Stochastic Approximation for Identification of Non-Linear Differential-Algebraic Equations with Process Disturbances

Robert Bereza, Oscar Eriksson, Mohamed R.-H. Abdalmoaty, David Broman and Håkan Hjalmarsson

Abstract— **Differential-algebraic equations, commonly used to model physical systems, are the basis for many equation-based object-oriented modeling languages. When systems described by such equations are influenced by unknown process disturbances, estimating unknown parameters from experimental data becomes difficult. This is because of problems with the existence of well-defined solutions and the computational tractability of estimators. In this paper, we propose a way to minimize a cost function—whose minimizer is a consistent estimator of the true parameters—using stochastic gradient descent. This approach scales significantly better with the number of unknown parameters than other currently available methods for the same type of problem. The performance of the method is demonstrated through a simulation study with three unknown parameters. The experiments show a significantly reduced variance of the estimator, compared to an output error method neglecting the influence of process disturbances, as well as an ability to reduce the estimation bias of parameters that the output error method particularly struggles with.**

I. Introduction

Differential-algebraic equations (DAEs) are a class of models commonly used to describe physical systems. In particular, they are the mathematical basis for equation-based object-oriented modeling languages [1] such as Modelica[1], MathWorks Simscape, or VHDL-AMS. These kinds of languages are commonly used to model and simulate complex physical systems. However, in practice, such models can contain unknown parameters that have to be identified using measured data. The presence of both measurement noise and process disturbances affecting the dynamics makes the identification of such parameters challenging. In this paper, we address the problem of computationally tractable and consistent estimation of unknown parameters for a class of non-linear DAEs using experimental data.

DAEs are different from ordinary differential equations (ODEs), and solving DAEs is generally harder than solving ODEs. The differential index (or index) of a system of DAEs is the minimum number of times we need to differentiate all or some equations of the DAEs to explicitly determine the solution as a function of time, at which point the DAEs become equivalent to a set of (implicit) ODEs. Consequently, an ODE has index 0 and we can view the index of a system of DAEs as a measure of its distance to (implicit) ODEs [2]. Numerical DAE solvers, such as the DASSL [3] family of solvers, typically handle DAEs with index 1 (and index 2 in special cases). Solving high-index DAEs, therefore, involves a transformation of the DAEs prior to numerical solving so that the transformed DAEs are both of sufficiently low index and are numerically stable with respect to their original constraint equations. Several efficient algorithms for index reduction [4] [5] and stabilization [6] [7] of DAEs exist. An additional challenge with DAEs is the modeling of process disturbances. Derivatives of the disturbances can appear in the solution of the DAEs, so then special care has to be taken to ensure that the solution is well-defined.

A. Prior Work

Because of the aforementioned difficulties, while some prior work on parameter estimation for DAE models has been done, process disturbances have often been neglected, e.g. in [8] [9]. It is known that neglecting process disturbances when estimating parameters for non-linear systems can lead to a loss of consistency as well as an increased variance error, see e.g. [10]. Process disturbances have been occasionally considered in methods for state estimation, such as [11] [12], but then the DAE system is assumed to be in semi-explicit form with only additive disturbances. Conditions for the identification problem to be well-defined for linear DAEs are provided in [13], and the same is done for non-linear DAEs in [14]. In the latter, an approximate method for state estimation using a particle filter is developed. However, the proposed method for computing the likelihood function is restricted to systems that can be rewritten in a particular form, which can be challenging or require heuristics such as the removal of disturbances from certain parts of the model.

To the best of our knowledge, the method proposed in [15] is the only one of its kind, treating non-linear DAE models that can be simulated using existing numerical solvers, with process disturbances modeled as stochastic processes. The method is implemented and demonstrated with a numerical example for estimating a scalar parameter using grid search, but the method scales very poorly with the number of unknown parameters and the size of the search space. This is because the estimates are found by approximating the cost function using Monte-Carlo simulations, and the number of required simulations grows exponentially with the number

This work was supported by the Swedish Research Council under contracts 2019-04956 and 2016-06079 (the research environment NewLEADS), by Digital Futures, and by the Swedish Foundation for Strategic Research (FFL15-0032).

R. Bereza and H. Hjalmarsson are with the Division of Decision and Control Systems, EECS and Digital Futures, KTH Royal Institute of Technology, SE-100 44 Stockholm Sweden (robbj, hjalmars)@kth.se.

O. Eriksson, D. Broman are with the Division of Software and Computer Systems, EECS and Digital Futures, KTH Royal Institute of Technology, SE-100 44 Stockholm Sweden (oerikss, dbro)@kth.se

M. Abdalmoaty is with the Division of Systems and Control, Uppsala University, 751 05 Uppsala Sweden mohamed.abdalmoaty@it.uu.se

[1] https://www.modelica.org/

of unknown parameters. In this paper, we consider the same problem and show how stochastic gradient descent can be applied for its solution, which improves computational tractability. We solve the problem of obtaining an unbiased estimate of the gradient cost function, allowing us to solve problems with several unknown parameters in a computationally tractable way. Implementation details are also provided, and we demonstrate the approach through a simulation study on a simple model with three unknown parameters.

B. Contributions

We consider the problem of estimating parameters of non-linear DAEs with process disturbances that are modeled as continuous-time stochastic processes. We use the same method as in [15], which gives us an estimator that is consistent even in the presence of such disturbances, but with a different algorithm. The newly proposed algorithm is based on classical stochastic gradient descent and scales significantly better with the number of unknown parameters than the algorithm used in the aforementioned paper. This makes problems with several unknown parameters computationally tractable. However, for stochastic gradient descent to converge, an unbiased estimate of the gradient of the cost function is needed. Obtaining such an estimate for the considered cost function requires additional steps, which are taken in this paper. Specifically, our contributions are:

1) We demonstrate how the considered prediction error can be minimized with stochastic gradient descent and how to compute unbiased gradient estimates using sensitivity analysis of DAEs.
2) We provide implementation details on how to use the developed approach, together with the ADAM algorithm [16] that extends stochastic gradient descent, to obtain estimates that are computationally tractable in cases with several unknown variables.

II. Problem Formulation

We consider general non-linear DAEs on the form

$$F(\dot{x}(t), x(t), u(t), w(t); \theta) = 0 \tag{1a}$$

$$y(t) = q(x(t), u(t); \theta) + v(t), \tag{1b}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state of the system, $u(t) \in \mathbb{R}^{n_u}$ is the control input, $y(t) \in \mathbb{R}^{n_y}$ is the output, $w(t) \in \mathbb{R}^{n_w}$ is the process disturbance, and $v(t) \in \mathbb{R}^{n_v}$ is white zero-mean measurement noise. The model is parameterized by the parameter vector $\theta \in \mathbb{R}^{n_\theta}$. The model of the process disturbance is described in more detail in Section III-C. The problem we aim to solve is to, given the data set

$$D_N(T) = \left\{ (y(t_k), u(s)) : k = 1, \dots, N, \quad s \in [0, T] \right\},$$

estimate the value of the parameter vector θ parameterizing a model in the model set (1), that is also assumed to have generated the data. Note that the control input u is user-specified and therefore known for all t, while we only have access to N samples of the otherwise unknown output y.

III. Estimator using Stochastic Gradient Descent

A. Estimation Method and Gradient Computation

Estimators, such as the maximum likelihood estimator and the prediction error method using the one-step ahead optimal predictor, are intractable to compute for general non-linear DAEs. We instead use a simpler predictor proposed in [17], which provides consistent estimates and was used previously for non-linear DAEs in [15]. This predictor is the mean of the model output, and our estimator $\hat{\theta}$ is then obtained by minimizing the following cost function:

$$J_N(\theta) = \frac{1}{N} \sum_{k=1}^{N} \| y(t_k) - \mathrm{E}[y(t_k; \theta); \theta] \|^2, \tag{2}$$

where $y(t_k)$ is the measured output of the system, and $y(t_k; \theta)$ is the simulated output of the model parameterized by θ, both computed at time t_k. The expectation of the model output is taken over the process disturbance $w(t)$ and the measurement noise $v(t)$. Assuming that we can interchange derivatives and expectations, the gradient of the cost function is given by

$$\nabla_\theta J_N(\theta) = \frac{2}{N} \sum_{k=1}^{N} \mathrm{E}[\nabla_\theta y(t_k; \theta)]^T (\mathrm{E}[y(t_k; \theta)] - y(t_k)), \tag{3}$$

where $\nabla_\theta y(t_k; \theta)$ denotes the Jacobian matrix of $y(t_k; \theta)$ with respect to θ. In general, the expected values of the model output and its Jacobian are intractable to compute. However, it is possible to obtain an unbiased estimate of (3), which can be used for minimizing the cost function using stochastic gradient descent. For a given θ, let $\nabla_\theta \hat{y}^{(1)}(t_k; \theta)$ and $\hat{y}^{(2)}(t_k; \theta)$ be two independent and unbiased estimates of $\mathrm{E}[\nabla_\theta y(t_k; \theta)]$ and $\mathrm{E}[y(t_k; \theta)]$ respectively, which are also replaced in (3). Then the obtained value is an unbiased estimate of $\nabla_\theta J_N(\theta)$. The estimates $\nabla_\theta \hat{y}^{(1)}(t_k; \theta)$ and $\hat{y}^{(2)}(t_k; \theta)$ can be computed by taking the mean over some independent simulated realizations of the model output and its gradient. Note that, if these two estimates are not made independent, the estimate of the gradient of the cost function will, in general, be biased and not allow stochastic gradient descent to converge. The estimate $\nabla_\theta \hat{y}^{(1)}(t_k; \theta)$ is non-trivial to compute, but we show how this can be done in Section III-B.

Note that this approach provides improved scaling with the number of unknown parameters, compared to the most recent method for this type of problem from [15], which uses grid search to find the minimizer of (2). For that method, the number of times the model (1) has to be solved scales exponentially with the number of unknown parameters. For the proposed approach using stochastic gradient descent, the number of times the system of DAEs has to be solved does not change with the number of parameters. Instead, the number of equations in the set of DAEs, as well as the number of variables, will grow linearly with the number of unknown parameters, as we will see in the coming section.

B. Sensitivity Analysis

In this section, we allow the parameter vector θ to parameterize not only the model in (1), but also the process disturbance $w(t)$, whose model will be discussed in Section III-C. For simplicity, we will drop the explicit dependence on t from our notation. As we saw earlier, to estimate the gradient of the cost function, we need to compute the derivatives (sensitivities) of the model output with respect to θ. There are two families of methods able to do this: forward sensitivity methods [18, Sec. 2.5] and the adjoint sensitivity method [19]. Forward sensitivity methods are simpler and suitable for situations when the parameter vector is not of too high dimension. It is this type of method we will use in this paper, and one can obtain the desired sensitivities as follows:

Define new variables $s^{(i)} := \frac{\partial x}{\partial \theta_i}$, $\dot{s}^{(i)} := \frac{\partial \dot{x}}{\partial \theta_i}$, and $r^{(i)} := \frac{\partial w}{\partial \theta_i}$, where θ_i denotes the i:th element of the parameter vector θ. If we differentiate (1) with respect to θ_i, using the chain rule, we obtain

$$\frac{\partial F}{\partial \dot{x}}\dot{s}^{(i)}+\frac{\partial F}{\partial x}s^{(i)}+\frac{\partial F}{\partial w}r^{(i)}+\frac{\partial F}{\partial \theta_i}=0, \tag{4a}$$

$$\frac{\partial y}{\partial \theta_i}=\frac{\partial q}{\partial x}s^{(i)}+\frac{\partial q}{\partial \theta_i}. \tag{4b}$$

Note that this is a system of linear DAEs in the variables $s^{(i)}$ and $\dot{s}^{(i)}$, with an affine output function, if $r^{(i)}$ is considered as an external signal. These equations can then be added to the original DAE system (1). Note that the total number of equations and variables both grow linearly with the dimension of θ.

For details on implementing algorithms for solving sensitivity equations like (4), see [18, Sec. 2.5] and the references therein. Furthermore, many index-1 systems are such that their sensitivity equations are also index-1. This holds for DAEs that have been index-reduced and stabilized using the method [7], based on [2], as well as systems obtained by index-reducing DAEs according to the structural analysis methods of, e.g., [5] or [20]. In Section IV, we use this property to solve the sensitivity equations by simply appending them to the nominal system of DAEs.

C. Disturbance Modeling

We use the same approach as [15] for modeling the process disturbance as a continuous-time stochastic process. If we assume that the disturbances have rational spectrum, then their second-order properties can be modeled by

$$dx_w(t)=A(\theta)x_w(t)dt+B(\theta)dz_c(t) \tag{5a}$$

$$w(t)=Cx_w(t), \tag{5b}$$

where $dz_c(t)$ is a process with orthogonal increments and incremental variance $\mathbb{E}[dz_c(t)dz_c^T(t)]$ equal to the identity matrix. Details on how to ensure sufficient differentiability of the disturbance, as well as exact discretization of the model, can be found in [15]. The resulting discrete-time model (under the assumption of uniform sampling), that has the same second-order properties as its continuous-time counterpart, is then given by

$$x_w^{(m)}(\tau_{k+1})=A_d(\theta)x_w^{(m)}(\tau_k)+B_d(\theta)z_m(\tau_k) \tag{6a}$$

$$w^{(m)}(\tau_k)=Cx_w^{(m)}(\tau_k). \tag{6b}$$

The superscript (m) denotes the index of the realization, where realizations for different m are made to be independent. We also assume that the disturbance $w(t)$ is a Gaussian process, which means that $z_m(\tau_k)$ is a discrete-time zero-mean Gaussian white noise process with identity covariance.

As we will see later, our DAE solver will have varying step sizes and we will therefore need to compute $w(t)$ for arbitrary t, which are not known a priori. This is challenging, because it is unfeasible to store values of the process disturbance for all $t \in [0,T]$, and the discrete-time model (6) only provides samples at a finite number of time instants. The approach we take is to generate N_w samples $w(\tau_1), \cdots, w(\tau_{N_w})$ using (6), and linearly interpolate between the two neighbors closest in time when we need to compute $w(t)$ for a different value of t. While more advanced interpolation methods—such as spline interpolation or polynomial interpolation of higher order—are also applicable, linear interpolation performs sufficiently well for the problem considered in Section IV. These interpolation methods have in common that they will cause our realization of the disturbance to have a different frequency spectrum than our disturbance models (5) and (6), though this effect can be reduced by using densely spaced τ_k. If one would want to preserve the spectrum of the disturbance realization even after sampling, one could sample $w(t)$ from its conditional distribution given the neighboring samples $w(\tau_k)$ and $w(\tau_{k+1})$, which can be done easily when the white noise used to generate the disturbance is Gaussian.

IV. Numerical Experiment

In this section, we perform simulation experiments on the model of a pendulum in Cartesian coordinates. The mathematical description of the model is given in Section IV-A. Section IV-B describes how the data used in the experiment is generated, and an output error method neglecting the disturbance $w(t)$—used for comparison with the proposed method explicitly modeling the disturbances—is also described. In Section IV-C, ADAM, the optimization method used for minimizing the prediction error, is introduced. Finally, in Section IV-D, the proposed method using the ADAM algorithm is compared to the approach based on grid-search from [15] as well as the output error method that neglects the process disturbances altogether. To solve the system of DAEs (1), we use the IDA solver from the Sundials suite [18], similar to default solvers in many equation-based object-oriented modeling languages, such OpenModelica or Dymola. It uses a variable step method, which allows it to handle stiff problems, and means that the time step is dynamically changed during solving. The code is implemented in Julia[2] using the DiffEq package [21] and can be found on

[2] https://julialang.org/

Github[3].

A. Model

The method is evaluated on the same model as in [15]. The original form of the model is not suitable for numerical solving due to its high index, and it is therefore transformed to a stabilized, index-1, first-order form by manually applying the method [7]. In this form, the model is given by

$$\dot{x}_1(t) = x_4(t) - 2\dot{x}_6(t)x_1(t) \tag{7a}$$

$$\dot{x}_2(t) = x_5(t) - 2\dot{x}_6(t)x_2(t) \tag{7b}$$

$$m\dot{x}_4(t) = \dot{\tilde{x}}_3(t)x_1(t) - k|x_4(t)|x_4(t) + u(t) + w^2(t) \tag{7c}$$

$$m\dot{x}_5(t) = \dot{\tilde{x}}_3(t)x_2(t) - k|x_5(t)|x_5(t) - mg \tag{7d}$$

$$L^2 = x_1^2(t) + x_2^2(t) \tag{7e}$$

$$0 = x_4(t)x_1(t) + x_5(t)x_2(t) \tag{7f}$$

where g is the gravitational acceleration, $x_1(t)$ and $x_2(t)$ denote the x- and y-position of the pendulum respectively, $\dot{\tilde{x}}_3(t)$ is a differential variable substituted for the tension per unit length of the pendulum-arm during the index reduction process, $x_4(t)$ and $x_5(t)$ are the pendulum velocity in the x- and y-direction respectively, while $x_6(t) = 0$ is a dummy variable ensuring that we have the same number of variables as equations. The free parameters of this model are subsets of $\{m, L, k\}$ whose elements represent, respectively, the mass of the pendulum, the length of the pendulum arm, and the drag coefficient. The process disturbance is assumed to be scalar and is not parameterized by θ in these experiments. The model of the disturbance is given by (5) with matrices

$$A = \begin{bmatrix} 0 & 1 \\ -4^2 & -0.8 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{8}$$

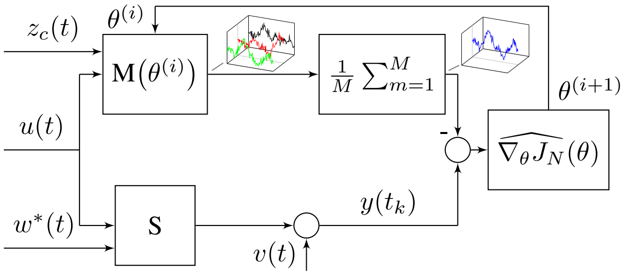
The output of this model is then re-scaled depending on its use, as described in the next section. The control input is taken as a known realization of the same model as the process disturbance, but with a different scale. The output of the model is chosen as the angle of the pendulum, given by

$$y(t) = \arctan(-x_1(t)/x_2(t)) + v(t). \tag{9}$$

The sensitivity equations are computed from (7) as described in Section III-B, and remain index-1.

B. Experimental Setup

The experimental setup used for a single data set is shown in Figure 1. We do not use real data in this example, but instead generate 100 independent data sets $D_N^{(1)}(T), ..., D_N^{(100)}(T)$, representing measurements of the true system, simulated from the model (7) with $N = 50000$ samples. The data sets contain $u(t)$ and the output (9) with $v(t)$ sampled from a Gaussian distribution with zero mean and variance 0.002. The model output realizations are instead generated with $v(t) = 0$, since the measurement noise has zero mean and including $v(t)$ neither lowers the variance nor improves the bias of the output estimate. For both the data sets and the model output, the process disturbances $w(t)$ and input $u(t)$ are generated using (5) with matrices given by (8), scaled with factors 0.6 and 0.2 respectively. The used realizations of $w(t)$ are independent between the data sets and the model output. To closer imitate a realistic scenario with measured data, inter-sample values of the process disturbance for the true system are obtained by sampling them from their distribution conditioned on the a priori generated samples, while linear interpolation is used instead for the input and for simulating the model. The output is sampled uniformly with a sampling time of $t_{k+1} - t_k = 0.1\,\mathrm{s}$, while the process disturbance and input are also sampled uniformly and 10 times as frequently as the output, with a sampling time of $\tau_{k+1} - \tau_k = 0.01\,\mathrm{s}$. Finally, the unbiased estimates of the model output and its gradient, used in the proposed method, are generated averaging 4 realizations each. The relative and absolute tolerances of the DAE solver are set to 10^{-5} and 10^{-8}, respectively.



$z_c(t)$
$\theta^{(i)}$
$\mathrm{M}\big(\theta^{(i)}\big)$
$\frac{1}{M}\sum_{m=1}^{M}$
$\theta^{(i+1)}$
$u(t)$
-
$\widehat{\nabla_\theta J_N(\theta)}$
$w^*(t)$
S
$y(t_k)$
$v(t)$

Fig. 1: The setup of the estimation procedure. $\mathrm{M}\Big(\theta^{(i)}\Big)$ represents the model, including the model of the pendulum (7), its sensitivity equations, and the disturbance model (6). S denotes the true system, where $w^*(t)$ represents the true, unknown, process disturbance. The model takes the true control input and simulated white noise $z_c(t)$ to produce several output realizations (shown in the small 3D plot to the left) that are averaged to estimate the expected value of the output and the Jacobian of the output (shown in the small 3D plot to the right). This allows one to estimate the cost function gradient, used to iteratively optimize the parameters θ.

To illustrate the benefits of explicitly modeling the process disturbances, an output error method neglecting the disturbance $w(t)$ is also implemented for comparison. For that method, the model output is computed in the same way as for the proposed method, except that we set $w(t) = 0$ for all t. This makes the model output deterministic, which allows us to compute the cost function exactly, and there is no need to use stochastic gradient descent for the minimization. Instead, the Levenberg-Marquardt algorithm, first proposed in [22], is used to minimize the cost. Specifically, the implementation of the algorithm included in the Julia package LsqFit[4] is used, with default tolerances. The Leverberg-Marquardt algorithm also uses the Jacobian of the output with respect to the unknown parameters, computed using forward sensitivity analysis as described in Section III-B.

[3] `https://github.com/br4sco/dae-param-est/tree/CDC22`

[4] `https://github.com/JuliaNLSolvers/LsqFit.jl`

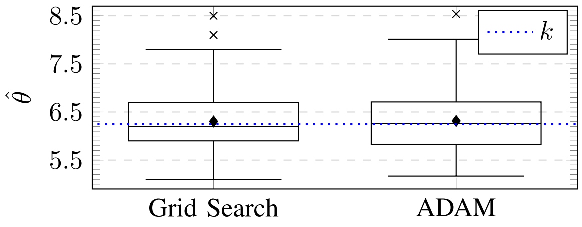


Fig. 2: Comparison of grid-search and ADAM algorithm for estimating a single unknown parameter, marked by a blue dotted line. For both methods, the same 100 independent data sets are used, from which 5000 samples are taken. The horizontal lines and diamonds represent the median and mean, respectively.

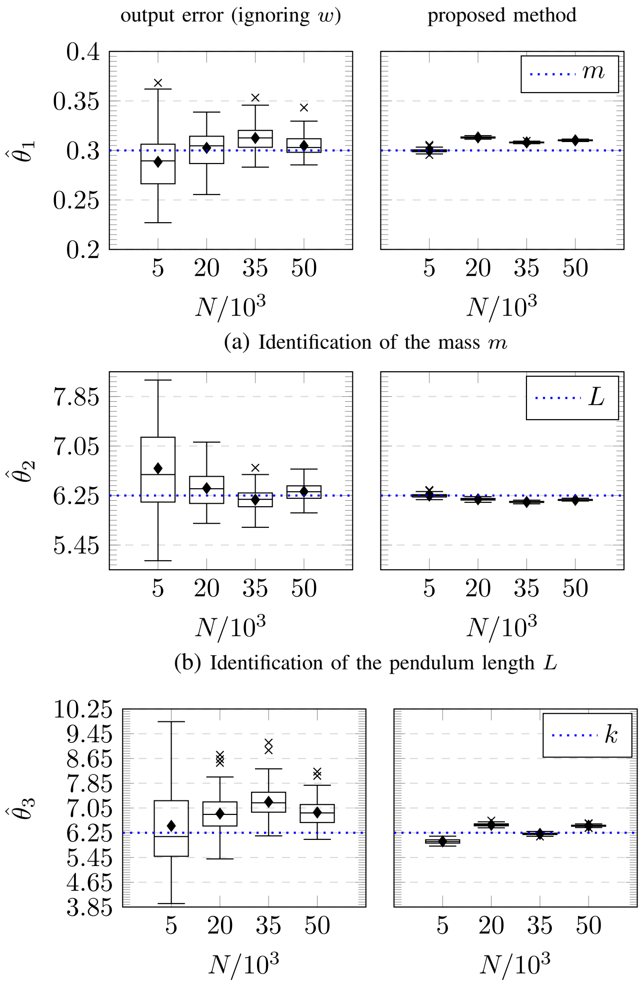
C. Stochastic Optimization Method

For minimizing the cost function (2) with process disturbances, the ADAM algorithm [16], extending stochastic gradient descent, is used. This algorithm is meant to speed up convergence, especially when the problem is ill-conditioned, and it is commonly used for optimizing neural networks. We use Algorithm 1 from [16] exactly, except that we alter the returned value. Since the algorithm might only converge to a noise ball around the minimizer when a constant step size is used, we make the algorithm return the average of the estimates from the last K iterations of the algorithm. This approach is inspired by Polyak-Ruppert averaging [23].

The hyper-parameters for the ADAM algorithm, described in detail in [16], are set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\alpha = 1$, and $\epsilon = 0$. The algorithm is run for $n_{its} = 100$ iterations. When the gradient of the cost function (2) is computed, the first 500 samples of the outputs are discarded to allow for any transient effects from the initial conditions to die out. Finally, to ensure that all parameters of the pendulum model (7) are approximately of the same order of magnitude for the iterative minimization, we re-scale the mass m with a factor 10 and instead work with $\tilde{m} = 10m$. In practice, this is equivalent to re-scaling the component corresponding to m in each gradient step with a factor 0.1. For convergence guarantees of the ADAM algorithm, see [16].

D. Simulation Results

As a first experiment, the proposed method using the ADAM algorithm is compared to the approach based on grid-search from [15]. Only the drag coefficient k is estimated, and $N = 5000$ samples are used. For the grid search, the cost function is estimated for 26 evenly spaced values of the parameter in the interval $\theta \in [5.0, 7.5]$. For each value of θ, the mean is taken over 100 independent simulations of the model output. The same 100 realizations of $\{z_m(\tau_k)\}_k$ are used for every value of θ, to improve the smoothness of the cost function. For each of the 100 data sets $D_N^{(1)}(T)$ to $D_N^{(100)}(T)$, the grid search chooses the one out of 26 values of θ that minimizes the cost (2), with the expected value replaced by the mean over the 100 independent realizations. The proposed method using the ADAM algorithm uses the initial parameter estimate $\theta^{(0)} = 5.0$. A comparison of the two methods can be seen in Figure 2.



(a) Identification of the mass m

(b) Identification of the pendulum length L

(c) Identification of the drag coefficient k

Fig. 3: Comparing the output error method ignoring w (left column) with the proposed method explicitly modeling w (right column). For every N, the statistics of the estimators are summarized over $E = 100$ data sets. The true parameter values are marked with blue dotted lines. The horizontal line and the diamond in each box denote the median and the mean of the estimates, respectively.

In a second experiment, the proposed method is compared to an output error method assuming $w(t) = 0$, realized using the Levenberg-Marquardt algorithm for minimization. It also uses the forward sensitivity analysis for computing the state Jacobian, as described in more detail in Section III-B. The methods are used to simultaneously identify the three free parameters of the pendulum, i.e. $\theta = [m \;\; L \;\; k]^T$. The two methods are tested for different numbers of data points. A comparison of their performance can be found in Figure 3, with the initial estimate chosen as $\theta^{(0)} = [0.5 \;\; 4.25 \;\; 4.25]^T$ and the true value $\theta = [0.3 \;\; 6.25 \;\; 6.25]^T$.

Grid search was not included in this experiment because of its poor scaling. With three unknown parameters and $N = 50000$, the proposed method took approximately $55\,\mathrm{min}$ to compute an estimate for a single dataset. In this time, the grid search only managed to sample the cost function at 15 different parameter values, instead of 26^3 values needed to maintain the same grid density as in the first experiment.

V. Discussion

In Figure 2, we see that the stochastic method has comparable performance to grid search in the case where a single scalar parameter is unknown. This indicates that the ADAM algorithm successfully approximates the same minima that were found using grid search. Figure 3 shows the results of applying the proposed stochastic method to a problem with three unknown parameters. Note that the proposed method provides estimates with significantly lower variance than an output error method neglecting the process disturbances. However, while the used estimator is consistent in theory (see [15]), no clear improvement of the small bias for the proposed method is visible as N increases. The bias is still significantly lower than for the output error method which, despite having a relatively low bias for its estimates of θ_1 and θ_2, retains a large bias when estimating θ_3, even for large N. This matches the observations in [15], and demonstrates an ability of the proposed method to lower the bias due to explicitly modeling the disturbances.

The proposed method provides a significant improvement over the state-of-the-art method for this type of model with process disturbances used in [15]. Instead of having exponential scaling with the number of unknown parameters, the bottleneck of the proposed method is the size of the system of DAEs that has to be solved when computing sensitivities. However, while the size grows linearly with the number of unknown parameters, this can also become prohibitively expensive to solve for large problems with many unknowns. The scaling can be improved by using the adjoint sensitivity method, described in [19], instead of the forward sensitivity approach. How the adjoint method can be applied for the problem formulation considered in this paper, and how to also then include parameters of the disturbance model, is something we are currently working on. For future work, we are also considering how the presence of multivariate, and potentially correlated, disturbances can affect the estimation.

VI. Conclusions

We propose a way to use stochastic gradient descent to compute consistent estimators of parameters for non-linear DAE models under the influence of process disturbances. Other methods for identification of non-linear DAEs either neglect process disturbances, are restrictive in the types of DAEs they consider, or require a large number of Monte-Carlo simulations to compute. The proposed approach allows one to iteratively compute the minimizer of the prediction error with only a few solutions of the DAEs per iteration, even when several parameters are unknown. Implementation details of the method are provided, and its performance is demonstrated in a simulation study on a non-linear system of DAEs with three unknown parameters, where a previous method using grid search would be computationally unfeasible to perform on the same hardware. The proposed method demonstrates a significant reduction of the variance of the estimator compared to a method neglecting process disturbances, and a reduction of the bias in estimating a parameter that the latter method particularly struggles with.

References

[1] D. Broman, "Meta-Languages and Semantics for Equation- Based Modeling and Simulation," Ph.D. dissertation, Department of Computer and Information Science, Linköping University, Sweden, 2010.

[2] C. W. Gear, "Differential-Algebraic Equation Index Transformations," *SIAM Journal on Scientific and Statistical Computing*, vol. 9, no. 1, pp. 39–47, Jan. 1988.

[3] L. R. Petzold, "Description of DASSL: a differential/algebraic system solver," Sandia National Labs., Livermore, CA (USA), Tech. Rep. SAND-82-8637; CONF-820810-21, Sep. 1982.

[4] J. Pryce, "Solving high-index DAEs by Taylor series," *Numerical algorithms*, vol. 19, no. 1, pp. 195–211, 1998.

[5] C. C. Pantelides, "The Consistent Initialization of Differential-Algebraic Systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 9, no. 2, pp. 213–231, 1988.

[6] S. E. Mattsson and G. Söderlind, "Index reduction in differential-algebraic equations using dummy derivatives," *SIAM journal on scientific computing*, vol. 14, no. 3, pp. 677–692, 1993.

[7] M. Otter and H. Elmqvist, "Transformation of Differential Algebraic Array Equations to Index One Form," in *Proceedings of the 12th International Modelica Conference*, 2017, pp. 565–579.

[8] H. G. Bock, E. Kostina, and J. P. Schlöder, "Numerical Methods for Parameter Estimation in Nonlinear Differential Algebraic Equations," *GAMM-Mitteilungen*, vol. 30, no. 2, pp. 376–408, 2007.

[9] W. R. Esposito and C. A. Floudas, "Global Optimization for the Parameter Estimation of Differential-Algebraic Systems," *Industrial & Engineering Chemistry Research*, vol. 39, no. 5, pp. 1291–1310, 2000.

[10] A. Hagenblad, L. Ljung, and A. Wills, "Maximum likelihood identification of Wiener models," *Automatica*, vol. 44, no. 11, pp. 2697–2705, 2008.

[11] P. Mobed, S. Munusamy, D. Bhattacharyya, and R. Rengaswamy, "State and parameter estimation in distributed constrained systems. 1. Extended Kalman filtering of a special class of differential-algebraic equation systems," *Industrial and Engineering Chemistry Research*, vol. 56, no. 1, pp. 206–215, 2017.

[12] V. Becerra, P. Roberts, and G. Griffiths, "Applying the extended Kalman filter to systems described by nonlinear differential-algebraic equations," *Control engineering practice*, vol. 9, no. 3, pp. 267–281, 2001.

[13] M. Gerdin, T. B. Schön, T. Glad, F. Gustafsson, and L. Ljung, "On parameter and state estimation for linear differential-algebraic equations," *Automatica*, vol. 43, no. 3, pp. 416–425, 2007.

[14] M. Gerdin, "Identification and estimation for models described by differential-algebraic equations," Ph.D. dissertation, Linköpings universitet, Sweden, 2006.

[15] M. R.-H. Abdalmoaty, O. Eriksson, R. Bereza, D. Broman, and H. Hjalmarsson, "Identification of non-linear differential-algebraic equation models with process disturbances," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 2300–2305.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *The International Conference on Learning Representations (ICLR)*, 2015.

[17] M. R.-H. Abdalmoaty and H. Hjalmarsson, "Linear prediction error methods for stochastic nonlinear models," *Automatica*, vol. 105, pp. 49–63, 2019.

[18] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.

[19] Y. Cao, S. Li, L. Petzold, and R. Serban, "Adjoint sensitivity analysis for differential-algebraic equations: The adjoint dae system and its numerical solution," *SIAM journal on scientific computing*, vol. 24, no. 3, pp. 1076–1089, 2003.

[20] J. D. Pryce, "A Simple Structural Analysis Method for DAEs," *BIT Numerical Mathematics*, vol. 41, no. 2, 2001.

[21] C. Rackauckas and Q. Nie, "Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in Julia," *Journal of Open Research Software*, vol. 5, no. 1, 2017.

[22] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[23] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM journal on control and optimization*, vol. 30, no. 4, pp. 838–855, 1992.