2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Active Directional Modifier Adaptation with Trust Region – Application to Energy-Harvesting Kites *

Matthieu Jonin[1], Martand Singhal[1], Sanket Diwale[1], Colin N. Jones[1] and Dominique Bonvin[1]

Abstract— **Many real-time optimization schemes maximize process performance by performing a model-based optimization. However, due to plant-model mismatch, the model-based solution is often suboptimal. In modifier adaptation, measurements are used to correct the model in such a way that the first-order necessary conditions of optimality are satisfied *for the plant*. However, performing experiments to obtain measurements can be costly. This paper uses a sensitivity analysis that allows making only partial corrections to the model, thereby relying on fewer experiments. Furthermore, this sensitivity analysis is of global nature, which ensures that the corrections are sufficient in the presence of large parametric uncertainties. However, since the corrections are still only locally valid, this paper proposes to control the update step length via a trust-region technique. The resulting algorithm is illustrated via the simulation of an energy-harvesting kite.**

I. INTRODUCTION

Model-based optimization enables process plants to attain optimal performance, while meeting constraints on product quality, environmental regulations and safety. Since process models are mere approximations of the actual plant, they need to be corrected based on process data. The corrections are done to ensure that the plant behavior is adequately captured by the model. This paper presents an approach that builds on the existing modifier-adaptation (MA) methods [1]–[4] that correct the model cost and constraint functions to ensure that the resulting model-optimal solution satisfies the plant Karush-Kuhn-Tucker (KKT) conditions upon convergence.

This paper deals with two issues that are important in MA implementation:

1) The estimation of plant gradients for the cost and constraints. These estimates are obtained via appropriate (but costly) experiments. The experimental burden increases linearly with the number of process inputs.
2) The control of step length for input updates. Since the model corrections are only valid around the current operating point and, in addition, no Hessian corrections are made, control of the input step length is essential to progress within the locally valid region. Traditionally, a filter is applied to reduce the step length. The use of filters may be difficult to implement, in particular when the models deployed are non-convex. Too large of a gain may result in over-correction, while a low gain may cause the algorithm to progress too slowly.

To reduce the experimental burden, gradient estimates can be obtained only in a few *privileged* directions of the input space. The set of privileged directions are obtained via model-based sensitivity analysis. Directional Modifier adaptation (DMA) based on local sensitivity analysis was shown to perform well for the real-time power maximization of kites [5]. However, local sensitivities are highly dependent on model accuracy. In contrast, a more robust methodology based on global sensitivity analysis (GSA), labeled Active Directional Modifier Adaptation (ADMA), has shown to overcome the sub-optimality of DMA in the presence of large parametric uncertainty [2], [3].

To control the step length in MA, researchers have used the concept of a trust region [4], [6]. Along the same line, this paper presents an ADMA scheme that considers a trust region for the input moves. Applied to a simulated kite system, this algorithm converges to near plant optimality despite plant-model mismatch and the presence of measurement noise.

The paper is structured as follows. Section II formulates the problem and introduces both the MA schemes for performing real-time optimization and the global sensitivity analysis used in ADMA. Section III presents the Trust Region - Active Directional Modifier Adaptation (TR-ADMA) scheme. Section IV discusses the application of TR-ADMA to the kite problem, while Section V concludes the paper.

II. PRELIMINARIES

A. Problem formulation

The optimization of process operation consists in minimizing operating costs, or maximizing economic profit, in the presence of constraints. Mathematically, this problem can be formulated as follows:

$$\begin{aligned}\min_{\mathbf{u}} \quad & \Phi_p(\mathbf{u}) := \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \\ \text{s.t.} \quad & G_{p,i}(\mathbf{u}) := g_i(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \leq 0, \quad i = 1, \dots, n_g,\end{aligned} \tag{II.1}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ are the decision (or input) variables; $\mathbf{y}_p \in \mathbb{R}^{n_y}$ are the measured output variables; $\phi : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$ is the cost function to be minimized; $g_i : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$, $i = 1, \dots, n_g$, is the set of process-dependent inequality constraint functions. The subscript p refers to a plant quantity. These plant quantities are a priori unknown, but they can be measured.

*This work was supported by Swiss National Science Foundation under the project 149258 and the project 'Autonomous Airborne Wind Energy (A2WE)' 141386.

[1] with Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland {matthieu.jonin, martand.singhal, sanket.diwale, colin.jones, dominique.bonvin}@epfl.ch

A mathematical model is available, thus leading to the following model-based optimization problem:

$$\begin{aligned} \min_{\mathbf{u}} \quad & \Phi(\mathbf{u},\boldsymbol{\theta}) := \phi(\mathbf{u},\mathbf{y}(\mathbf{u},\boldsymbol{\theta})) \\ \text{s.t.} \quad & G_i(\mathbf{u},\boldsymbol{\theta}) := g_i(\mathbf{u},\mathbf{y}(\mathbf{u},\boldsymbol{\theta})) \le 0, \quad i=1,\dots,n_g, \end{aligned} \tag{II.2}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is the vector of model parameters.

B. Modifier adaptation

Modifier adaptation is an iterative RTO scheme that solves Problem (II.2) with the following modified cost and constraint functions to reach *plant* optimality:

$$\min_{\mathbf{u}} \ \Phi_{m,k}(\mathbf{u}) := \Phi(\mathbf{u},\boldsymbol{\theta}) + \varepsilon_k^{\Phi} + (\boldsymbol{\lambda}_k^{\Phi})^{\mathsf{T}}(\mathbf{u}-\mathbf{u}_k) \tag{II.3a}$$

$$\begin{aligned} \text{s.t. } G_{m,i,k}(\mathbf{u}) &:= G_i(\mathbf{u},\boldsymbol{\theta}) + \varepsilon_k^{G_i} + (\boldsymbol{\lambda}_k^{G_i})^{\mathsf{T}}(\mathbf{u}-\mathbf{u}_k) \le 0, \\ & i=1,\dots,n_g, \end{aligned} \tag{II.3b}$$

with the modifiers $\varepsilon_k^{\Phi} \in \mathbb{R}$, $\varepsilon_k^{G_i} \in \mathbb{R}$, $\boldsymbol{\lambda}_k^{\Phi} \in \mathbb{R}^{n_u}$, and $\boldsymbol{\lambda}_k^{G_i} \in \mathbb{R}^{n_u}$. The modifiers are typically computed as follows:

$$\varepsilon_k^{\Phi} := \Phi_p(\mathbf{u}_k) - \Phi(\mathbf{u}_k,\boldsymbol{\theta}), \tag{II.4a}$$

$$\varepsilon_k^{G_i} := G_{p,i}(\mathbf{u}_k) - G_i(\mathbf{u}_k,\boldsymbol{\theta}), \quad i=1,\dots,n_g, \tag{II.4b}$$

$$(\boldsymbol{\lambda}_k^{\Phi_i})^{\mathsf{T}} := \nabla\Phi_p(\mathbf{u}_k) - \nabla\Phi(\mathbf{u}_k,\boldsymbol{\theta}), \tag{II.4c}$$

$$(\boldsymbol{\lambda}_k^{G_i})^{\mathsf{T}} := \nabla G_{p,i}(\mathbf{u}_k) - \nabla G_i(\mathbf{u}_k,\boldsymbol{\theta}), \ i=1,\dots,n_g, \tag{II.4d}$$

where the plant quantities (subscript p) are estimated from the output of the system $\boldsymbol{y}_p$ and the model quantities are calculated through the simulation. Note that since ε^{Φ} does not influence the optimum of Problem (II.3), it is often omitted. However, the modified model and the plant quantities will not be equal locally if this modifier is missing. Because this local equality matching is needed for the trust-region algorithm to converge [4], this modifier is incorporated here.

C. *Sensitivity-based* MA *schemes*

The modifiers introduced in MA schemes compensate for the model's inadequacy to accurately predict the cost and constraint values and their respective gradients. Since computing gradients is experimentally expensive, it becomes necessary to rely on model gradients along certain input directions. By computing the gradient sensitivities with respect to model parameters, the input space can be divided into two sets of directions, one set in which the directions are such that the model gradients are robust to parametric uncertainties, and the second set of directions along which the model gradients are sensitive to parametric perturbations. Then, one will correct the model gradients only along the second set of directions called *privileged directions*. DMA uses the local sensitivity of the model Lagrangian gradient to compute the set of privileged directions. However, local sensitivities are unable to capture the global behavior, especially when the Lagrangian is a nonlinear function of the parameters. To find a globally robust set of privileged directions, ADMA performs a global sensitivity analysis over a given set of parametric uncertainty. Instead of computing sensitivity only at the nominal value of model parameters, ADMA computes the gradient sensitivities for various parameter values obtained via Monte-Carlo sampling. The sensitivity matrix of the i^{th} parameter sample at the k^{th} iteration is expressed as follows:

$$\boldsymbol{B}_k^{(i)} := \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u} \partial \boldsymbol{\theta}}(\mathbf{u}_k, \boldsymbol{\mu}_k, \boldsymbol{\theta}_i), \quad i=1,\dots,N, \tag{II.5}$$

where $\mathcal{L}(\mathbf{u},\boldsymbol{\mu},\boldsymbol{\theta}) := \Phi(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{G}(\mathbf{u},\boldsymbol{\theta})$ is the model Lagrangian, $\boldsymbol{\theta}_i$ is the i^{th} Monte-Carlo sample, and N is the total number of samples. Note that $\boldsymbol{G}$ is the vector of constraints G_i and $\boldsymbol{\mu}_k$ is the vector of Lagrange multipliers obtained by solving the optimization problem at the k^{th} iteration of ADMA. All computed sensitivities are collected into the single matrix $\hat{\boldsymbol{B}}_k \in \mathbb{R}^{n_u \times n_\theta N}$,

$$\hat{\boldsymbol{B}}_k := \frac{1}{\sqrt{N}}[\boldsymbol{B}_k^{(1)}\, \boldsymbol{B}_k^{(2)} \cdots \boldsymbol{B}_k^{(N)}]\,. \tag{II.6}$$

Then, based on the singular-value decomposition of $\hat{\boldsymbol{B}}_k$, one selects the $n_r \le n_u$ most sensitive directions, that is, the directions that are associated with the n_r largest singular values [2]:

$$\hat{\boldsymbol{B}}_k = \boldsymbol{W}_k \boldsymbol{S}_k \boldsymbol{V}_k^{\mathsf{T}}, \quad \boldsymbol{S}_k \boldsymbol{S}_k^{\mathsf{T}} = \operatorname{diag}(\sigma_{1,k},\dots,\sigma_{n_u,k}) \tag{II.7}$$

$$\boldsymbol{W}_k = [\boldsymbol{W}_{1,k}\ \boldsymbol{W}_{2,k}], \quad \boldsymbol{W}_{1,k} \in \mathbb{R}^{n_u \times n_r}, \tag{II.8}$$

$$n_r = \min\{i, n_u\} : \sigma_{i+1,k} \ll \sigma_{i,k}, \tag{II.9}$$

where $\sigma_{1,k} \ge \cdots \ge \sigma_{n_u,k} \ge 0$. The columns of $\boldsymbol{W}_{1,k} \in \mathbb{R}^{n_u \times n_r}$ represent the set of privileged directions. Plant gradients are then estimated only in the directions given by the columns of $\boldsymbol{W}_{1,k}$. Note that the number n_r can be set as a fixed number for every iteration k based on the experimental cost that one is ready to accept. However, this may result in larger optimality loss if some of the non-estimated gradients are fairly inaccurate.

The plant cost and constraint directional derivatives along $\boldsymbol{W}_{1,k}$ are denoted bx $\nabla_{W_{1,k}}\Phi_p$ and $\nabla_{W_{1,k}}G_{p,i}$, respectively. The estimated plant gradients are computed as follows:

$$\begin{aligned} \widehat{\nabla\Xi}_k =& \nabla\Xi(\mathbf{u}_k,\boldsymbol{\theta})(\boldsymbol{I}_{n_u} - \boldsymbol{W}_{1,k}\boldsymbol{W}_{1,k}^{+}) + \nabla_{W_{1,k}}\Xi_p(\mathbf{u}_k)\boldsymbol{W}_{1,k}^{+}, \\ & \Xi \in \{\Phi,\, G_i\} \end{aligned} \tag{II.10}$$

where $(\cdot)^{+}$ is a Moore-Penrose pseudo inverse and $\boldsymbol{I}$ denotes the identity matrix. In ADMA, instead of computing the first-order modifiers (II.4c)-(II.4d) with the plant gradients $\nabla\Phi_p$ and $\nabla G_{p,i}$, this is done using the estimates $\widehat{\nabla\Phi}_k$ and $\widehat{\nabla G}_{i,k}$.

III. TRUST REGION IN MODIFIER ADAPTATION

A. Trust region to control the step length

The inputs $\mathbf{u}_{k+1}$ obtained by solving the modified optimization problem (II.3) at the k^{th} iteration may be far away from the previous inputs $\mathbf{u}_k$. As the corrections applied to the cost and constraint functions are only locally valid around the operating point determined at the k^{th} iteration, the application of $\mathbf{u}_{k+1}$ to the plant may result in poor performance. To limit the over-reliance on the model, a trust region around the current inputs is defined, wherein the modified model is a fairly accurate representation of the plant. The modified optimization problem (II.3) is then solved inside the trust region so as to iteratively improve the plant performance. The size of the trust region is critical and is determined based on the predicted (model-based) and actual (plant-based) performance. The use of a trust region in MA was shown to result in global convergence [4]. Given the benefits of the trust-region concept, we propose to include a trust region within ADMA, thereby resulting in the TR-ADMA algorithm described as Algorithm 1.

B. Trust-region adjustment in TR-ADMA

The key element in any trust-region-based algorithm is the choice of its size $\Delta_k \in \mathbb{R}^{n_u}$ at each iteration. This choice is based on the agreement between the performance predicted by the model and the actual plant performance. To this end, the following ratio between the plant Lagrangian and the Lagrangian of the modified model is introduced:

$$\rho_k = \frac{\mathcal{L}_p(\mathbf{u}_k, \boldsymbol{\mu}_k) - \mathcal{L}_p(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})}{\mathcal{L}_{m,k}(\mathbf{u}_k, \boldsymbol{\mu}_k) - \mathcal{L}_{m,k}(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})}, \tag{III.4}$$

where $\mathcal{L}_p(\mathbf{u}, \boldsymbol{\mu}) := \Phi_p(\mathbf{u}) + \boldsymbol{\mu}^\mathsf{T}\boldsymbol{G}_p(\mathbf{u})$ is the plant Lagrangian and $\mathcal{L}_{m,k}(\mathbf{u}, \boldsymbol{\mu}) := \Phi_{m,k}(\mathbf{u}) + \boldsymbol{\mu}^\mathsf{T}\boldsymbol{G}_{m,k}(\mathbf{u})$ is the modified model Lagrangian, with $\boldsymbol{G}_p$ denoting the vector of plant constraints $G_{p,i}$ and $\boldsymbol{G}_{m,k}$ denoting the vector of model constraints $G_{m,i,k}$. Based on the value of the ration ρ_k, it is decided whether to enlarge the trust region by a factor $\eta_1 > 0$, keep it constant or decrease the trust region by a factor of $\eta_2 > 0$.

At k^{th} iteration of Algorithm 1, the solution to Problem (III.1) is $(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})$. The ratio ρ_k is evaluated at $(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})$. Decision regarding accepting the solution $(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})$ is also taken based on the value of the ratio ρ_k. To this end, the scalar parameters η, γ_1 and γ_2 are defined with $0 \leq \eta \leq \gamma_1 < \gamma_2 < 1$. Then, the trust region is adapted as follows:

- $\rho_k < 0$ implies that the modified model predicts a decrease in Lagrangian value, while the plant Lagrangian value actually increases, or vice-versa. The model gives a wrong prediction as it is not sufficiently accurate in the prevailing trust region. Hence, the input vector $\mathbf{u}^\star_{k+1}$ is rejected, and the trust region is reduced in order to find a region in which the corrected model can be trusted.
- For $0 \leq \rho_k < \eta$, the modified model predicts a large change, while the plant differs only a little from the previous iteration, thus indicating a large disagreement between the two. Hence, the trust region is decreased.
- For $\eta \leq \rho_k \leq \gamma_1$, the changes in Lagrangians are sufficiently similar to accept the new input vector $\mathbf{u}^\star_{k+1}$, but the trust region is still decreased as the model prediction is not sufficiently close to the actual performance.
- For $\gamma_1 \leq \rho_k < \gamma_2$, the prediction is good as it mostly agrees with the actual performance. Hence, the input vector $\mathbf{u}^\star_{k+1}$ is accepted and the trust region Δ_k is kept at the same value for the next iteration.
- For $\gamma_2 \leq \rho_k$, the agreement between prediction and actual performance is excellent. The input vector $\mathbf{u}^\star_{k+1}$ is accepted and the trust region is enlarged.

Algorithm 1 TR-ADMA

Choose the eigenvalues of the (typically diagonal) filter matrices $\boldsymbol{K}^{\varepsilon^\Phi}$, $\boldsymbol{K}^{\lambda^\Phi}$, $\boldsymbol{K}^{\varepsilon^{G_i}}$ and $\boldsymbol{K}^{\lambda^{G_i}}$ in the interval]0,1]. Initialize the input vector $\mathbf{u}_0$. Set the modifiers $\varepsilon^\Phi_0 = 0$, $\boldsymbol{\lambda}^\Phi_0 = \mathbf{0}$, $\varepsilon^{G_i}_0 = 0$ and $\boldsymbol{\lambda}^{G_i}_0 = \mathbf{0}$. Choose the maximal allowable step size Δ^{max}, the parameters $0 < \eta_1 < 1$, $\eta_2 > 1$, the updating range parameters $0 \leq \eta \leq \gamma_1 < \gamma_2 < 1$ and set Δ_0 to an arbitrary large value.

```
for $k = 0$ to $\infty$ do
  Step 1. Compute the optimal inputs and Lagrange multipliers of the modified model-based problem:
      $(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1}) = \underset{\mathbf{u}}{\text{argmin}}\ \Phi_{m,k}(\mathbf{u})$   (III.1a)
      s.t. $G_{m,i,k}(\mathbf{u}) \leq 0,\ i = 1, \ldots, n_g$   (III.1b)
      $\mathbf{u}_k - \Delta_k \leq \mathbf{u} \leq \mathbf{u}_k + \Delta_k.$   (III.1c)
  Step 2. Apply $\mathbf{u}^\star_{k+1}$ to the plant and measure the noisy cost $\widetilde{\phi}_p(\mathbf{u}^\star_{k+1})$ and noisy constraints $\widetilde{G}_p(\mathbf{u}^\star_{k+1})$. Compute an estimate of the plant Lagrangian $\widetilde{\mathcal{L}}_p(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1}) := \widetilde{\Phi}_p(\mathbf{u}^\star_{k+1}) + \boldsymbol{\mu}^\mathsf{T}_{k+1}\ \widetilde{G}_p(\mathbf{u}^\star_{k+1})$.
  Step 3. Compute the ratio between the real and predicted Lagrangian decreases (trust-region step) and update the step size:
  if $k = 0$ then
    $\rho_k = 1$
  else
      $\rho_k = \frac{\widetilde{\mathcal{L}}_p(\mathbf{u}_k, \boldsymbol{\mu}_k) - \widetilde{\mathcal{L}}_p(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})}{\mathcal{L}_{m,k}(\mathbf{u}_k, \boldsymbol{\mu}_k) - \mathcal{L}_{m,k}(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})}$   (III.2)
  end if
  if $\rho_k > \gamma_2$ then
    $\Delta_{k+1} = min(\ \eta_2 \cdot \Delta_k, \Delta^{max}\ )$
  else if $\rho_k < \gamma_1$ then
    $\Delta_{k+1} = \eta_1 \cdot \Delta_k$
  else
    $\Delta_{k+1} = \Delta_k$
  end if
  Step 4. Update the operating point based on the value of the ratio $\rho_k$
  if $\rho_k \geq \eta$ then
    Accept the new operating point $(\mathbf{u}_{k+1}, \boldsymbol{\mu}_{k+1}) := (\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})$.
    Step 5. Find the privileged directions matrix $\boldsymbol{W}_{1,k+1}$ by performing SVD on the Monte-Carlo samples of the sensitivity of the Lagrangian, see Section (II-C).
    Step 6. Estimate the directional derivatives of the plant cost $\nabla_{W_{1,k+1}}\Phi_p(\mathbf{u}_{k+1})$ and constraints $\nabla_{W_{1,k+1}}G_p(\mathbf{u}_{k+1})$, for example, via finite differences. Compute the full gradient estimate using (II.10).
    Step 7. Update the modifiers:
      $\varepsilon^\Phi_{k+1} := (\boldsymbol{I}_{n_g} - \boldsymbol{K}^{\varepsilon^\Phi})\varepsilon^\Phi_k + \boldsymbol{K}^{\varepsilon^\Phi}(\widetilde{\Phi}_p(\mathbf{u}_{k+1}) - \Phi(\mathbf{u}_{k+1}, \boldsymbol{\theta}))$
      $\varepsilon^{G_i}_{k+1} := (\boldsymbol{I}_{n_g} - \boldsymbol{K}^{\varepsilon^{G_i}})\varepsilon^{G_i}_k + \boldsymbol{K}^{\varepsilon^{G_i}}(\widetilde{G}_p(\mathbf{u}_{k+1}) - G(\mathbf{u}_{k+1}, \boldsymbol{\theta}))$
      $\boldsymbol{\lambda}^\Phi_{k+1} := (\boldsymbol{I}_{n_u} - \boldsymbol{K}^{\lambda^\Phi})\boldsymbol{\lambda}^\Phi_k + \boldsymbol{K}^{\lambda^\Phi}(\widehat{\nabla\Phi}_k - \nabla\Phi(\mathbf{u}_{k+1}, \boldsymbol{\theta}))^\mathsf{T}$
      $\boldsymbol{\lambda}^{G_i}_{k+1} := (\boldsymbol{I}_{n_u} - \boldsymbol{K}^{\lambda^{G_i}})\boldsymbol{\lambda}^{G_i}_k + \boldsymbol{K}^{\lambda^{G_i}}(\widehat{\nabla G}_{i,k} - \nabla G_i(\mathbf{u}_{k+1}, \boldsymbol{\theta}))^\mathsf{T}$
          $i = 1, \ldots, n_g$   (III.3)
  else
    Keep the last operating point $(\mathbf{u}_{k+1}, \boldsymbol{\mu}_{k+1}) := (\mathbf{u}_k, \boldsymbol{\mu}_k)$, keep the last privileged directions $\boldsymbol{W}_{1,k+1} = \boldsymbol{W}_{1,k}$, the last modifiers value $(\varepsilon^\Phi_{k+1}, \varepsilon^{G_i}_{k+1}, \boldsymbol{\lambda}^\Phi_{k+1}, \boldsymbol{\lambda}^{G_i}_{k+1}) := (\varepsilon^\Phi_k, \varepsilon^{G_i}_k, \boldsymbol{\lambda}^\Phi_k, \boldsymbol{\lambda}^{G_i}_k)$ for $i = 1, \ldots, n_g$ and the last estimation of the gradients $(\widehat{\nabla\Phi}_{k+1}, \widehat{\nabla G}_{i,k+1}) := (\widehat{\nabla\Phi}_k, \widehat{\nabla G}_{i,k})$ for $i = 1, \ldots, n_g$.
  end if
end for
```

Remark 1 *Note that the TR-*ADMA *algorithm is a heuristic approach that is used to enforce convergence. However, to guarantee global convergence, any trust-region approach requires to satisfy the Cauchy decrease condition (see [7]) which is not enforced here. Moreover, the local gradient corrections are only partial, thus, resulting in gradient errors. Hence, global convergence properties are difficult to achieve. For the locally corrected models, it has been shown in [6] that, if the gradient error is bounded, then global convergence can be guaranteed. It is of future interest to establish such properties for TR-*ADMA *as well.*

IV. A SIMULATED KITE SYSTEM

A. Kite dynamics

For this simulation study, we use the Erhard model [8] previously studied in [5] for MA applications. In the Erhard model, the kite is modeled as a point in a spherical coordinate system. Using a fixed tether length, the states are the spherical angles ϑ and φ and the turning angle ψ, the latter being the kite orientation (see [5] for more details). The manipulated variable for controlling the kite is the steering deflection δ. The kite dynamics are described by the following differential equations:

$$\begin{aligned} \dot{\vartheta} &= \frac{w_{app}}{r}(\cos\psi - \frac{\tan\vartheta}{E}), \\ \dot{\varphi} &= -\frac{w_{app}}{r\sin\vartheta}\sin\psi, \\ \dot{\psi} &= w_{app}(g_s\delta - \frac{\sin\psi}{r\tan\vartheta}), \\ w_{app} &= wE\cos\vartheta, \end{aligned} \tag{IV.1}$$

where w_{app} is the apparent wind speed, r is the fixed kite radius, and w is the wind speed. The other parameters and their values are given in Table I, which also highlights the parametric mismatch between the plant and the model. The plant parameter values match closely the prototypes under development [9], [10]. The glide ratio (lift/drag ratio) E is given by

$$E = E_0 - c\delta^2, \tag{IV.2}$$

where c is the turning penalty factor. The wind speed is a function of the altitude z of the kite. To introduce structural plant-model mismatch in addition to parametric mismatch, the following wind speed laws are used:

$$w = \begin{cases} w_{ref} + (z - z_{ref})\Delta w & \text{for the model,} \\ w_{ref}(\frac{z}{z_{ref}})^a & \text{for the plant,} \end{cases} \tag{IV.3}$$

with $z = r\sin\vartheta\cos\varphi$, and where a is the surface friction coefficient and w_{ref} the reference wind speed at the reference altitude z_{ref}. The wind-altitude relationship is linear for the model and follows a power law for the plant [11].

The objective is to maximize the average line tension T over one loop,

$$T = \int_0^{tf} (\frac{1}{2}\rho A w^2)(E+1)\sqrt{E^2+1}\cos^2\vartheta \; dt, \tag{IV.4}$$

where t_f is the time period of one loop.

TABLE I: Plant and model parameters. The uncertain model parameters $\boldsymbol{\theta}$ are highlighted.

Parameter	Description	Plant value	Nominal model value	Uncertainty interval (uniform distribution)
r (m)	tether length	250	250	-
A (m^2)	surface of the kite	25	25	-
ρ (kg/m^3)	air density	1.2	1.2	-
E_0	initial lift to drag ratio	6	4.5	$\pm$11%
g_s (rad/m^2)	turning constant factor	$5{\cdot}10^{-3}$	$7{\cdot}10^{-3}$	$\pm$14%
c (1/m^2)	turning penalty factor	0.06	0.02	$\pm$10%
z_{ref} (m)	reference altitude	10	10	-
w_{ref} (m/s)	reference wind speed	8	8	-
a	surface friction coefficient	0.15	-	-
Δw (1/s)	wind speed change rate	-	10^{-3}	$\pm$5%
δ_{max} (m)	max of steering deflection	7.5	7.5	-
z_{min} (m)	minimal altitude	12.5	12.5	-

The standard constraints for kites are the periodicity of the path, a minimal altitude, bounds on the states ϑ and φ, and the saturation of the input δ:

$$\begin{aligned} \text{altitude constraint } z &= r\sin\vartheta\cos\varphi \geq z_{min}, \\ \text{bounds } &\begin{cases} 0 \leq \vartheta(t) < \frac{\pi}{2}, \\ -\frac{\pi}{2} \leq \varphi(t) \leq \frac{\pi}{2}, \\ -\delta_{max} \leq \delta(t) \leq \delta_{max}, \end{cases} \end{aligned} \tag{IV.5}$$

where t is the time. Note that there is no bound on the third state ψ.

B. Tracking the optimal reference trajectory

The optimization layer is used to generate an optimal reference trajectory for the kite. A controller is then assumed to be present to track this reference.

For optimization, we represent the reference curve as a smooth parameterized curve and we optimize over the curve parameters. In this work, the parametrization of the reference is a closed Bézier curve defined as a polynomial for $\alpha \in [0, 1]$:

$$\mathbf{c}(\alpha) := \sum_{i=0}^{n} \binom{n}{i} (1-\alpha)^{i-1}\alpha^i \mathbf{P}_i, \tag{IV.6}$$

where $\mathbf{P}_i \in \mathbb{R}^2$ play the role of curve parameters. The polynomial is used to represent the (ϑ, φ) references with $\mathbf{c}(\alpha) = [\vartheta(\alpha), \varphi(\alpha)]^T$. The dynamics (IV.1) is scaled with respect to time to make $\alpha = 1$ represent the full time period t_f using $\dot{\alpha} = 1/t_f$. The objective (IV.4) is maximized with respect to this scaled dynamics using TR-ADMA with the decision variables $\mathbf{P}_i, i = 0 \ldots n$, and t_f, with $n = 7$. Hence, the optimal curve parameters $\mathbf{P}_i^\star$ lead to a dynamically feasible periodic optimal trajectory. The resulting smooth parameterized curve can then be tracked as done in [12]. Based on the approximation of the Erhard model in cross-wind flight, one can show that the apparent wind speed w_{app} can be approximated by the kite speed w_k [13]. ψ and δ are represented in terms of the (ϑ, φ)-curve by approximating them as:

$$\psi \approx -\gamma = -\tan^{-1}(\frac{\dot{\varphi}\cos\vartheta}{\dot{\vartheta}}), \text{ and } \delta \approx -\frac{\dot{\gamma}}{g_s w_k} \tag{IV.7}$$

C. Gradient estimation

The key elements necessary to estimate the gradients in (II.10) are the computation of the privileged directions matrix $\boldsymbol{W}_{1,k}$ and the experiments needed to evaluate the directional

TABLE II: Trust-region parameters

η	γ_1	γ_2	Δ_{max}	η_1	η_2
0.015	1/4	3/4	0.1	1/2	2

derivatives along the privileged directions. The matrix $\boldsymbol{W}_{1,k}$ is computed by assuming a parametric uncertainty interval of $\pm\,(5$ to $14)\%$ of the respective nominal model values given in Table I. The Monte-Carlo samples are collected within the uncertainty interval by assuming a uniform probability distribution. $\boldsymbol{W}_{1,k}$ is then computed via (II.5)- (II.9). At the first RTO iteration, the number of privileged directions are set by the criterion (II.9) on the singular value gap. However, for the subsequent input updates, this number is kept fixed as it fixes the experimental cost for gradient estimation. Although the privileged directions are recomputed at each iteration, their number is fixed at $n_r = \bar{n}_r$. Note that updating the gradients in fewer than necessary directions may result in larger optimality loss.

To estimate the directional derivatives, forward finite-difference experiments are performed on the plant along the privileged directions. To reduce the impact of measurement noise, the modifiers are filtered as described in (III.3).

D. Optimization results

As the kites used for power generation are unstable, a controller must continuously adjust the steering deflection δ to ensure that the kite does not crash. For the purpose of this simulation study, we assume that a perfect path-following controller exists, which ensures that the kite follows a given periodic reference path. Gaussian noise with a magnitude of 3% of the nominal quantities is added to the plant measurements. The uncertain parameters are E_0, g_s, c, a and Δw. Upon computing the privileged directions offline at the model optimum, it is possible to have at most four privileged directions with DMA. However, with ADMA, the user can choose from 1 to n_u directions for gradient estimation. This becomes a trade-off between the number of iterations for gradients evaluation and the rate of convergence that can be achieved. Here, the number of input directions is fixed at $\bar{n}_r = 2$.

In this simulation study, the plant optimal inputs $\mathbf{u}_p^\star$ are known, which allows quantifying the optimality loss, that is, the fraction of improvement compared to the plant optimum, as:

$$\text{optimality loss} := \frac{\Phi_p(\mathbf{u}_p^\star) - \Phi_p(\mathbf{u}_\infty)}{\Phi_p(\mathbf{u}_p^\star)}. \tag{IV.8}$$

Note that convergence is reached after 14 iterations. Three cases are discussed and their important features are summarized in Table III. The trust-region parameters are given in Table II.

Case 1 considers the application of ADMA using noise-free measurements and no trust region. The top plots of Figure 1 shows that ADMA drives the plant towards its optimum but then the system becomes unstable and starts oscillating. This is due to the fact that, by construction,

TABLE III: Investigated cases (NA: not applicable)

$Case$	1	2	3
$\bar{n}_r$	2	2	2
N	500	500	500
trust region	**no**	**yes**	**yes**
noise	**0**	**0**	**3**
Filter eigenvalues $\boldsymbol{K}^{\varepsilon^{\Phi}}$,$\boldsymbol{K}^{\varepsilon^{G_i}}$	0.1	-	0.95
Filter eigenvalues $\boldsymbol{K}^{\lambda^{G_i}}$, $\boldsymbol{K}^{\lambda^{\Phi}}$	0.1	-	0.25
optimality loss	NA	2%	7.1±2.2%

first-order corrections are enforced, but because the second-order (Hessian) corrections are not made, ADMA without trust region "jumps" ahead of the plant optimum and needs to start again far away from it. The left plot at the top compares the predicted and actual behavior. The large jump in model prediction is the result of letting the plant free to jump far away from one iteration to the next. Hence, without controlling the input steps, the method is not applicable to real cases. In fact, even if full gradients are corrected as is traditionally done in MA, plant optimality is not reached, and the algorithm results in oscillations. A better performance can be achieved by formulating a convex problem at the price of reduced model accuracy. Clearly, the incorporation of a trust-region can help control the step length, thereby enforcing convergence (due to first-order KKT matching), while avoiding large jumps.

Case 2 considers TR-ADMA using noise-free measurements. As a result, the plant exhibits a perfect behavior as seen in the middle plots of Figure 1. Large improvement in plant performance happens during the first seven iterations, after which the algorithm slows down as the trust region adjusts itself to avoid jumping around the plant optimum. The left plot in the middle shows how the trust-region adjustment decisions are made. A poor prediction causes tightening of the trust region, thereby leading to a better agreement between prediction and actual performance. Two privileged directions appear to be appropriate in the sense that they lead to little optimality loss. More input directions could be chosen, which however will inevitably result in a larger experimental effort for gradient evaluation. We also observed that the incorporation of four privileged directions reduces the optimality loss very close to zero.

Case 3 considers TR-ADMA and noisy measurements. The finite-difference scheme gives only rough estimates of the plant gradients. Therefore, the modifiers need filtering. The bottom plots of Figure 1 show that it is possible to obtain good performance if the filtering is appropriate. Note that, even with high noise, convergence is achieved. Since the gradient corrections are not so accurate due to noise, the trust region shrinks, thereby resulting in slower convergence. For Case 3, we performed 40 simulation runs and each run with different random noise realizations, for which the TR-ADMA algorithm converges every time with an average

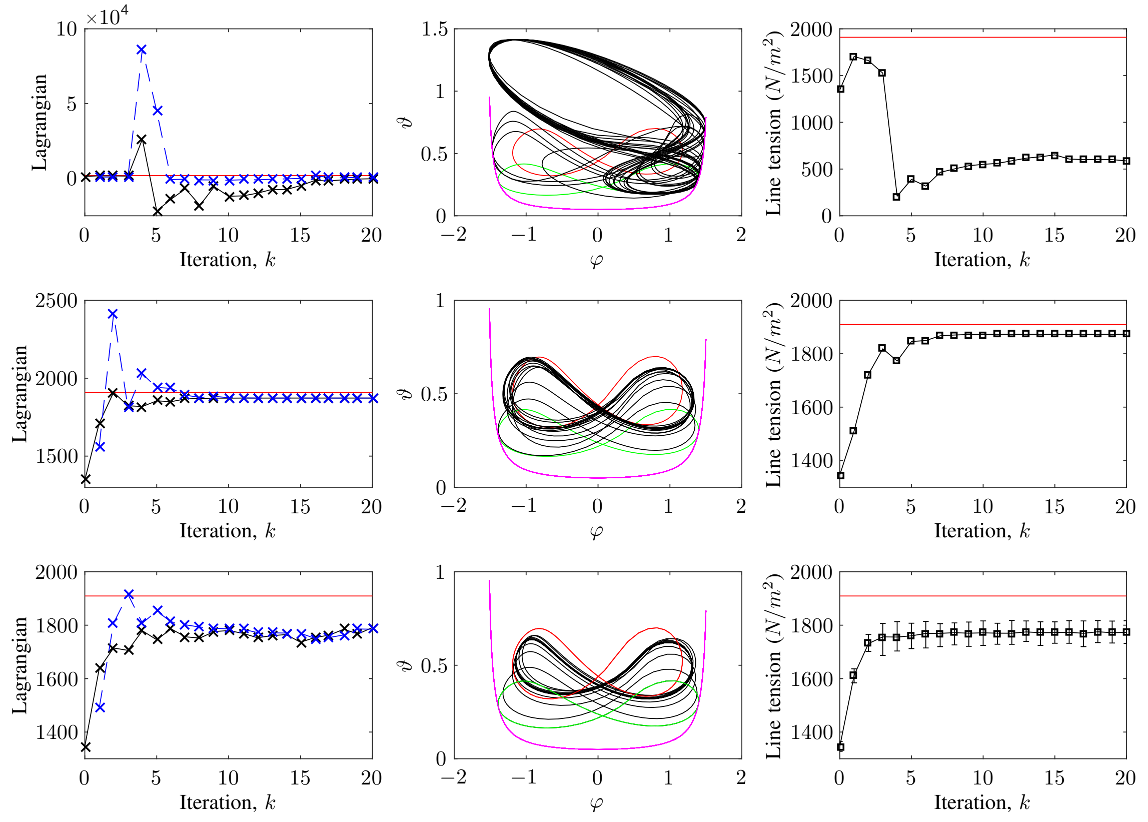

Fig. 1: ADMA applied to a simulated kite. **Top plots:** Case 1. **Middle plots:** Case 2. **Bottom plots:** Case 3. The first two plots correspond to a single noise realization, while the last plot of Case 3 shows the average performance over 40 noise realizations. **Left plots**: Black solid lines - Plant Lagrangian, Blue dashed lines - Model Lagrangian, Red - Plant Lagrangian at the plant optimum. **Center Plots**: Green - Model optimal path, Red - Plant optimal path, Black - Plant behavior over several iterations, Magenta - Altitude constraint. **Right plots**: Black - Plant output evolution. Red - Optimal line tension.

optimality loss of 7.1 percent.

V. CONCLUSIONS

For the case study of a simulated airborne wind energy system, ADMA without a trust region is not able to converge as the affine local corrections are insufficient to capture the plant behavior far from the current operating point. Such a problem can be overcome by controlling the input steps via trust region, which changes the algorithm behavior drastically, leading to excellent overall performance. Measurement noise is nicely handled by the combination of filtering the modifiers and trust-region adaptation.

REFERENCES

[1] A. Marchetti, B. Chachuat, and D. Bonvin, "Modifier-adaptation methodology for real-time optimization," *Ind. Eng. Chem. Res.*, vol. 48, no. 13, pp. 6022–6033, 2009.

[2] M. Singhal, A. G. Marchetti, T. Faulwasser, and D. Bonvin, "Improved directional derivatives for modifier-adaptation schemes," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5718 – 5723, 2017.

[3] M. Singhal, A. G. Marchetti, T. Faulwasser, and D. Bonvin, "Active directional modifier adaptation for real-time optimization," *Comp. Chem. Engng.*, 2018 (In press).

[4] G. Bunin, "On the equivalence between the modifier-adaptation and trust-region frameworks," *Comp. Chem. Engng.*, vol. 71, pp. 154 – 157, 2014.

[5] S. Costello, G. François, and D. Bonvin, "A directional modifier-adaptation algorithm for real-time optimization," *J. Process Contr.*, vol. 39, pp. 64 – 76, 2016.

[6] L. Biegler, Y. Lang, and W. Lin, "Multi-scale optimization for process systems engineering," *Comp. Chem. Engng.*, vol. 60, pp. 17–30, 2014.

[7] A. Conn, N. Gould, and P. Toint, *Trust Region Methods*, vol. 1. Philadelphia, PA, USA: MOS-SIAM Series on Optimization, 2000.

[8] M. Erhard and H. Strauch, "Control of towing kites for seagoing vessels," *IEEE Trans. Contr. Syst. Tech.*, vol. 21, pp. 1629–1640, Sept 2013.

[9] R. Ruiterkamp and S. Sieberling, *Description and Preliminary Test Results of a Six Degrees of Freedom Rigid Wing Pumping System*, pp. 443–458. Springer Berlin Heidelberg, 2013.

[10] F. Fritz, *Application of an Automated Kite System for Ship Propulsion and Power Generation*, pp. 359–372. Springer Berlin Heidelberg, 2013.

[11] C. L. Archer, *An Introduction to Meteorology for Airborne Wind Energy*, pp. 81–94. Springer Berlin Heidelberg, 2013.

[12] S. Diwale, T. Faulwasser, and C. Jones, "Model predictive path-following control for airborne wind energy systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13270 – 13275, 2017.

[13] S. Costello, G. François, and D. Bonvin, "Real-time optimizing control of an experimental crosswind power kite," *IEEE Trans. Contr. Sys. Tech.*, vol. PP, no. 99, pp. 1–16, 2017.