of the RTO scheme. The use of lower-fidelity models is favored when this is the case as the global properties of the plant are maintained without these artifacts. Another advantage to using lower-fidelity models which is relevant to MSMA is the solve time for each NLP is significantly reduced, resulting in RTO solve times which do not slow down the convergence of the RTO scheme.

Using the multi-model schemes which are recommended have the potential to improve the convergence properties of MA. This will be analyzed by looking firstly at the model adequacy conditions of the proposed schemes. For this article, a model is considered adequate if convergence to the plant optimum is possible (i.e., if the current operating point is the plant optimum, then the solution to the RTO scheme is also at this operating point). For standard MA, a necessary condition for this is that the model has a positive definite Hessian of the Lagrangian at the plant optimum, but this is only sufficient to guarantee that the current operating point is a local minimum of the NLP, and not a global minimum.

For using the modifiers as the selection method, the model is adequate if, when operating at the plant optimum, the model which has the minimum overall modifier is adequate under standard MA. That is that the chosen model by the solution processing equation has a positive definite Hessian of the Lagrangian at the plant optimum. The basis of this selection method is that the model with the minimum overall modifier will more closely match the plant. Therefore, this can also apply to the Hessian of the Lagrangian of the chosen model, where it may be more similar to the plant, which will be positive definite at the plant optimum. This could provide a better level of guarantee of model adequacy than standard MA, but is case dependent and not guaranteed.

For using central tendencies, it depends on the method used. For the geometric median, if at least half of the NLPs are adequate, then model adequacy is also satisfied for the median (as half the set of potential operating points will all suggest the current operating point, which will then equal the median). This is not required as under certain conditions, solutions from non-adequate NLPs will cancel one another out in the median calculation. For the mean and most likely, as every potential solution contributes to the next operating point, model adequacy is only obtained if all individual models have a positive definite Lagrangian at the plant optimum. In reality, if measurement noise is considered, as long as the majority of the model NLPs are adequate, good convergence stability can be observed. If quadratic approximations are used then every individual model will have a positive definite Lagrangian, therefore when used in conjunction with the central tendency methods, model adequacy is guaranteed.

Finally, considering the closest operating point. If any model in the set of NLPs satisfies the model adequacy of standard MA, then this will be produce a solution at the current operating point, and therefore be the closest solution, and the model adequacy is guaranteed.

The other convergence properties which may improve when using multi-model approaches is an increased rate of convergence and a increased likelihood that the next iteration is feasible for the plant. These two properties are highly linked, as typically an increase to the rate of convergence is determined by the magnitude of the filter used by the scheme, which can be made more aggressive if the probability of safety is increased. Proof of improvement to these properties cannot be done mathematically without making further assumptions about the mismatch between the plant and model, which is typically not possible to check in reality. This is because under the current assumption of structural mismatch, there can always be an edge case where improvement is not observed (e.g., in cases where standard MA lucks out and finds the true plant optimum in a single iteration, which cannot be improved upon).

These schemes also depend on the accuracy of the set of models used to define the potential solutions. Poor performance may be observed if an inaccurate model used to produce one of the potential solutions, for example if a model constraint with a significantly larger second derivative is used, then this will likely be the closest solution, and result in a slow convergence. Despite this lack of mathematical proof of improvement, what can be looked into is several case studies of differing complexity and compare the performance of the proposed schemes to standard MA. This is not rigorous, but does highlight where improvement may be observed, and potentially why it is observed. For this reason, three different case studies will be analyzed in the following section, with a CSTR, a distillation column, and a semi-batch reactor.

## 6. Case Studies

Each of the three recommended schemes are illustrated on three different case studies of varying difficulty. The first is a benchmark case study in RTO schemes known as the Williams-Otto CSTR [8,40]. This CSTR has 6 state equations, and 3 inputs, where the nominal model is inadequate for use with standard MA. The second case is a distillation column for the separation of methanol from n-propanol. This system has over 200 state equations for the plant, therefore is more computationally challenging. The final case study looks at a semi-batch bioreactor with two inputs which are controlled over the whole batch time. After discretization, a higher-dimensional problem is produced, with a total number of inputs of 12. At the end of this section, some comments are made about what improvement may be observed using the MSMA approaches.

For each case study, a simulation of the plant is run, and steady-state measurements are taken at the current operating point which are used to replicate taking measurements in practice. Therefore, knowledge of the set of equations used to simulate the plant is not known/used by the RTO schemes. In these case studies, measurement noise, disturbances, or other errors have not been considered. This produces an idealized scenario which cannot be realistically met in industrial applications. This idealized simulation does still provide valuable insight into the proposed approaches, where the rate of convergence, stability of convergence (from a mathematical standpoint), and robustness against infeasible operation caused by plant-model mismatch can all be assessed.

#### 6.1. Williams-Otto CSTR

Each of the three recommended schemes are illustrated on the Williams-Otto case study [40], which is a benchmark case study for RTO schemes.

#### 6.1.1. System Overview

This problem is concerned with the optimization of a CSTR which converts two reactants, *A* and *B*, into desired products, *P* and *E*. There is also an intermediate component,

*C*, and an undesired product, *G*. The system of equations which determine the states of the plant can be written as the following three reactions,

$$A + B \to C, \tag{61}$$

$$B + C \to P + E$$
. (62)

$$C + P \to G,$$
 (63)

The operating conditions are controlled via three independent inputs. These inputs are the flow rates of the reactants in kg h<sup>-1</sup>, and the temperature of the reactor in °C, given as  $u = [F_A, F_B, T]$ . The states of this system are the mass fractions of the components, and the measured outputs are the states. The CSTR is assumed to be ideal, and the reactions are assumed to be elementary.

The model of this system is given as a simplified reaction mechanism with only two reactions [8],

$$A + 2B \to P + E, \tag{64}$$

$$A + B + P \to G, \tag{65}$$

The NLP to be solved is as follows,

$$\boldsymbol{u}^* = \arg \max \quad J(\boldsymbol{u}, \boldsymbol{y}_p(\boldsymbol{u})), \tag{66}$$

s.t. 
$$g(u, y_p(u)) = X_G - 0.08 \le 0,$$
 (67)

where *J* is the objective function, and *g* is the constraint. The aim of this NLP is the maximize the objective function which is defined with respect to the profit of the system, given as,

$$U = 1143.38X_PF + 25.92X_EF - 76.23F_A - 114.34F_B.$$
 (68)

where  $F = F_A + F_B$  is the overall flow rate of the system. The inputs are limited to the following input space,

$$\begin{cases} 3\\6\\80 \end{bmatrix} \leq \begin{bmatrix} F_A\\F_B\\T \end{bmatrix} \leq \begin{bmatrix} 4.5\\11\\105 \end{bmatrix}.$$
 (69)

The parameters used for the plant and model, along with the state equations are given in Appendix A. The known uncertainty in the parameters is confined to the activation energies of the two reactions, which is characterized via a joint normal distribution. The set of models is formed of 8 sets of parameters which are one standard deviation from the mean, 12 sets of parameters which are two standard deviations from the mean, and the nominal model, forming a uniform set of 21 models.

#### 6.1.2. Standard MA

The standard MA approach using the nominal parameters for the model is illustrated in Figure 4. The resulting RTO trajectory does not convergence to the plant optimum when using either a high input gain (K = 0.7I), or a low gain (K = 0.3I). It has been shown that using the nominal parameters, the Hessian of the Lagrangian is positive definite at the plant optimum, however when operating at the plant optimum, the modified NLP has a global minimum away from the plant optimum, thus convergence cannot occur [28].



**Figure 4.** WO case study using the standard MA approach using a high gain and a low gain, both producing unstable solutions.

#### 6.1.3. MSMA: Selection Based on Modifiers

The first approach investigated uses the modifiers to determine which model is the most appropriate for use in the MA scheme. The difference between the plant and model value and gradients of the objective function and constraint functions are considered. In order to make these values comparable, the relative difference for each quantity is defined, and then ranked from lowest to highest. The sum of these ranks is used to define the model used for the next iteration of the MA scheme (with the lowest overall rank being chosen).

Using this approach with an input filter of K = 0.7I the true optimum can be found, as shown in Figure 5. Initially this scheme does not perform well, and is likely because the chosen model does not direct the RTO scheme in the correct direction. However, an appropriate model is found, and convergence occurs after around 10 iterations, which is not observed by standard MA. Using the modifiers to select the model used has a fundamental flaw in that no matter which model is selected, they are all corrected such that the value and gradient match the plant. Therefore, ideally choosing a model which has similar second order properties to the plant is desired. However, this is not known for the plant without further experiments, and even if it was known, it could be corrected for [41], rather than used as a basis for model selection.



**Figure 5.** WO case study using the minimum modifier selection method, compared to standard MA using the same input filter.

This comes at the profit of being able to apply this approach to virtually any scenario for which standard MA is applicable, as the number of experiments required is similar, and the computing power required for each RTO calculation is comparable.

#### 6.1.4. MSMA: Selection Based on Central Tendencies

The second approach investigated relates to the use of central tendencies. For this case, the model uncertainties are assumed to be unknown, and only the set of nominal parameters are known, therefore using region-based quadratic approximations is applicable. These regions are formulated firstly by gridding the input space, here an  $11 \times 11 \times 11$  grid

is used. This grid can be used for form overlapping regions, here each region is formed of a  $4 \times 4 \times 4$  set of points, resulting in  $8^3$  regions. At each region, a convex second order polynomial w.r.t. the inputs is fit to the region, and the quadratic term is saved. As the system is relatively simple, solving the state equation for the model can be done rapidly (in this case a few milliseconds), therefore calculating the constants used in the parameters to formulate the set convex regions takes under a minute wall clock time.

At each operating point, the set of regions which have used points local to this operating point are used to find the set of potential operating points. Most points have 27 overlapping regions, but towards the edge of the input space, fewer regions exist. The mean of the potential operating points is taken as the next operating point, using an input filter gain of K = 0.7I. As the models used are convex, the model adequacy conditions are guaranteed, and this approach rapidly converges to the true optimum of the plant without violating the constraints, as shown in Figure 6. As at each iteration, only up to 27 convex quadratic NLPs are solved, this can be solved rapidly.



**Figure 6.** WO case study using the mean solution selection method and regional quadratics, compared to standard MA using the same input filter.

## 6.1.5. MSMA: Selection Based on Closeness

The final approach investigated looks at using the closest potential operating point to the current operating point as the next RTO iteration. This approach requires solving the optimum of all the models for each iteration, and selecting the closest to the current operating point, which is more computationally intensive than the previous approaches. As each NLP is approximately the same fidelity as the nominal model, this takes in the order 20 times longer than for standard MA. This approach does however successfully convergence to the true optimum faster than the other approaches using an input filter gain of K = 0.7I, as shown in Figure 7.



**Figure 7.** WO case study using the closest solution selection method, compared to standard MA using the same input filter.

#### 6.2. Distillation Column

As a more elaborate example with more complex dynamics to illustrate the proposed methods on, a simulation of a distillation column has been simulated [23,42]. This example is of a high purity binary distillation column, used in the separation of methanol and

n-propanol. The column has 40 trays (N = 40), with the feed on tray 21 ( $N_f = 21$ ), a partial reboiler, and total condenser. The open-loop system inputs are the reboiler heat duty, Q, and the reflux volumetric flow rate,  $L_{vol}$ . A PI control scheme is in place based around controlling the temperatures on the key stages of trays 14 and 28.

#### 6.2.1. System Overview

A detailed description of the plant simulation is given in [23]. To summarize, this 204 state DAE is formed of 82 differential states and 122 algebraic states. The differential states are formed of 42 composition states, and 40 molar holdup states. Whilst the algebraic states are formed of 41 volumetric fluxes, 40 liquid fluxes, and 41 temperatures.

Significant structural changes have been given to the model to provide structural mismatch to the plant. The model is based off a distillation column with 100% efficient trays, but with a fewer number of trays, such that the overall separation is comparable. Therefore, the nominal model has a rectifying section with 9 trays and a stripping section with 14 trays. Other simplifications include,

- 1. The molar-hold up of the trays are assumed to be constant
- 2. The reboiler and condenser molar holdup are assumed to be constant (with variable volumetric holdup)
- 3. The condenser is assumed to operate at the saturated liquid temperature
- 4. Incorrect feed concentration and flow rate

The resulting model has 71 states. Additionally, the feed is assumed to be fixed with a flow rate of  $14 \text{ L} \text{ h}^{-1}$  and a concentration of 0.32 at 71 °C, however the true flow rate into the plant is with a concentration of 0.30, and a flow rate of  $14.93 \text{ L} \text{ h}^{-1}$ . The uncertainty in the model is confined to the number of trays in the rectifying section, with the range being from 7 to 11. This forms a pool of five models to be used by the proposed multiple model schemes.

The NLP for the system is based on the flow rates of the bottoms and distillate, and the cost of the heat flux to the reboiler. Typically the objective of a separation process like this one will solely depend on the heat flux, however in this case it would result in an optimum solely defined by the constraints. The constraints on the system relate to the concentration of the bottoms and distillate. The NLP for the system is as follows,

$$\min_{\mathbf{r}} \quad \phi(\mathbf{u}(\mathbf{r}), \mathbf{y}(\mathbf{r})) := 0.12Q - 23040D - 86400B + 3.7 \tag{70a}$$

s.t. 
$$g(u(r), y(r)) := [x_B - 0.005, 0.986 - x_D] \le 0,$$
 (70b)

$$T_{28,s} \in [70,74], \quad T_{14,s} \in [82,86].$$
 (70c)

where *x* is the mass fraction of methanol, *B* and *D* are the bottoms and distillate flow rates, respectively,  $\mathbf{r} = [T_{28,s}, T_{14,s}]$ , and  $T_{28,s}$  and  $T_{14,s}$  are the set points to the temperatures in the key trays of the rectifying section and the stripping section, respectively.

# 6.2.2. Standard MA

The first approach is using standard MA which will give a baseline for the performance of MA schemes on the distillation column defined above. The three characteristics which are used to gauge the performance of the scheme are the controller set points, the constraint function values, and the objective function value. The controller set points are displayed as their normalized forms, which are defined as follows,

$$\overline{T}_{28} = \frac{T_{28} - 70}{4},\tag{71}$$

$$\overline{T}_{14} = \frac{T_{14} - 82}{4}.$$
(72)

The performance of the standard MA scheme is illustrated in Figure 8 using both an input filter of K = 0.9I, and K = 0.6I. With the high gain, the system does not



converge, but does approach a cycle with similar plant profit to the true plant optimum, without breaking the constraints. The RTO trajectory with the reduced filter approaches the optimum more slowly, but does successfully converge to the plant to optimum.

**Figure 8.** Distillation column RTO trajectory for standard MA. Top row, unstable solution using K = 0.9I. Bottom row, stable solution using K = 0.6I.

# 6.2.3. MSMA: Selection Based on Modifiers

The first of the MSMA scheme investigated is the minimum modifier approach, where the solution is selected based on the relative magnitudes of the modifiers. The performance of the minimum modifier MSMA scheme is illustrated in Figure 9, where the scheme successfully converges to the optimum with a decaying oscillation in the set points around the true plant optimum. The choice of model NLP by the minimum modifier selection is where the number of rectifying trays is 8, other than the second iteration, where 9 trays is preferred.



**Figure 9.** Distillation column RTO trajectory for MSMA using the minimum modifier solution approach shown as the red diamond line. Stable solution for standard MA is shown for comparison as the grey circle line.

This result is an improvement over standard MA, where a lower input gain was required in order to achieve stability in the convergence, which slowed the convergence rate. This only comes at the additional cost of calculating the modifiers of each of the 5 models.

# 6.2.4. MSMA: Selection Based on Central Tendencies

The next MSMA approach uses regional quadratic approximations of the nominal model to generate the solutions, and uses the mean of these quadratics to generate the next RTO solution. The regions are formed from a uniform  $11 \times 11$  gridded set point space, and each region is formed of a  $4 \times 4$  region within this uniform gird.

The performance of the regional quadratic MSMA scheme is illustrated in Figure 10, where the scheme successfully converges to the optimum. As the regional quadratics used to find the set of potential operating points are bounded to a small subspace of the true set point space, the magnitude of step-size for some of the solutions is short, thus the resulting trajectory is more cautious than other MSMA approaches, and similar to that of the reduced gain standard MA.



**Figure 10.** Distillation column RTO trajectory for MSMA using the mean solution approach shown as the blue square line. Stable solution for standard MA is shown for comparison as the grey circle line.

# 6.2.5. MSMA: Selection Based on Closeness

The final MSMA approach uses the closest solution from the set of solutions to define the next operating point. The performance of the closest MSMA scheme is illustrated in Figure 11, where the scheme successfully converges to the optimum faster than any of the previous approaches, without any instability or oscillation in the set points. The choice of model NLP is where the number of rectifying trays is 7 for the whole trajectory, other than the fifth iteration, where 11 trays is preferred.



**Figure 11.** Distillation column RTO trajectory for MSMA using the closest solution approach shown as the green triangle line. Stable solution for standard MA is shown for comparison as the grey circle line.

# 6.2.6. Analysis of Computational Complexity

For this case study, one method to solve the steady state of the column for a given set of inputs is to use an ODE solver over a long time horizon until steady-state is reached. This is initialized by running the model of the column at the current operating point and using these saved states as a basis for each model run for the current iteration. This also has the benefit of being more likely to achieve the same steady-state as the plant if multiple steady-states exist for a given set of inputs. This analysis is carried out by firstly looking at the computation time of the model and standard MA. As each RTO scheme is run on the same machinery, the relative complexity of each proposed method can be compared to one another.

A single solving of the nominal model depends on the current states and relative change to the input, but is of the order of 0.5 s to 1 s wall time. Each of the models in the set takes roughly a similar amount of time as one another, with the shorter columns being marginally faster than larger columns as they have fewer states. As the system only has two inputs, the number of function evaluations is relatively low, with only 30–100 model simulations required to find the optimum of the model, taking approximately 30 s. For standard MA, this is very similar, with the linear modifier term taking negligible time to solve.

The computation required to solve the modifiers for most MA schemes can be carried out as soon as the previous operating point is found. This is because the majority of the computation required is in solving several steady-state simulations of the model (if done numerically), which do not depend on the measured/estimated values of the plant. Therefore, the additional modifier calculations associated with the method of using the modifiers to determine the solution (and the closest solution method) will not cause additional time operating suboptimally. As the chosen model has fewer states than the nominal model, the overall computation time is marginally faster than standard MA.

As for use of the regional quadratics, pre-processing of the pool of models is required to create the set of convex models used by this scheme. This requires running the nominal model at each point in the gridded input space (121 simulations, approximately 110 s), then fitting the quadratic for each region (512 regions, approximately 25 s), taking 135 s in total and requires 45 kB of data to be saved. Solving the quadratic NLPs is very fast in comparison to the model column. Even solving the minimization of all maximum number of 27 nearby quadratics only takes between 1 s to 2 s per iteration, or over 10 times less computation time as standard MA.

Finally, for the closest solution, solving the individual NLPs takes a similar solving time as for the nominal one, therefore the total computation time per iteration is approximately 5 times longer that of standard MA.

Overall, this lines up well with the discussion in the previous sections, where using the modifiers results in similar complexity to standard MA, using regional quadratics on low dimensional problems is faster than standard MA, and using the closest solution is slower than MA.

# 6.3. Semi-Batch Bioreactor

The final case study looks at a semi-batch reactor. An example of a semi-batch reactor is the photobioreactor for the production of phycocyanin [43], which has been investigated by other RTO [44] and MPC schemes [45]. Two of the three recommended multiple solution MA approaches (of using the modifiers, and closeness for selection) are applied, along with a third scheme of using the central tendencies without convex NLPs to investigate why this is not a recommended scheme.

# 6.3.1. System Overview

This reactor uses the blue-green cyanobacterium known as Arthrospira platensis as the biomass (X), which is grown on nitrates (N) to produce the phycocyanin (P). The primary objective of this reactor is to maximize the concentration of P at the end of the batch cycle (t = 240 h), whilst meeting the path and terminal constraints. The initial conditions are fixed, and the decision variables are the light intensity throughout the batch cycle I(t) [µmol m<sup>-2</sup> s<sup>-1</sup>], and the nitrate inflow rate  $F_N(t)$  [mg L<sup>-1</sup> h<sup>-1</sup>]. These continuous variables can be discretized such that the optimization can be carried out. The easiest method of discretizing the inputs is to break the batch cycle into 6 uniform time segments of 40 h each, and during each time segment, the inputs are held constant. Therefore the system has 12 decision variables.

The constraints on the system consist of two path constraints, and a terminal constraint. The first path constraint is on the phycocyanin-to-biomass ratio, which must be kept below  $11 \text{ mg g}^{-1}$ , while the second path constraint is that the nitrate concentration is kept below  $800 \text{ mg L}^{-1}$ . The terminal constraint is that the nitrate concentration is below  $150 \text{ mg L}^{-1}$ . The initial conditions of the reactor are such that the concentration of the biomass is at  $1 \text{ g L}^{-1}$ , the nitrate is at  $150 \text{ mg L}^{-1}$ , and the phycocyanin is at  $0 \text{ mg L}^{-1}$ . Measurements of the concentrations of the components are taken at regular intervals of 10 h, therefore the NLP for the system can be defined as follows,

- min  $\phi(u(t), y(t)) := C_P(240)$  (73a)
- s.t.  $C_P(0) = 0$ ,  $C_X(0) = 1$ ,  $C_N(0) = 150$ , (73b)  $g_1(u(t), y(t)) := C_P(t) - 11C_X(t) \le 0$   $t \in \{0, 10, \dots, 240\}$ , (73c)  $g_2(u(t), y(t)) := C_N(t) - 800 \le 0$   $t \in \{0, 10, \dots, 240\}$ , (73d)  $g_3(u(t), y(t)) := C_N(240)150 \le 0$ , (73e)
  - $120 \le I(t) \le 400, \quad 0 \le F_N(t) \le 40. \tag{73f}$

where  $C_X(t)$  is the concentration of the biomass in  $[gL^{-1}]$ ,  $C_N$  is the concentration of nitrate in  $[mgL^{-1}]$ , and  $C_P$  is the concentration of phycocyanin in  $[mgL^{-1}]$ , and are the measured states of the system. u(t) represents the inputs of  $[I(t), F_N(t)]$ , and y(t) are the measured outputs of the concentration of each component at each time interval. Therefore, there is a total of 51 discrete constraints on the system.

A detailed review of the simulation used for this case study is given in Appendix B. The objective of the plant at the optimal conditions is illustrated in Figure 12a, where the black line shows the concentration of *P*. The objective of the system is to maximize the terminal concentration, which for the optimal conditions is  $C_P(240) = 0.1890 \text{ mg L}^{-1}$ . The first path constraint is also illustrated in Figure 12a, where the value of  $C_P$  is limited to be below the value of  $11C_X$ , which is only active at t = 240 h. The second path constraint and the terminal constraint are illustrated in Figure 12b, where the value of  $C_N$  is limited to be below 800 mg L<sup>-1</sup>, which is only active at t = 50 h and t = 100 h. Therefore, there are three active constraints, where the following are true:

$$C_P(240) - 11C_X(240) = 0, (74)$$

$$C_N(50) - 800 = 0, (75)$$

$$C_N(100) - 800 = 0. (76)$$

The model is assumed to follow a simplified set of state equations, where part of the light intensity term is neglected, leading to structural mismatch between the plant and model. In addition to the structural mismatch, there is an error in one of the parameters,  $Y_{N/X}$ , which is 20% higher than the true value used to simulate the plant. The pool of models is developed through the uncertainty in three of the parameters, including  $Y_{N/X}$ . The other parameters match those used to simulate the plant. This is given in more detail in Appendix B.

From this uncertainty, the pool of models is formed from 9 discrete set of parameters. These are produced from all of the combinations from the extreme values of the uncertain parameters (i.e., mean plus the uncertainty), producing 8 different sets, plus the set of parameters using the mean values.



Figure 12. Semi-batch reactor plant optimum conditions.

# 6.3.2. Standard MA

Firstly, as a benchmark to compare the proposed approaches to, the standard MA approach is applied to this case study. As with the other case studies, noise and errors in the gradient estimation methods are not considered, therefore it is assumed that measurements of the plant gradient can be taken at the current operating point, and all measurements are perfect without error. This is unrealistic, but allows for the convergence and robustness of the approach to be compared at ideal conditions.

The results of the standard MA approach with an input gain of K = 0.5I are shown in Figure 13. For the inputs, the 10 of the 12 inputs converge to near optimal points, however, the value of the light intensity for the fourth and sixth time ranges do not converge and are unstable. The value of the inputs for the flowrate of nitrate after t = 80 follow the same RTO iterates, as the solution to the modified model always predicts  $F_N = 40$  as the optimum, therefore they all exponentially approaches this solution, and all four lines fully overlap.



**Figure 13.** Standard MA inputs and NLP functions. Top row, unstable inputs shown with a dashed line, stable with a solid line. Bottom row, objective and constraints which are active at plant optimum.

As the instability in the inputs comes from times after t = 120, the active path constraints earlier than this are stable and converge near to the plant optimum. However, the active constraint after this time are unstable and infeasible, as shown in Figure 13d.

Running at lower input gains does not solve the instability in the inputs, which suggests the model is inadequate. Analyzing the Lagrange of the modified model at the plant optimum, there exists vectors which are orthogonal to the active constraints (and to the active input limits) which do not satisfy the second order conditions of optimality, therefore this model is not adequate.

# 6.3.3. MSMA: Selection Based on Modifiers

The first approach is the minimum modifier selection method, as illustrated in Figure 14, using an input filter gain of K = 0.5I. This method successfully overcomes the model adequacy issues of standard MA and rapidly converges to the plant optimum in less than 10 iterations. The RTO trajectory slightly breaks the active path constraint,  $g_1(240)$ , as shown in Figure 14d, and has some erratic behavior with the light intensity in the first few iterations, but overall is a big improvement over standard MA.



**Figure 14.** Multi-solution MA inputs and NLP functions using the minimum modifier solution approach shown as the red diamond line. Top row, inputs. Bottom row, objective and constraints which are active at plant optimum, where the standard MA approach is shown as the grey circle line.

# 6.3.4. MSMA: Selection Based on Central Tendencies

Next, the MSMA approach using the mean solution is applied using the full models (i.e., non-convex NLPs). As is discussed in the previous sections, the nominal model is inadequate for this system, and it is included in the set of models. The analysis of the model adequacy conditions for the mean selection method has shown that this method should also fail to converge.

As illustrated in Figure 15, the RTO approach does fail to converge using an input filter gain of K = 0.5I, hence why this is not a recommend approach. As only three of the nine models in the set are adequate, the instability is even greater than that of standard MA. This highlights both that using central tendencies without convex models comes with additional risks, and that the selection of the set of models used to produce the potential solutions is important to the method. For the value of  $Y_{N/X}$ , the uncertainty is approximately  $\pm 20\%$  of the true value, but the nominal value is already 20% over the true value, therefore some models use the value of  $Y_{N/X}$  that is 40% over the true value used to simulate the plant. Making sure every model in the set does somewhat accurately model the plant is important, especially for using the mean and most likely, which take all the models into consideration.



**Figure 15.** Multi-solution MA inputs and NLP functions using the mean solution approach shown as the blue square line. Top row, unstable inputs shown with a dashed line, stable with a solid line. Bottom row, objective and constraints which are active at plant optimum, where the standard MA approach is shown as the grey circle line.

# 6.3.5. MSMA: Selection Based on Closeness

The final MSMA approach using the closest solution is applied. Using an input filter gain of K = 0.5I, the solution is shown in Figure 16, where the RTO system successfully converges to the plant optimum. Compared to the minimum modifier method, this approach is far smoother, but breaks the constraints more, and takes longer to converge to the optimum. The basis for using the closest solution is to reduce the risk by taking shorter steps. This works against the method when the current operating point is infeasible as remaining close to the current operating point will also likely give infeasible conditions.



**Figure 16.** Multi-solution MA inputs and NLP functions using the closest solution approach shown as the green triangle line. Top row, inputs. Bottom row, objective and constraints which are active at plant optimum, where the standard MA approach is shown as the grey circle line.

# 6.4. Conclusions of MSMA Performance

Three case studies have been simulated giving an idea of the expected performance that MSMA scheme can provide. For the WO CSTR and the semi-batch reactor, the improvement to the model adequacy conditions are shown, where for the minimum modifier selection,

central tendencies using convex NLPs, and closest solution methods are all able to converge to the plant optimum.

Using any of the recommended approaches on these case studies improves the performance of the RTO trajectory by either providing stability, reducing infeasibility in the iterates before convergence, or by increasing the rate of improvement to the objective function. Using the minimum modifier overall performed well for the second and third case studies, where it is the (joint-) best scheme, however for the first case study it performs less well. Using the regional quadratics with the mean solution also performs well and does not require a set of uncertainty to be used. However, it is difficult to use on higher dimensional problems, and using the mean without convex NLPs performs poorly. The final recommended approach is using the closest solution, which is the most consistent when applied to the cases above. This comes at the cost of having to solve all the models in the set, which can be time consuming for more complex problems.

## 7. Conclusions

Modifier adaptation is an RTO approach which uses estimates of the first order properties of the plant to modify a nominal model such that it matches these estimates at the current operating point. During the design stage of an industrial process, this nominal model is generated, and due to simplifications, noise and disturbances, there is always some degree of plant-model mismatch. Typically, the parameters used in the model come with a degree of uncertainty to account for this discrepancy. The use of this known uncertainty to account for this mismatch is limited in RTO, where it is discarded in favor of real-time measurements.

A new framework is introduced known as multiple solution modifier adaptation (MSMA) which uses this known uncertainty, along with the real-time measurements to improve the choice of the next RTO iteration. The framework is broken down into two choices, firstly how to generate the set of potential solutions, and secondly how to combine these potential operating points into a single set of inputs to apply to the plant. Several different methods have been introduced for each of these options, which are applicable in a wide range of scenarios, even when the uncertainties in the model parameters are unknown.

Each of the methods proposed have been shown to maintain KKT matching upon convergence if either the selected operating point is one of the potential operating points, or within the convex hull of a set of points generated from convex NLPs. Three approaches are recommended from this framework. The improvement to the performance is analyzed through three different case studies, where the recommended schemes outperform standard MA in terms of rate of convergence and stability at the plant optimum.

These case studies have been simulated under an idealized scenario of no measurement noise, no disturbances, and no other errors (other than plant-model mismatch). Therefore, the conclusions of the performance of these approach need to be considered with this in mind, and further work is required to understand how the proposed schemes operate under more realistic assumptions.

**Author Contributions:** J.S.: Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Visualization, Writing—original draft, Writing—review & editing. G.F.: Conceptualization, Methodology, Supervision, Writing—review & editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by The Engineering and Physical Sciences Research Council grant number EP/R513209/1.

#### Data Availability Statement: Not applicable

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

# Abbreviations

The following abbreviations are used in this manuscript:

Real-time optimization
Karush–Kuhn–Tucker
Modifier adaptation
Non-linear program
Continuous stirred-tank reactor
Multiple solution modifier adaptation
Probability distribution function
Williams-Otto
Proportional-integral
Differential-algebraic equations
Model predictive control
Biomass
Nitrate
Phycocyanin

# Appendix A. Williams-Otto CSTR Simulation Details

This section gives further details to the simulation of the plant and model used in the Williams-Otto CSTR case study in Section 6.1.

# Appendix A.1. Plant

The plant is assumed to be ideal and follow a three reactions set-up, given by:

$$\begin{array}{l} A+B \xrightarrow{k_1} C,\\ B+C \xrightarrow{k_2} P+E,\\ C+P \xrightarrow{k_3} G, \end{array}$$

where  $k_j$  is reaction rate constant for reaction j. These reactions are assumed to be elementary and the rate constant is assumed to follow the Arrhenius equation, given by:

$$k_j = \nu_j e^{E_j / (T_R + 273.15)} \tag{A1}$$

where *T* is the temperature of the reactor in °C, and  $v_j$  and  $E_j$  are the pre-exponential factor and the activation energy of reaction *j*, respectively. The plant has three input variables, the mass flow rate of components A and B into the reactor, and the reactor temperature,  $u = [F_A, F_B, T_R]$ , with the total flow rate out of the reactor equal to the total flow rate into the reactor  $F = F_A + F_B$ , and all flow rates are in kg s<sup>-1</sup>. The steady-state mass balances of the components can be written as follows:

$$\begin{aligned} \frac{\partial X_A}{\partial t} &= \frac{F_A}{M} - k_1 X_A X_B - \frac{F}{M} X_A = 0, \\ \frac{\partial X_B}{\partial t} &= \frac{F_B}{M} - k_1 X_A X_B - k_2 X_C X_B - \frac{F}{M} X_B = 0, \\ \frac{\partial X_C}{\partial t} &= 2k_1 X_A X_B - 2k_2 X_C X_B - k_3 X_C X_P - \frac{F}{M} X_C = 0, \\ \frac{\partial X_P}{\partial t} &= k_2 X_C X_B - 0.5k_3 X_C X_P - \frac{F}{M} X_P = 0, \\ \frac{\partial X_E}{\partial t} &= 2k_2 X_C X_B - \frac{F}{M} X_E = 0, \\ \frac{\partial X_G}{\partial t} &= 1.5k_3 X_C X_P - \frac{F}{M} X_G = 0, \end{aligned}$$

where  $X_i$  is the mass fraction of component *i*, and M is the mass holdup for the reactor, with the plant parameters summarized in Table A1. The measured outputs of the system are the mass fractions of each of the components.

Table A1. Plant parameters and constants.

Parameter	Value	Unit	
$\nu_1$	$1.6599 \times 10^6$	$\mathrm{s}^{-1}$	
$\nu_2$	$7.2117  imes 10^8$	$\mathrm{s}^{-1}$	
$\nu_3$	$2.6745 \times 10^{12}$	$\mathrm{s}^{-1}$	
$E_1$	6666.7	K	
$E_2$	8333.3	K	
$E_3$	11,111	К	
М	2105	kg	

# Appendix A.2. Model

As realistic model parameter uncertainties are required, the 2-reaction model introduced in [8] is reparameterized using simulated experiments corresponding to how model parameters are determined in practice. The two reaction system is as follows:

$$A + 2B \xrightarrow{k_1^*} P + E,$$
$$A + B + P \xrightarrow{k_2^*} G,$$

To calculate appropriate parameters for the 2-reaction mechanism, a simulation of the batch experiments of the true reaction mechanism is carried out. Concentrations of the reactants and products are taken at regular time intervals. The parameters of the 2-reaction process can be adjusted to best fit these concentration-time curves. Running the experiment at several different initial concentrations and temperatures, a set of parameters can be obtained. This set can be used to determine approximate values for the parameters and associated uncertainty bounds. The concentration time curves are fit using the dynamic equations of the reaction mechanism. The reactions are assumed to be elementary based on the stoichiometry of the reactants, and the rate constants are assumed to follow the Arrhenius equations, as with the simulation of the plant.

Running the experiment at 6 different temperatures from 80  $^{\circ}$ C to 105  $^{\circ}$ C, and at 9 different initial concentrations ranging from 0.3 to 0.7 mass fraction of A. Each run consisted of 24 samples at different time steps of the reaction. Noise was introduced in the time of measurement and the concentration calculation.

The results of the reparameterization are shown in Table A2, along with the associated error in the parameters. Additionally, there is a strong positive covariance between the activation energies, with a correlation between them of 0.85.

**Table A2.** New parameters of the model based on experimental data from the plant with the associated standard deviations.

Parameter	Mean Value	Standard Deviation	Unit
$\nu_1^*$	$1.06  imes 10^7$	$0.055  imes 10^7$	$s^{-1}$
$\nu_2^*$	$3.13 imes10^{11}$	$0.15 imes10^{11}$	$s^{-1}$
$\bar{E_1^*}$	7034	70	Κ
$E_2^{\overline{*}}$	10,680	160	K

# **Appendix B. Semi-Batch Bioreactor Simulation Details**

# Appendix B.1. Plant

The reactor is assumed to have a fixed volume, and the plant is simulated based on Monod kinetics, and using Aiba model [46], which captures the effects of light intensity on the growth of microalgae and bioproduction. The state equations for the concentrations of the three components are given as follows:

$$\frac{\partial C_X(t)}{\partial t} = u_m \frac{I(t)}{I(t) + k_s + \frac{I(t)^2}{k_i}} C_X(t) \frac{C_N(t)}{C_N(t) + K_N} - u_d C_X(t),$$
(A2)

$$\frac{\partial C_N(t)}{\partial t} = -Y_{N/X} u_m \frac{I(t)}{I(t) + k_s + \frac{I(t)^2}{k_i}} C_X(t) \frac{C_N(t)}{C_N(t) + K_N} + F_N(t),$$
(A3)

$$\frac{\partial C_P(t)}{\partial t} = k_m \frac{I(t)}{I(t) + k_{sq} + I(t)^2 / k_{iq}} C_X(t) - k_d \frac{C_P(t)}{C_N(t) + K_{Np}}.$$
 (A4)

The terms what have not yet been defined are constant parameters which are defined in Table A3 and the optimum of the plant occurs at the conditions given in Table A4.

Table A3. Parameter values for plant
--------------------------------------

Constant	Value	Units
<i>u</i> <sub>m</sub>	0.0572	$h^{-1}$
<i>u</i> <sub>d</sub>	0.0	$h^{-1}$
$K_N$	393.1	$mgL^{-1}$
$Y_{N/X}$	504.5	$mgg^{-1}$
$k_m$	0.00016	$mgg^{-1}h^{-1}$
k <sub>d</sub>	0.281	$h^{-1}$
$k_s$	178.9	$\mu$ mol m <sup>-2</sup> s <sup>-1</sup>
k <sub>i</sub>	447.1	$\mu mol m^{-2} s^{-1}$
$k_{sq}$	23.51	$\mu$ mol m <sup>-2</sup> s <sup>-1</sup>
k <sub>iq</sub>	800	$\mu mol m^{-2} s^{-1}$
K <sub>Np</sub>	16.89	$mgL^{-1}$

Table A4. Plant optimum inputs.

Input	I(t)	$F_N(t)$
$0 \ge t < 40$	282	24.9
$40 \ge t < 80$	281	17.8
$80 \ge t < 120$	277	40.0
$120 \ge t < 160$	146	40.0
$160 \ge t < 200$	120	40.0
$200 \ge t \ge 240$	174	40.0

# Appendix B.2. Model

The model is assumed to follow a simplified set of state equations, where part of the light intensity term is neglected, leading to structural mismatch between the plant and model. The state equations for the model are given as follows:

$$\frac{\partial C_X(t)}{\partial t} = u_m \, \frac{I(t)}{I(t) + k_s} \, C_X(t) \, \frac{C_N(t)}{C_N(t) + K_N} - u_d \, C_X(t), \tag{A5}$$

$$\frac{\partial C_N(t)}{\partial t} = -Y_{N/X} u_m \frac{I(t)}{I(t) + k_s} C_X(t) \frac{C_N(t)}{C_N(t) + K_N} + F_N(t), \tag{A6}$$

$$\frac{\partial C_P(t)}{\partial t} = k_m \frac{I(t)}{I(t) + k_{sq}} C_X(t) - k_d \frac{C_P(t)}{C_N(t) + K_{Np}}.$$
(A7)

In addition to the structural mismatch, there is an error in the parameter  $Y_{N/X}$ , which is 20% higher than the true value used to simulate the plant. The pool of models is developed through the uncertainty in three of the parameters, which are given in Table A5. The uncertainty in  $u_m$  and  $k_s$  is approximately 5%, whereas for  $Y_{N/X}$  it is close to 20% of the true value. The other parameters match those used to simulate the plant.

Table A5. Parameter uncertainty for model	Table A5.	Parameter	uncertainty	for mode	1
---	-----------	-----------	-------------	----------	---

Constant	Mean Value	Uncertainty	Units
$u_m$ $Y_{N/X}$	0.0572 605.4 178.9	$\pm 0.003$ $\pm 100$	$h^{-1}$ mg g <sup>-1</sup> umal m <sup>-2</sup> s <sup>-1</sup>
$\kappa_s$	178.9	$\pm 9$	µmolm <sup>2</sup> s <sup>1</sup>

The optimal inputs for the nominal model (defined by the mean parameters) produce a control sequence which is infeasible for the plant, therefore they are modified such that the initial set of inputs for the RTO scheme is feasible for the plant. Table A6 shows the initial set of inputs for each of the RTO schemes.

Table A6	. Initial	RTO	inputs
----------	-----------	-----	--------

Input	I(t)	$F_N(t)$
$0 \ge t < 40$	320	15
$40 \ge t < 80$	320	20
$80 \ge t < 120$	320	35
$120 \ge t < 160$	160	35
$160 \ge t < 200$	120	35
$200 \ge t \ge 240$	150	35

#### References

- 1. Ariyur, K.B.; Krstic, M. Real Time Optimization by Extremum Seeking Control; Wiley: New York, NY, USA, 2003.
- Skogestad, S. Self-optimizing control: The missing link between steady-state optimization and control. *Comput. Chem. Eng.* 2000, 24, 569–575. [CrossRef]
- François, G.; Srinivasan, B.; Bonvin, D. Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. J. Process Control 2005, 15, 701–712. [CrossRef]
- Jang, S.S.; Joseph, B.; Mukai, H. On-line optimization of constrained multivariable chemical processes. *AIChE J.* 1987, 33, 26–35. [CrossRef]
- 5. Quelhas, A.D.; de Jesus, N.J.C.; Pinto, J.C. Common vulnerabilities of RTO implementations in real chemical processes. *Can. J. Chem. Eng.* **2013**, *91*, 652–668. [CrossRef]
- 6. Câmara, M.M.; Quelhas, A.D.; Pinto, J.C. Performance evaluation of real industrial RTO systems. Processes 2016, 4, 44. [CrossRef]
- Chen, C.Y.; Joseph, B. On-Line Optimization Using a Two-Phase Approach: An Application Study. Ind. Eng. Chem. Res. 1987, 26, 1924–1930. [CrossRef]
- 8. Roberts, P.D. An algorithm for steady-state system optimization and parameter estimation. *Int. J. Syst. Sci.* **1979**, *10*, 719–734. [CrossRef]
- 9. Tatjewski, P. Iterative optimizing set-point control The basic principle redesigned. *IFAC Proc. Vol. (IFAC-PapersOnline)* 2002, 15, 49–54. [CrossRef]
- 10. Gao, W.; Engell, S. Iterative set-point optimization of batch chromatography. Comput. Chem. Eng. 2005, 29, 1401–1409. [CrossRef]
- 11. Marchetti, A.; Chachuat, B.; Bonvin, D. Modifier-adaptation methodology for real-time optimization. *Ind. Eng. Chem. Res.* 2009, 48, 6022–6033. [CrossRef]
- 12. Marchetti, A.; François, G.; Faulwasser, T.; Bonvin, D. Modifier Adaptation for Real-Time Optimization—Methods and Applications. *Processes* **2016**, *4*, 55. [CrossRef]
- 13. Marchetti, A.; Chachuat, B.; Bonvin, D. A dual modifier-adaptation approach for real-time optimization. *J. Process Control* **2010**, 20, 1027–1037. [CrossRef]
- 14. Brdyś, M.; Tatjewski, P. An Algorithm for Steady-State Optimizing Dual Control of Uncertain Plants. *IFAC Proc. Vol.* **1994**, 27, 215–220. [CrossRef]
- 15. Navia, D.; Briceño, L.; Gutiérrez, G.; De Prada, C. Modifier-Adaptation Methodology for Real-Time Optimization Reformulated as a Nested Optimization Problem. *Ind. Eng. Chem. Res.* **2015**, *54*, 12054–12071. [CrossRef]

- Costello, S.; François, G.; Bonvin, D. A Directional Modifier-Adaptation Algorithm for Real-Time Optimization. *J. Process Control* 2016, 39, 64–76. [CrossRef]
- 17. Gros, S. An analysis of the directional-modifier adaptation algorithm based on optimal experimental design. *Processes* **2017**, *5*, 1. [CrossRef]
- Singhal, M.; Marchetti, A.G.; Faulwasser, T.; Bonvin, D. Active directional modifier adaptation for real-time optimization. *Comput. Chem. Eng.* 2018, 115, 246–261. [CrossRef]
- 19. Gao, W.; Wenzel, S.; Engell, S. A reliable modifier-adaptation strategy for real-time optimization. *Comput. Chem. Eng.* 2016, 91, 318–328. [CrossRef]
- 20. Gao, W.; Hernández, R.; Engell, S. A Study of Explorative Moves during Modifier Adaptation with Quadratic Approximation. *Processes* **2016**, *4*, 45. [CrossRef]
- François, G.; Bonvin, D. Use of transient measurements for the optimization of steady-state performance via modifier adaptation. *Ind. Eng. Chem. Res.* 2014, 53, 5148–5159. [CrossRef]
- 22. Rodríguez-Blanco, T.; Sarabia, D.; Pitarch, J.; de Prada, C. Modifier Adaptation methodology based on transient and static measurements for RTO to cope with structural uncertainty. *Comput. Chem. Eng.* **2017**, *106*, 480–500. [CrossRef]
- Speakman, J.; François, G. Real-time optimization via modifier adaptation of closed-loop processes using transient measurements. Comput. Chem. Eng. 2020, 140, 106969. [CrossRef]
- 24. Wenzel, S.; Gao, W.; Engell, S. Handling disturbances in modifier adaptation with quadratic approximation. *IFAC-PapersOnLine* **2015**, *28*, 132–137. [CrossRef]
- Navia, D.; Puen, A.; Quintanilla, P.; Briceño, L.; Bergh, L. On dealing with measured disturbances in the modifier adaptation method for real-time optimization. *Comput. Chem. Eng.* 2019, 128, 141–163. [CrossRef]
- Ferreira, T.D.A.; Shukla, H.A.; Faulwasser, T.; Jones, C.N.; Bonvin, D. Real-Time Optimization of Uncertain Process Systems using Machine Learning via Gaussian Processes. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; pp. 465–470.
- 27. del Rio Chanona, E.A.; Petsagkourakis, P.; Bradford, E.; Graciano, J.E.; Chachuat, B. Real-time optimization meets Bayesian optimization and derivative-free optimization: A tale of modifier adaptation. *Comput. Chem. Eng.* **2021**, *147*, 107249. [CrossRef]
- 28. Speakman, J.; Papasavvas, A.; François, G. A robust modifier adaptation method via Hessian augmentation using model uncertainties. *J. Process Control* **2021**, *99*, 28–40. [CrossRef]
- 29. Speakman, J.; François, G. Robust Modifier Adaptation via Worst-Case and Probabilistic Approaches. *Ind. Eng. Chem. Res.* 2021, 61, 515–529. [CrossRef]
- 30. Bunin, G.A.; François, G.; Bonvin, D. Sufficient Conditions for Feasibility and Optimality of Real-Time Optimization Schemes—II. Implementation Issues. *arXiv* **2013**, arXiv:1308.2625.
- Ruppen, D.; Benthack, C.; Bonvin, D. Optimization of batch reactor operation under parametric uncertainty—Computational aspects. J. Process Control 1995, 5, 235–240. [CrossRef]
- 32. Bernardo, F.P.; Saraiva, P.M. Robust optimization framework for process parameter and tolerance design. *AIChE J.* **1998**, 44, 2007–2017. [CrossRef]
- 33. Mönnigmann, M.; Marquardt, W. Steady-state process optimization with guaranteed robust stability and flexibility: Application to HDA reaction section. *Ind. Eng. Chem. Res.* 2005, 44, 2737–2753. [CrossRef]
- 34. Bunin, G.A.; François, G.; Bonvin, D. Sufficient Conditions for Feasibility and Optimality of Real-Time Optimization Schemes—I. Theoretical Foundations. *arXiv* **2013**, arXiv:1308.2620.
- 35. François, G.; Bonvin, D. Use of convex model approximations for real-time optimization via modifier adaptation. *Ind. Eng. Chem. Res.* **2013**, *52*, 11614–11625. [CrossRef]
- Matias, J.O.A.; Jäschke, J. Online model maintenance via output modifier adaptation. Ind. Eng. Chem. Res. 2019, 58, 13750–13766. [CrossRef]
- Matias, J.; Kungurtsev, V.; Egan, M. Simultaneous online model identification and production optimization using modifier adaptation. J. Process Control 2022, 110, 110–120. [CrossRef]
- Martinez Villegas, R.; Budman, H.; Elkamel, A. Identification of Dynamic Metabolic Flux Balance Models Based on Parametric Sensitivity Analysis. *Ind. Eng. Chem. Res.* 2017, 56, 1911–1919. [CrossRef]
- Hille, R.; Budman, H.M. Simultaneous identification and optimization of biochemical processes under model-plant mismatch using output uncertainty bounds. *Comput. Chem. Eng.* 2018, 113, 125–138. [CrossRef]
- 40. Williams, T.J.; Otto, R.E. A generalized chemical processing model for the investigation of computer control. *Trans. Am. Inst. Electr. Eng. Part I Commun. Electron.* **1960**, *79*, 458–473. [CrossRef]
- Faulwasser, T.; Bonvin, D. On the use of second-order modifiers for real-time optimization. *IFAC Proc. Vol. (IFAC-PapersOnline)* 2014, 19, 7622–7628. [CrossRef]
- Diehl, M.; Uslu, I.; Findeisen, R.; Schwarzkopf, S.; Allgöwer, F.; Bock, H.G.; Bürner, T.; Gilles, E.D.; Kienle, A.; Schlöder, J.P.; et al. Real-Time Optimization for Large Scale Processes: Nonlinear Model Predictive Control of a High Purity Distillation Column. In Online Optimization of Large Scale Systems; Springer: Berlin/Heidelberg, Germany, 2001; pp. 363–383.
- 43. Del Rio-Chanona, E.A.; Dechatiwongse, P.; Zhang, D.; Maitland, G.C.; Hellgardt, K.; Arellano-Garcia, H.; Vassiliadis, V.S. Optimal Operation Strategy for Biohydrogen Production. *Ind. Eng. Chem. Res.* **2015**, *54*, 6334–6343. [CrossRef]

- 44. Del Rio Chanona, E.A.; Alves Graciano, J.E.; Bradford, E.; Chachuat, B. Modifier-adaptation schemes employing Gaussian processes and trust regions for real-time optimization. *IFAC-PapersOnLine* **2019**, *52*, 52–57. [CrossRef]
- 45. Bradford, E.; Imsland, L.; Zhang, D.; del Rio Chanona, E.A. Stochastic data-driven model predictive control using gaussian processes. *Comput. & Chem. Eng.* 2020, 139, 106844.
- 46. Aiba, S. Growth Kinetics of Photosynthetic Microorganisms; Springer: Berlin/Heidelberg, Germany, 1982; pp. 85–156.