Change-point and Model Estimation with Heteroskedastic Noise and Unknown Model Structure

Anas Alhashimi[1,2], Thomas Nolte[1], Alessandro V. Papadopoulos[1]

***Abstract*— In this paper, we investigate the problem of modeling time-series as a process generated through (i) switching between several independent sub-models; (ii) where each sub-model has heteroskedastic noise, and (iii) a polynomial bias, describing nonlinear dependency on system input. First, we propose a generic nonlinear and heteroskedastic statistical model for the process. Then, we design Maximum Likelihood (ML) parameters estimation method capable of handling heteroscedasticity and exploiting constraints on model structure. We investigate solving the intractable ML optimization using population-based stochastic numerical methods. We then find possible model change-points that maximize the likelihood without over-fitting measurement noise. Finally, we verify the usefulness of the proposed technique in a practically relevant case study, the execution-time of odometry estimation for a robot operating radar sensor, and evaluate the different proposed procedures using both simulations and field data.**

I. Introduction

Assume that the generated time-series of a process is affected by a nonlinear function of an input variable and heteroskedastic noise variance, so both the bias and variance depend on system states. We also assume that the time-series is the result of switching between several sub-sequences, each one is generated using a different and independent sub-model. Such heteroskedastic models appear in financial risk analysis and economics to detect changes in stock returns and short-term interest rates [9], in health care, for prostate cancer incidence and mortality rates [22], in tire Industry, for example determine the length of the footprint [15], and in environmental time-series, for significant wave heights of storm peak events across the Gulf of Mexico [21].

A practically relevant case study is the execution-time of the odometry estimation pipeline [1], [2], where the robot pose is estimated using a scanning radar sensor mounted on top of a mobile robot. The radar returns ranges of existing targets surrounding the robot. The robot translation and orientation are determined after each new scan by matching detections from consecutive scans. The matching problem is solved using numerical optimization so that the execution-time is related to the number of detections in each scan, but the relationship might not be linear and also has variable variations. The dataset plotted in Figure 1 shows the relation between the point-cloud density (the number of detections in a scan) with the execution-time, t_{ex}, of the odometry estimation application. We assume that the input (number of detections in our example) is already available information. Considering estimation of robot translation and orientation in real-time, the estimation is improved as we process more scans. However, relying on worst-case execution-time will limit the frequency of running the estimation process. On the other hand, building more precise model for the execution-time will assist in reducing the number of ignored scans compared to worst-case execution-time instance and hence will improve odometry estimation.

The general idea here is to identify the number of sub-models and then for each one estimate both the bias and variance functions using recorded data sets of inputs and corresponding outputs (similar to the one depicted in Figure 1).

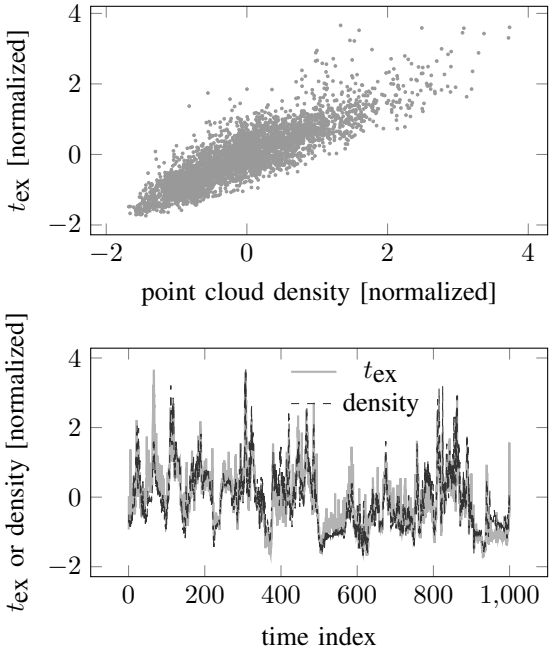

Fig. 1. The relation between execution-time and point cloud density for the odometry estimation example is shown on the top plot. The variance is changing with density values, which indicates heteroscedasticity. The bottom plot shows the time series for both the input (cloud density) and the output (execution-time).

We go with the following approach in the paper: starting from a training sequence similar to the one in Figure 1, we: 1) divide the training sequence into sub-sequences using opportune change-point detection algorithm; 2) learning model parameters and complexity for each sub-model using ML estimation methods on the corresponding sub-sequence; 3)

This work was supported by the Swedish Research Council (VR) via the projects "Practical Probabilistic Timing Analysis of Real-Time Systems (PARIS)" and "Pervasive Self-Optimizing Computing Infrastructure (PSI)", and by the Knowledge Foundation (KKS) via the project FIESTA.

[1] Mälardalen University, Västerås, Sweden.
[2]University of Baghdad, Baghdad, Iraq.

apply the learned model on the validation set to verify the usefulness of this approach.

In this paper, we consider a generic nonlinear model that switches among different nonlinear heteroskedastic sub-models. We propose an approach for learning the number of sub-models composing the nonlinear model, and we estimate the model parameters using ML methods (closed-form for homoskedastic models and numerically using population-based Particle Swarm Optimization (PSO) for heteroskedastic ones).

The remainder of this paper is structured as follows. Section II analyzes the related work, Section III formalizes the problem, Section IV describes the proposed approach, Section V discusses the simulation and numerical results, and Section VI concludes the paper.

II. Related work

We are focusing on solving two problems in this research, the first one is change-point detection and the second one is parameter estimation for nonlinear and heteroskedastic models. So we list the important related work for each one in separate paragraphs.

change-point detection

In the literature, finding a single change-point in a sequence is solved optimally using non-parametric methods like cumulative sum (CUSUM) [27] and cropped version of CUSUM to improve detection at sequence boundaries [10], or parametric methods like Likelihood Ratio Test (LRT) [23] that requires exact knowledge of the signal model.

Multiple change-points detection is done either using global methods that are considering the whole sequence to determine the cost of each candidate or local methods that are considering only samples inside a window around the candidate sample. Global methods are usually based on Binary Segmentation [29], [30], an iteratively repeating single change-point method over all the resulted sub-sequences for minimizing a cost function that favors the newly created sub-sequences against the original whole sequence over all possible change-points in a sequence. For example, Generalized Likelihood Ratio (GLR) cost is used in [14] to decide change-points in the execution-time sequence of hidden Markov model (HMM). Penalized likelihood cost used to account for over-fitting due to increased model complexity. Akaike Information Criterion (AIC), Schwarz Information Criterion (SIC), and Maximum Description Length (MDL) are common examples of penalization [6]. Segment Neighborhood [4] is an exact method that computes costs for all possible change-points between zero and a specified max number. However, the computation cost is significant, i.e., $O(n^3)$ where n is the sequence length. All those methods assume a known signal model. An example of local methods is screening and ranking algorithm [26] which is based on detecting changes in signal statistics inside the sliding window. A useful review of existing methods is [34], [31] and [13]. The last compared several methods and their performances, simulation results favor Bayesian methods over classical ones. In this paper, we consider heteroskedastic models, change-points with heteroskedastic noise discussed in [36], [11] and [16] where summing the squares of the partial sums statistics and its variants were proposed. In this paper we are doing another approach to deal with the heteroskedastic models, it is based on transforming the heteroskedastic model into a homoskedastic one and then applying the classical methods to find change-points.

parameter estimation with heteroskedastic noise

When small heteroskedasticity is presented, Ordinary Least Squares (OLS) is usually used for estimating model parameters. The solution is asymptotically unbiased but also an inefficient estimator [8]. However, when large heteroskedasticity is presented, it will give biased estimate of the variance and incorrect confidence intervals and statistical inference for the parameters [35].

The heteroskedasticity in heteroskedastic systems are classified to either be dependent on the states or independent of the states. We focus in this paper on state-dependent noise structure, when the noise structure is an unknown nonlinear function of the states. Similar models have been studied in econometrics literature, e.g., a two-step estimation procedure for models where the i^{th} disturbance variance σ_i is of the form $\sigma_i = \sigma^2 x_i^{\lambda}$, where x_i is the state, σ^2 and λ are unknown model parameters was proposed by [28]. The authors in [17] examined this procedure in details with more generic variance models similar to $\sigma_i = e^{x_i \lambda}$, and compared with the iterative ML. The analysis carried out by [17] shows that the two-step estimation procedure is inconsistent in the heteroskedasticity parameters and then proposed a consistent two-step estimation procedure. The authors in [7] proposed Bayesian estimators for heteroskedastic systems for variance models similar to $\sigma_i = \sigma^2 x_i^{-\lambda}$. The Bayesian approach of [7] were extended to multidimensional by [33]. They used opportune Markov chain Monte Carlo (MCMC) sampler to estimate the parameters then compared the performance with the aforementioned estimators (the two-step and iterative ML). They showed that Bayesian estimators commit better performance in the Root-Mean-Square Error (RMSE) and the interquartile range sense.

The precedently cited papers [28], [17], [7], [33] assume models useful for econometrics, however, in this research, the authors will present analysis for parameter estimation in generic non-linear and heteroskedastic systems, in which, both the non-linearity and variance functions are modeled using a polynomial of suitable order. Recently, a Bayesian parameters estimation method handling heteroskedasticity and capable to exploit prior information about the model parameters presented in [3] to solve sensor calibration applications for several heteroskedastic models with increased complexity. The Bayesian problem was solved using MCMC methods. The idea of [3] is to simplify the heteroskedastic model in the favor of reduced computation complexity. In this paper, we are not doing approximations but we seek to estimate the parameters of the full noise model. The estimation problem will be of increased complexity and hence

intractable. Therefore, we investigate the use of population-based stochastic methods for solving ML estimation problem.

Statement of contributions

1) a generic nonlinear model constructed from switching between several nonlinear heteroskedastic sub-models,
2) learning the number of sub-models from data using statistical methods,
3) estimate model parameters using ML methods (closed-form for homoskedastic models and numerically using population-based PSO for heteroskedastic ones)

III. Problem Statement

A general model consists of several nonlinear sub-models

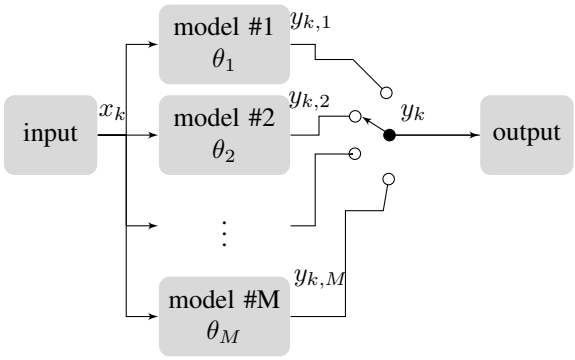

Fig. 2. The proposed model structure.

We propose the following generic model, depicted in Figure 2, for the sequence of outputs $y_{0:N}$ and inputs $x_{0:N}$ (execution-times and density in our example) where N is the sample size. It consists of M different sub-models that are combined according to an unobserved variable, each sub-model is described by a different set of parameters θ_i, $i \in \{1, 2, \ldots, M\}$. At each time instant k the output is taken from only one of those models, according to

$$y_k = \sum_{i=1}^{M} \delta_{k,i} y_{k,i} \tag{1}$$

where $\delta_{k,i} \in \{0, 1\}$ is a binary selection variable such that only one element of the vector $\Delta_k := \begin{bmatrix}\delta_{k,1} & \cdots & \delta_{k,M}\end{bmatrix}^\top$ is 1 and all others are zeros i.e. $\sum_{i=1}^{M} \delta_{k,i} = 1$. We also assume that when the output is chosen from a model i at an instant k, i.e $\delta_{k,i} = 1$, it continues to have the output from the same sub-model i during the instants $k+1, k+2, \cdots$ for a sufficient number of samples to avoid identifiability problems. This sufficient number is not known precisely for each model, but a lower bound for all sub-models $K_{\min}$ is assumed to be known.

Each one of the sub-models is described by the following generic heteroskedastic non-linear model

$$y_{k,i} = f_{\text{bias,i}}(x_k) + f_{\sigma,i}(x_k) e_{k,i} \tag{2}$$

where

- $y_{k,i} \in \mathbb{R}$ is the execution-time of the model i at instant k;
- $x_k \in \mathbb{R}$ is the recorded input of the process at instant k;
- $f_{\text{bias}}(\cdot)$ and $f_\sigma(\cdot)$ are generic nonlinear functions for bias and variance respectively, such that $f : \mathbb{R} \to \mathbb{R}_{>0}$;
- $e_{k,i} \sim \mathcal{N}(0, 1)$ is independent and identically distributed (iid) standard Gaussian noise.

Using polynomial expansions of the functions and for sub-model m we get

$$\begin{aligned} f_{\text{bias,m}}(x_k) &= \sum_{i=0}^{nb_m} \alpha_{i,m} x_k^i \\ f_{\sigma,m}(x_k) &= \sum_{i=0}^{nv_m} \beta_{i,m} x_k^i \end{aligned} \tag{3}$$

where $\alpha_m := [\alpha_{0,m}, \cdots, \alpha_{nb_m,m}]^\top$ and $\beta_m := [\beta_{0,m}, \cdots, \beta_{nv_m,m}]^\top$, nb_m, and nv_m are the linear model parameters for bias and variance functions and the corresponding model orders, respectively;

Then each sub-model is characterized by the set of parameters $\theta_m = [\alpha_m, \beta_m]$ of size $nb_m + nv_m + 2$. Rewriting the above equations in vector form gives

$$y_{0:N} = \text{diag}\left(\begin{bmatrix} y_{0:N,1} & y_{0:N,2} & \cdots & y_{0:N,M} \end{bmatrix} \Delta\right) \tag{4}$$

where

$$\begin{aligned} y_{0:N,m} &:= \begin{bmatrix} y_{0,m} & \cdots & y_{N,m} \end{bmatrix}^\top \\ \Delta &:= \begin{bmatrix} \Delta_0 & \cdots & \Delta_N \end{bmatrix} \end{aligned} \tag{5}$$

Where $N > m$ is the number of samples. The full model parameter vector is the stack of all parameter vectors $\theta = [\Delta, \theta_1, \ldots, \theta_M]$ of size $N+1+\sum_{i=1}^{M} nb_i + \sum_{i=1}^{M} nv_i + 2M$. Rewriting (3) in vector form give

$$y_{0:N,m} = H_b \alpha_m + \text{diag}\left(H_v \beta_m\right) e_m \tag{6}$$

where

$$\begin{aligned} H_b &:= \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{nb_m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^{nb_m} \end{bmatrix} \\ H_v &:= \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{nv_m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^{nv_m} \end{bmatrix} \\ e_m &:= \begin{bmatrix} e_{0,m} & \cdots & e_{N,m} \end{bmatrix}^\top \end{aligned} \tag{7}$$

and $\text{diag}(A)$ is the diagonal matrix of vector A. The second term is a zero mean Gaussian with covariance matrix $\text{diag}\left(H_v \beta_m\right)^2$.

IV. Methodology

Starting from a training sequence similar to the one in Figure 1, we first learn the parameters and complexity for model (6) using ML estimation methods as described in Section IV-A. Then we divide the training sequence into two sub-sequences, if it is necessary, using change-point

detection methods described in Section IV-B. Then we repeat the above steps for each newly formed sub-sequence until no further sub-sequence can be obtained. The number of sub-sequences is $\bar{m} \geq m$ and the corresponding indexes vector $[\tau_0, \tau_1, \ldots, \tau_{\bar{m}+1}]$ where $\tau_0 = 1$ and $\tau_{\bar{m}+1} = K$. In the last step, we combine similar sub-sequences using the procedure described in Section IV-C ending up with m different models.

A. Estimating parameters and complexity of (6)

In this section, we devise a method for learning θ_m and the complexities n_{vm} and n_{bm} from training sub-sequences $y_{\tau_{m-1}:\tau_m}$ and inputs $x_{\tau_{m-1}:\tau_m}$ which have $\widetilde{N} = \tau_{m-1} - \tau_m$ samples. Given the frequentist assumptions on the unknowns, we want to apply ML estimate, i.e., seek to solve

$$\widehat{\theta}_m = \arg\max_{\theta_m} \mathbb{P}\left[y_{\tau_{m-1}:\tau_m} \mid x_{\tau_{m-1}:\tau_m}, \theta_m, n_{vm}, n_{bm}\right] \quad (8)$$

for all values of complexity starting from one to a specified maximum value. Larger order values makes (8) intractable, so we solve it numerically using PSO techniques, a widely used population-based method in optimization that one may use to explore the likelihood function searching for the maxima. It has the advantage of not using the gradient of the function being optimized, so it does not require the problem to be differentiable. It is a very powerful and flexible algorithm, but there is no guarantee to converge toward the global maxima. It has been firstly described by Kennedy and Eberhart [20] for solving collective intelligence in biological populations, and then been used to solve various optimization problems [18], [25], [24]. Many varieties are suggested for tuning the hyperparameters, see for example [5]. We used the linear decreasing inertia weight since it gives the minimum error [5]. We used the Matlab implementation of PSO the algorithm from Matthew Kelly [19], more details about the algorithm and its steps are in the early researches [20], [32], [12].

Once we have the estimated parameters for all suggested model complexities, we select the model that minimizes the corrected AIC score for sample size

$$\text{AIC} = 2p - 2\text{loglikelihood} + \frac{2p^2 + 2p}{\widetilde{N} - p - 1} \quad (9)$$

where $p = n_{vm} + n_{bm}$.

B. Dividing the sequence into sub-sequences

Assume that we have the estimated model

$$y_{\tau_{m-1}:\tau_m,m} = H_b\widehat{\alpha}_m + \text{diag}\left(H_v\widehat{\beta}_m\right) e_m \quad (10)$$

or

$$y^*_{\tau_{m-1}:\tau_m,m} := y_{\tau_{m-1}:\tau_m,m} - H_b\widehat{\alpha}_m = \text{diag}\left(H_v\widehat{\beta}_m\right) e_m \quad (11)$$

with its parameters estimated from the output sub-sequence $y_{\tau_{m-1}:\tau_m}$ and its corresponding input $x_{\tau_{m-1}:\tau_m}$. Now we investigate the possibility of having the execution-time sequence consisting of two smaller sub-sequences each one generated using a different and unknown model. For doing so, we need to find if a change-point will result in better modeling for the sequence (expressing the data more efficiently).

Common statistical measures for deciding the changes are mean, variance, and the sum of squares. Since in our model, we have an available input sequence, we need to modify the change-point detection algorithm to consider not only the output sequence but also the input. Adding to that, the heteroscedasticity nature of the noise. We first transform (10) into a homoskedastic model

$$\begin{aligned} \breve{y}_{\tau_{m-1}:\tau_m,m} &:= \text{diag}(H_v\widehat{\beta}_m)^{-1} y_{\tau_{m-1}:\tau_m,m} \\ &= \text{diag}(H_v\widehat{\beta}_m)^{-1} H_b\widehat{\alpha}_m + e_m \end{aligned} \quad (12)$$

and write in terms of variations,

$$\widetilde{y}_{\tau_{m-1}:\tau_m,m} = e_m \quad (13)$$

where

$$\widetilde{y}_{\tau_{m-1}:\tau_m,m} := \text{diag}\left(H_v\widehat{\beta}_m\right)^{-1} \left(y_{\tau_{m-1}:\tau_m,m} - H_b\widehat{\alpha}_m\right).$$

Notice that the error distribution in model (13) is always standard Gaussian. Then we apply the following change-point algorithms

1) CUSUM test: It was first introduced by [27] to detect changes between the sample mean and the partial mean using the statistics [31]

$$T_k := \max_{1 \leq k < N} \left| \frac{1}{\sqrt{N}} \left(\sum_{i=1}^{k} x_i - k\overline{x} \right) \right| \quad (14)$$

where $\overline{x} := N^{-1} \sum_{i=1}^{N} x_i$ is the sample mean. This test is based on homoskedastic Gaussian models. It is not designed for our generic nonlinear and heteroskedastic model (2) and it is not easy to investigate its asymptotic properties due to heteroscedasticity, therefore, we suggest using the transformed models (13) or (12) in the test (14) which is homoskedastic when we use the ML estimate of the parameters. This will give the alternative test

$$T_k := \max_{\tau_{m-1} \leq k < \tau_m} \left| \frac{1}{\sqrt{\widetilde{N}}} \left(\sum_{i=\tau_{m-1}}^{k} \mathcal{Y}_i - k\overline{\mathcal{Y}} \right) \right| \quad (15)$$

where $\mathcal{Y}$ could be any of $y, y^*, \breve{y}$, or $\widetilde{y}$. The alternative test is easy to compute and will be useful if we can show that the change-points of the original model (2) correspond to the change-points of the transformed models (12) and (13). We are not going to show that mathematically since it is beyond the scope of this paper, but we alternatively use the ablation study in Section V-A to validate change-point detection in mean and variance. The selected T_k that maximize the CUSUM test will divide the sequence $y_{\tau_{m-1}:\tau_m}$ into two sub-sequences $y_{\tau_{m-1}:T_k}$ and $y_{T_{k+1}:\tau_m}$. We then accept this division when having at least K_{min} samples and minimizing the AIC score in (9) to avoid noise overfitting.

2) Likelihood Ratio Test (LRT): It is a parametric test, which means that the signal model should be known before the test. It compares the likelihood of the whole sequence to the likelihood of the sub-sequences. Applying LRT to model (2) requires parameters and complexity estimation for the generated sub-sequences at each possible change-point, this is a time-consuming process, especially for heteroskedastic models. However, it is possible to apply it on the transformed models (12) or (13) since the error will be Gaussian or standard Gaussian, respectively. The test will be

$$\Lambda_k = \max_{\tau_{m-1} \le k < \tau_m} \frac{L(\widehat{\mu}_{\tau_{m-1}:\tau_m})}{L(\widehat{\mu}_{\tau_{m-1}:k}, \widehat{\mu}_{k:\tau_m})} \tag{16}$$

where $\widehat{\mu}_{a:b} := (b-a)^{-1} \sum_{i=a}^{b} y_i$ is the ML estimate of sample mean between a and b, and $L(\cdot)$ is the likelihood function. After that, similar to CUSUM, AIC and K_{min} are used to decide whether to accept the change-point or not.

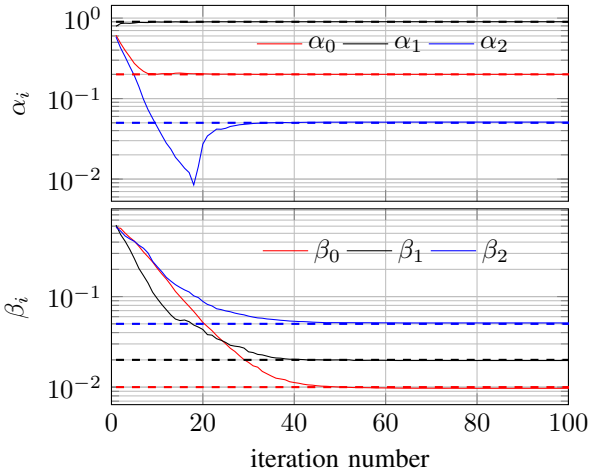

Fig. 3. Parameters estimation progress in PSO (synthetic data). The dashed lines are the true values of the parameters.

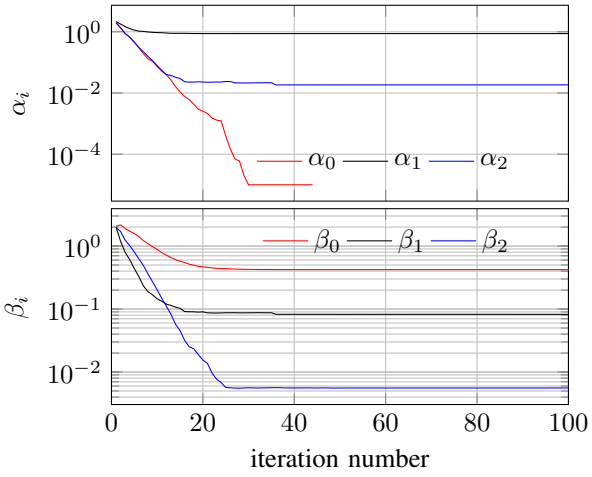

Fig. 4. Parameters estimation progress in PSO when applied to the same sequences of Figure 1.

C. Combining similar models

Once we have no more change-points, we examine all the $\bar{m}$ sub-models to combine similar models until we end up with m sub-models. We define similar models as the ones that have the same model order and the Mean Squared Error (MSE) between their parameter vectors do not exceed a defined maximum error ϵ_{max}.

V. Simulations and Numerical Results

A. Ablation study

To evaluate the performance of our method, we generated synthetic data set with heteroskedastic noise so we compare the estimated parameters with the actual ones. In PSO simulations, we used the same input sequence presented in Figure 1 to make the synthetic data closer to reality, while in change-point simulation we used synthetic input generated using the following random-walk process

$$x_{k+1} = x_k + w_k. \tag{17}$$

with $w_k \sim \mathcal{N}\left(0, 0.025^2\right)$.

PSO for parameter estimation with heteroskedasticity: To evaluate the performance of PSO for parameter estimation with heteroskedasticity, we simulated samples from a heteroskedastic model with second-order polynomial expansion for both bias and variance having random parameter values in the range $0 < \alpha_i, \beta_i < 1$. Applying PSO to estimate bias and variance functions parameters using 1000 particles, resulting in the estimation statistics presented in Table I, where each value is the average of 1000 simulations. We compared three initial sets for parameters initialization, zero vector, independent samples from standard Gaussian, and Least Squares (LS) for bias parameters and zeros for variance parameters, the results suggest using LS for initialization. Figure 3 shows one example of parameter evolution against iteration number, while Figure 4 shows the obtained parameters when using the same data set of Figure 1.

initial	mean error	error variance	iterations
LS	**0.00042682**	**2.2154e-06**	**180.285**
zeros	0.00047027	4.6296e-06	197.264
random	0.00055556	3.1156e-06	187.284

TABLE I

PSO estimation results for heteroskedastic system with different initialization. (simulated data)

Finding change-points on transformed model: We compared LRT and CUSUM for detecting change-points on our various models (10), (11), (12), and (13). Table II lists the corresponding change-point location errors (the unit is samples) where each value in the table is computed from 100 tests. As stated in Section IV-B.2, it is computationally expensive to apply LRT to heteroskedastic models (10) and (11), therefore, we approximate the variance model by keeping only the highest order parameter while forcing the others to zero (therefore colored with light gray). This will make closed-form solution possible. The results suggested

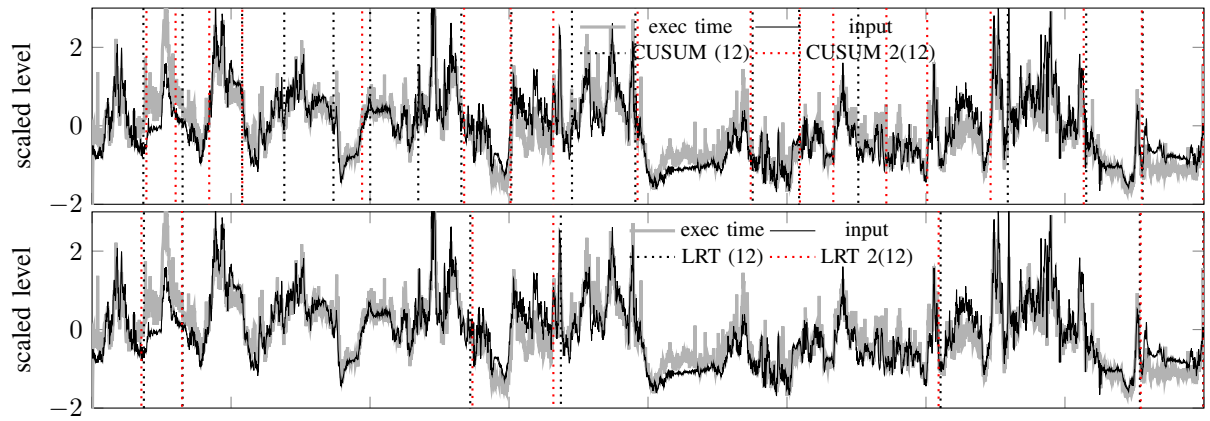

Fig. 5. The identified sub-sequences by applying CUSUM on (12) in the upper plot and LRT on (12) in the lower plot. The red dotted lines corresponds to the sub-sequences obtained from another sequence obtained from the same input (different realization). The plot generated using real dataset.

method	model	absolute error	median
LRT	y (10) (approx.)	102.78	79
LRT	y^* (11) (approx.)	132.64	29
LRT	$\breve{y}$ (12)	**36.36**	**1**
LRT	$\widetilde{y}$ (13)	122.76	28.5
CUSUM	y (10)	210.65	216
CUSUM	y^* (11)	232.62	231
CUSUM	$\breve{y}$ (12)	193.26	208
CUSUM	$\widetilde{y}$ (13)	212.72	215.5

TABLE II

CHANGE-POINT POSITION ESTIMATION ERROR FOR THE VARIOUS MODELS.

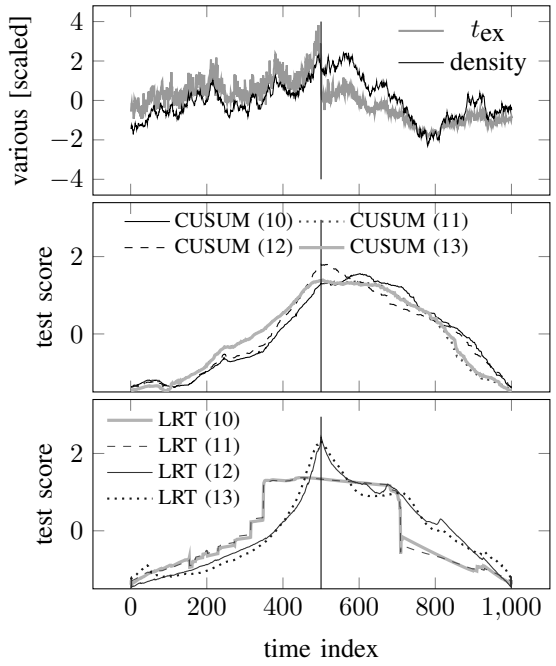

Fig. 6. Change-point detection statistics applied on the original and transformed models (synthetic data). Notice that the data has a change-point at 500.

that LRT returns a more accurate estimate of the change position than CUSUM, in general, and the heteroskedastic model (12) gives the smallest error for both methods with global min error using LRT. The second-best model based on change error median is (13) which shows that transforming the model improve estimating the change position. Surprisingly the original model (10) returns the second-best absolute error for both tests but not the error median, this could be due to neglecting the input resulting in less outliers (having absolute error closer to the median). CUSUM test, in general, having almost similar values for average absolute error and error median, which means it is over all performing worse than LRT. An example of those simulated sequences is presented in Figure 6 with the corresponding statistics.

B. Real data

We applied the algorithm described in Section IV on the same data plotted in Figure 1, the resulted sub-sequences are plotted in Figure 5 bounded by black dotted lines. The test applied to the transformed model (12) that is suggested by the simulations in the previous subsection. It is not easy to say which one performs better since we do not know the true change-points, and it should not reflect changes in the input. The red dotted lines corresponds to the sub-sequences obtained from another sequence obtained from the same input.

VI. Conclusion

PSO can be used to solve accurately the ML estimate of nonlinear and heteroskedastic model parameters when polynomial expansions are used to express the non-linearity and heteroscedasticity.

It is not practical to use LRT to find the points of change in a sequence with heteroscedasticity since the computations do not scale well, as it requires solving complex ML problems at each possible change-point. On the other hand, CUSUM does not have such restrictions, but it is hard to show the asymptotic properties.

Transforming the nonlinear heteroskedastic model into homoskedastic one maintains at least part of the change-points while making both LRT and CUSUM easier to deal with. Another advantage of the transformed model, it encodes

the input sequence with the output sequence in the newly transformed one which makes it easier to suppress possible change-points due to input rapid variations, not to actual model changes.

REFERENCES

[1] D. Adolfsson, M. Magnusson, A. Alhashimi, A. J. Lilienthal, and H. Andreasson, "CFEAR radarodometry–conservative filtering for efficient and accurate radar odometry," in *IROS*, Sept. 2021, pp. 5462–5469.

[2] ——, "Lidar-level localization with radar? the cfear approach to accurate, fast, and robust large-scale radar odometry in diverse environments," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1–20, 2022.

[3] A. Alhashimi, S. Del Favero, D. Varagnolo, T. Gustafsson, and G. Pillonetto, "Bayesian strategies for calibrating heteroskedastic static sensors with unknown model structures," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 2447–2453.

[4] I. E. Auger and C. E. Lawrence, "Algorithms for the optimal identification of segment neighborhoods," *Bulletin of mathematical biology*, vol. 51, no. 1, pp. 39–54, 1989.

[5] J. C. Bansal, P. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *2011 Third world congress on nature and biologically inspired computing*. IEEE, 2011, pp. 633–640.

[6] L. Birgé and P. Massart, "Minimal penalties for gaussian model selection," *Probability theory and related fields*, vol. 138, no. 1, pp. 33–73, 2007.

[7] W. J. Boscardin and A. Gelman, "Bayesian computation for parametric models of heteroscedasticity in the linear model," *University of California, Berkeley*, 1994.

[8] G. E. Box and W. J. Hill, "Correcting inhomogeneity of variance with power transformation weighting," *Technometrics*, vol. 16, no. 3, pp. 385–389, 1974.

[9] G. Chen, Y. K. Choi, and Y. Zhou, "Detections of changes in return by a wavelet smoother with conditional heteroscedastic volatility," *Journal of Econometrics*, vol. 143, no. 2, pp. 227–262, 2008.

[10] M. Csörgö, M. Csörgö, and L. Horváth, *Limit theorems in change-point analysis*. John Wiley & Sons, 1997.

[11] V. Dalla, L. Giraitis, and P. C. Phillips, "Testing mean stability of heteroskedastic time series," *Cowles Foundation Discussion Paper No. 2006*, (march 30, 2015), available at SSRN: https://ssrn.com/abstract=2622638 or http://dx.doi.org/10.2139/ssrn.2622638.

[12] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*, vol. 1. IEEE, 2000, pp. 84–88.

[13] I. A. Eckley, P. Fearnhead, and R. Killick, "Analysis of changepoint models," *Bayesian time series models*, vol. 205, p. 224, 2011.

[14] A. Friebe, F. Marković, A. V. Papadopoulos, and T. Nolte, "Adaptive runtime estimate of task execution times using bayesian modeling," in *2021 IEEE 27th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 2021, pp. 1–10.

[15] S. J. Ganocy and J.-Y. Sun, "Heteroscedastic change point analysis and application to footprint data," *Journal of Data Science*, vol. 13, no. 1, pp. 157–185, 2015.

[16] T. Górecki, L. Horváth, and P. Kokoszka, "Change point detection in heteroscedastic time series," *Econometrics and statistics*, vol. 7, pp. 63–88, 2018.

[17] A. C. Harvey, "Estimating regression models with multiplicative heteroscedasticity," *Econometrica: Journal of the Econometric Society*, pp. 461–465, 1976.

[18] Z. Iqbal, R. Ilyas, H. Y. Chan, and N. Ahmed, "Effective solution of university course timetabling using particle swarm optimizer based hyper heuristic approach," *Baghdad Science Journal*, vol. 18, no. 4 (Suppl.), pp. 1465–1465, 2021.

[19] M. Kelly, "Particle swarm optimization," https://github.com/MatthewPeterKelly/ParticleSwarmOptimization, 2016.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[21] R. Killick, I. A. Eckley, K. Ewans, and P. Jonathan, "Detection of changes in variance of oceanographic time-series using changepoint analysis," *Ocean Engineering*, vol. 37, no. 13, pp. 1120–1126, 2010.

[22] H.-J. Kim, M. P. Fay, E. J. Feuer, and D. N. Midthune, "Permutation tests for joinpoint regression with applications to cancer rates," *Statistics in medicine*, vol. 19, no. 3, pp. 335–351, 2000.

[23] H.-J. Kim and D. Siegmund, "The likelihood ratio test for a change-point in simple linear regression," *Biometrika*, vol. 76, no. 3, pp. 409–423, 1989.

[24] R. A. R. Mahmood, A. Abdi, and M. Hussin, "Performance evaluation of intrusion detection system using selected features and machine learning classifiers," *Baghdad Sci. J*, vol. 18, no. 2, pp. 884–898, 2021.

[25] O. S. Naif and I. J. Mohammed, "Woaip: Wireless optimization algorithm for indoor placement based on binary particle swarm optimization (bpso)," *Baghdad Science Journal*, vol. 19, no. 3, pp. 0605–0605, 2022.

[26] Y. S. Niu and H. Zhang, "The screening and ranking algorithm to detect dna copy number variations," *The annals of applied statistics*, vol. 6, no. 3, p. 1306, 2012.

[27] E. Page, "A test for a change in a parameter occurring at an unknown point," *Biometrika*, vol. 42, no. 3/4, pp. 523–527, 1955.

[28] R. E. Park, "Estimation with heteroscedastic error terms," *Econometrica (pre-1986)*, vol. 34, no. 4, p. 888, 1966.

[29] A. J. Scott and M. Knott, "A cluster analysis method for grouping means in the analysis of variance," *Biometrics*, pp. 507–512, 1974.

[30] A. Sen and M. S. Srivastava, "On tests for detecting change in mean," *The Annals of statistics*, pp. 98–108, 1975.

[31] X. Shi, C. Gallagher, R. Lund, and R. Killick, "A comparison of single and multiple changepoint techniques for time series data," *Computational Statistics & Data Analysis*, vol. 170, p. 107433, 2022.

[32] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*. IEEE, 1998, pp. 69–73.

[33] H. Tanizaki and X. Zhang, "Posterior analysis of the multiplicative heteroscedasticity model," *Communications in Statistics - Theory and Methods*, vol. 30, no. 5, pp. 855–874, 2001. [Online]. Available: https://doi.org/10.1081/STA-100002262

[34] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107299, 2020.

[35] H. White, "A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity," *Econometrica: Journal of the Econometric Society*, pp. 817–838, 1980.

[36] K.-L. Xu, "Testing for structural change under non-stationary variances," *The Econometrics Journal*, vol. 18, no. 2, pp. 274–305, 2015.