An Automatic Framework for Detecting and Characterizing Performance Degradation of Software Systems

Pengfei Zheng, Yong Qi, Yangfan Zhou, *Member, IEEE*, Pengfei Chen, Jianfeng Zhan, *Member, IEEE*, and Michael Rung-Tsong Lyu, *Fellow, IEEE*

***Abstract*—Software systems that run continuously over a long time have been frequently reported encountering gradual degradation issues. That is, as time progresses, software tends to exhibit degraded performance, deflated service capacity, or deteriorated QoS. Currently, the state-of-the-art approach of Mann-Kendall Test & Seasonal Kendall Test & Sen's Slope Estimator & Seasonal Sen's Slope Estimator (MKSK) detects and characterizes degradation via a combination of techniques in statistical trend analysis. Nevertheless, we pinpoint some drawbacks of MKSK in this paper: 1) MKSK cannot be automated for large scale software degradation analysis, 2) MKSK estimates the degradation trend of software in an oversimplified linear way, 3) MKSK is sensitive to noise, and 4) MKSK suffers from high computational complexity. To overcome all these limitations, we propose a more advanced approach called Modified Cox-Stuart Test & Iterative Hodrick-Prescott Filter (CSHP). The superiority of our CSHP approach over MKSK is validated through extensive Monte Carlo simulations, as well as a real performance dataset measured from 99 real-world web servers.**

***Index Terms*—Performance degradation, software aging, trend estimation, trend test.**

Acronym and Abbreviations:

ANOVA	Analysis of Variance
CSHP	Modified Cox-Stuart Test & Iterative Hodrick-Prescott Filter
DFT	Discrete Fourier Transform
FPR	False Positive Rate
HPF	Hodrick-Prescott Filter
IHPF	Iterative Hodrick-Prescott Filter
LMS	Least Mean Square
MKSK	Mann Kendall Test & Seasonal Kendall Test & Sen's Slope Estimator & Seasonal Sen's Slope Estimator
PSD	Power Spectral Density
QoS	Quality of Service
RMSE	Root of Mean Square
STFT	Short Time Fourier Transform
SPWVD	Smoothed Pseudo Wigner Ville Distribution
TTF, TTE	Time to Failure, Time to Exhaustion
TTP	Trend Test Power

Notations:

s-	implies statistical(ly)
$Pr\{\cdot\}$	probability
P	binary variable indicating periodicity
L	period length of periodic components
$sgn\{\cdot\}$	signum function
$sqrt\{\cdot\}$	square root
λ	smooth parameter of Hodrick-Prescott Filter
$Q_{90}\{\cdot\}$	the 90th percentile
$Q_{low}\{\cdot\}$	the 25th percentile
$Q_{high}\{\cdot\}$	the 75th percentile
$Cauchy_{stop}$	the Cauchy-type stopping criterion
η^2	the effect size of a factor in variance analysis

Manuscript received March 17, 2013; revised September 01, 2013; accepted January 08, 2014. Date of publication July 22, 2014; date of current version November 25, 2014. This work was supported by the Key Project of the National Natural Science Foundation of China (No. 60933003), the National Basic Research Program of China (973 Project No. 2014CB347701), the General Project of National Science Foundation of China (No. 61272460), the National High Technology Research & Development Program of China (No. 2012AA010904), the Shenzhen Basic Research Program (No. JCYJ20120619152636275), the Ph.D. Programs Foundation of China (No. 20120201110010), and the Research Grants Council of Hong Kong (N_CUHK405/11 of the NSFC/RGC Joint Research Scheme). This work was conducted when the first author was a visiting student in Shenzhen Research Institute, The Chinese University of Hong Kong. Associate Editor: S. Shieh.

P. Zheng, Y. Qi, and P. Chen are with the Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China (e-mail: p.f.zheng.phd@stu.xjtu.edu.cn; qiy@mail.xjtu.edu.cn).

Y. Zhou and M. R. Lyu are with the Shenzhen Research Institute, The Chinese University of Hong Kong, China, and also with the MoE Key Laboratory of High Confidence Software Technologies (CUHK Sub-Lab), The Chinese University of Hong Kong, China (e-mail: yfzhou@cse.cuhk.edu.hk; lyu@cse.cuhk.edu.hk).

J. Zhan is with the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China (e-mail: jfzhan@ncic.ac.cn).

Digital Object Identifier 10.1109/TR.2014.2338255

I. Introduction

IN THE PAST few years, performance degradation issues have been reported frequently to occur in commercial, industrial, and scientific computing software platforms or systems. Taking the multi-user telecommunication system of AT&T [1] as an example, when a communication node runs for a long time, it increasingly loses on-going calls, and is more inclined to deny access requests, even exposed to just moderate workload levels. In the meantime, complaints begin to pile up in the customer service center, as customers report the severe decline of voice quality. This abnormal condition can deteriorate in a node for several days or even weeks, gradually depleting all its capacity, and causing the node to be unavailable eventually. Typically, such a degraded software system does not exhibit a transient failure. Instead, it exhibits a gradual failure, where the system is still in service but will perform increasingly worse over time. Until now, there is no unified notion to describe this phenomenon. In different papers, researchers have named it *Software Aging* [2], *Chronics* [3], or *Smooth degradation* [1]. In this paper, we generally refer to it as *software degradation*.

To counteract degradation issues, operational monitoring of software performance provides us opportunities for in-depth analysis and fast troubleshooting. In this paper, operational monitoring refers to measuring system performance (also including resource utilization and Quality of Service (QoS)) metrics at periodic intervals. When performance measurements are gathered over a period of time, there are two core problems:

Problem 1. how to detect the presence of degradation from the collected measurements; and

Problem 2. how to quantitatively characterize the process of degradation for visual report, insightful diagnosis, and potential recovery in the next step.

This paper focuses around these two problems. We will transform them into concrete mathematical problems. The most typical syndrome of degradation is a gradual deterioration of system performance, progressive depletion of system resources, or a gradual decline of system QoS. All these symptoms can be embodied the by long-term descending or ascending trends of specific performance (or resource usage, or QoS) measurements. For instance, degradation can be substantiated by the long-term declining trend of *Throughput* [18], the chronically decreasing trend of *Available Memory* [4], [10], the ever-growing trend of service *Response Time* [4], or the conjunction thereof. We name these trends *degradation trends*. In this sense, degradation detection and quantitative characterization can be achieved by mathematically analyzing the degradation trends of relevant performance metrics. In trend analysis, *trend test*, also called *statistical test for trend*, means applying statistical hypothesis tests to detect whether the values within a time series (i.e., a series of observations with timestamp labelled) tend to increase (or decrease) over time. Meanwhile, *trend estimation* is the quantitative estimate of the long-term component of a time series. The estimated trend is a sub-time-series extracted from raw data, with short-term fluctuations and noises filtered out, characterizing the long-term dynamics. Consequently, Problem 1 and Problem 2 can be transformed as below.

Transformed Problem 1. What is the optimal trend test technique to detect the degradation trends of performance metrics?

Transformed Problem 2. What is the optimal trend estimation technique to quantitatively characterize the degradation trends detected?

Hereinafter, detection and quantitative characterization of degradation is equivalent to testing and estimating the degradation trends of performance metrics. In recent research in degradation analysis, [4], [13], [17]–[19], [25], researchers apply *the combination of the Mann-Kendall Test and the Seasonal Kendall Test for trend test*, and *the combination of Sen's Slope Estimator and the Seasonal Sen's Slope Estimator for trend estimation*. To summarize this state-of-the-art approach that mixes together four methods, we name it MKSK. The framework of MKSK is illustrated in Fig. 1. In this paper, through comprehensive investigation and empirical evaluation, we pinpoint five major deficiencies of MKSK. To overcome the limitations, we propose a more advanced framework called CSHP (a combination of our modified Cox-Stuart test and our Iterative Hodrick-Prescott filter, cf. Fig. 2). MKSK and CSHP are evaluated and compared upon a simulated dataset composed of 15,625,000 synthetic time series that simulate performance measurements, and a real dataset composed of throughput and availability data measured from 99 web servers over a three-month period. The Final results verify the drawbacks of MKSK we identify, as well as the corresponding improvements made by CSHP, summarized as follows.

1) MKSK cannot be automated, making it infeasible for large-scale degradation analysis such as in cloud data centers, where a great many virtualized nodes and numerous system metrics are monitored, and degradation trend analysis can only be done automatically. Actually, to automate MKSK, two prerequisite procedures must be automatic and accurate: 1) preliminary periodicity tests, and 2) preliminary period length estimation (cf. Fig. 1). However, in practice, manual assistance, prior knowledge, or ad-hoc parameter tuning are indispensable for accurate completion of the two prerequisite procedures, making automation of MKSK practically impossible (cf. Section III). In contrast, CSHP is intrinsically insusceptible to the periodic patterns superimposed within performance measurements (i.e., with no need for preliminary periodicity analysis). By this means, CSHP could be automated naturally, and applied for large-scale degradation analysis. The detection power of CSHP is validated to be comparable to the idealized detection power of MKSK, in which case MKSK is manually assisted to complete the two prerequisite procedures accurately (cf. Section V-C).
2) MKSK characterizes the degradation trends in an oversimplified linear fashion. Only two constants (i.e., the slope and the intercept of a straight line) are used by MKSK to manifest a complicated degradation process. This manner of degradation characterization is too inaccurate to provide details for degradation diagnosis and recovery. CSHP yields nonlinear estimation of the degradation trends in a data-driven way, with no sacrifice of computational complexity. CSHP can even identify multiple distinct degraded stages within a single degradation process. By this means,

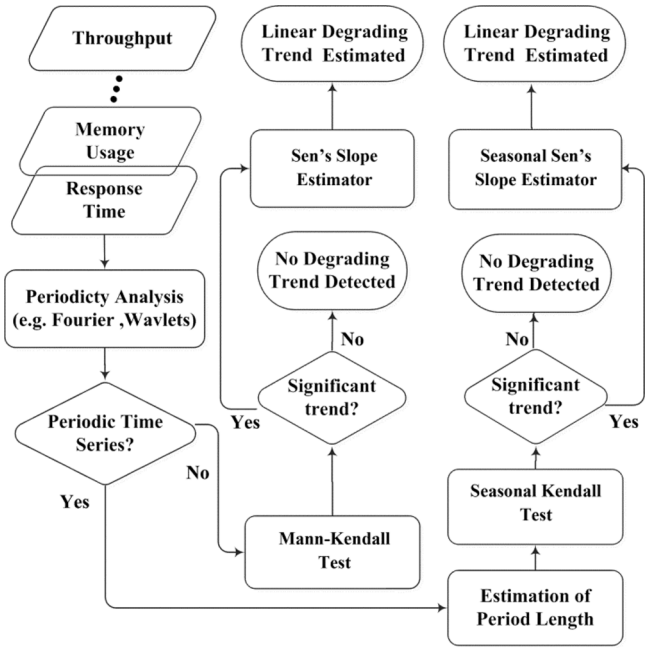

Fig. 1. Framework of the state-of-the art approach MKSK.

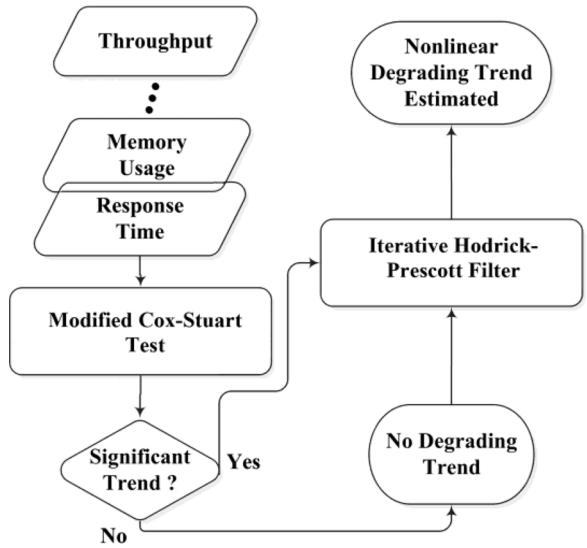

Fig. 2. Framework of CSHP.

CSHP contributes many more details to degradation diagnosis and recovery (cf. Sections V-E, V-F, and VI-B).

3) MKSK is validated to perform unstably in controlling false alarms for degradation detection. Through Monte Carlo simulations, over 50% of the simulation groups of MKSK exhibit false positive rates exceeding the nominal upper-limit 0.05. CSHP performs stable in controlling false alarms. Almost all of the simulation groups exhibit false positive rates below 0.05. In general, CSHP reduces 37.7% of the false alarms relative to MKSK (cf. Section V-D).
4) MKSK is relatively sensitive to noise. In contrast, CSHP is verified to be more robust. Via Monte Carlo simulations, with respect to the magnitude of noise sensitiveness, CSHP is approximately 50% lower than MKSK in degradation detection, and 78.9% lower in quantitative degradation characterization (cf. Sections V-C and V-E).
5) MKSK is intrinsically with high time complexity in both degradation detection and degradation characterization. Specifically, the time complexity of MKSK for degradation detection is $O(n^2)$, and $O(n^2 \log(n))$ for degradation characterization. CSHP can accomplish degradation detection and degradation characterization both in $O(n)$ time complexity. This advantage enables CSHP to be much more scalable towards vast scale degradation analysis (cf. Section VII).

II. Background and Related Work

A. Background

Degradation related system failure has been reported on a variety of software systems, such as JVM [27], [28], APACHE web servers [4], MySQL database systems [25], AT&T billing systems, SOAP servers [5], OLTP servers [9], Linux [10], clusters [14], and even Patriot missile systems [11]. For high-quality commercial software products (e.g., IBM WebSphere and Oracle11g), degradation is discovered and reported on their official websites [12], [13]. In a most recent research task, data from the VoIP (Voice over IP) platform of a major US based ISP revealed that, in each month, the number of calls affected (dropped or blocked) by common outages was only at most 30% higher than the number of calls impacted by degradation [3]. In addition, a 60-day track on a large-scale cluster (more than 4000 nodes, used as part of the index service of the Bing search engine) shows that at least 20% of machine failures have a long degradation period, during which the machine is deviating from normal behavior, but not yet failing [14]. In a word, before failure, degradation has already constituted a non-negligible threat to software systems in commercial and industrial production, especially performance-critical and safety-critical systems.

Degradation is in most cases caused by (or with) erroneous depletion of system resources (e.g., memory leaks [4], [16], orphan processes [15], memory fragmentation [10]). Other acknowledged causes include the cumulative effect of unreleased file-locks, corrupted data, and round-off errors [11]. All these degradation related bugs are usually classified as latent faults (or the so-called Mandel-bug), whose activating conditions are usually complex, uncertain, irreproducible, and may escape sophisticated software testing or verification.

B. Related Work for MKSK

In this subsection, we briefly introduce some work related to MKSK, and demonstrate its workflow. In [4], [10], [17]–[19], some performance metrics of an Apache web server were collected periodically, and MKSK was applied for degradation trend detection and estimation. As the first step, researchers manually determined whether the time series of a metric exhibits periodic patterns.

For the metrics without periodic patterns, Mann-Kendall Test was utilized. As a result, an increasing trend was detected from the time series of Response Time, and a decreasing trend was detected from the time series of Free Physical Memory [4],

[10], [17]–[19]. Thus, two types of degradation (performance deterioration, and memory depletion) were identified within the target Apache server. Sen's Slope Estimator was then utilized to estimate the degradation trends in a linear way, where for instances of Free Physical Memory the intercept represented the initial free-memory, and the slope represented the constant exhausting-rate. In this way the degradation process of the Apache server is quantitatively characterized by the degradation trends of response time and physical memory. For the metrics with periodic patterns (e.g., Used Swap Space, showing a diurnal cycle [4], [17], [19]), researchers incorporated visual analysis and harmonic techniques to estimate the period length (if multiple periodic components are superposed, only the largest scale period is considered). Then Seasonal Kendall Test and Seasonal Sen's Slope Estimator were respectively applied to identify and characterize the increasing trend of swap space [4], [17], [19].

III. Detailed Explanation for Some Drawbacks of MKSK

In Section I, we have pointed out five drawbacks of MKSK. In this section, we provide detailed explanations for the first two.

A. Susceptible to Periodic Patterns of Performance Metrics

For MKSK, depending on whether periodic (or cyclical) patterns are present in performance metrics, either the non-periodic version (i.e., Mann-Kendall Test plus Sen's Slope Estimator) or the periodic version (i.e., Seasonal Kendall Test plus Seasonal Sen's Slope Estimator) of MKSK must be selected alternatively. Therefore, the performance of MKSK is not just determined by the four trend analysis methods above, but it also relies on the correctness of two preliminary parameters:

Parameter 1 is P, a boolean type value to indicate whether the input time series of MKSK are periodic or non-periodic; and

Parameter 2 is L, an integer type value to represent the length of a period in units of the number of observations (L represents the largest scale period length if multiple periodic components are interlaced).

This induces a third question.

Problem 3. How do we prove and quantify the influence of Parameter 1 and Parameter 2 on MKSK?

In Section V-A, to answer Question 3, we will set MKSK at two distinct configurations, and compare their performance in degradation trend test and estimation. 1) The first configuration is *Ideal-MKSK*, where P and L are manually provided with always correct values. 2) The second configuration is *Random-MKSK* in which random values are assigned to P and L. In this sense, Ideal-MKSK is enriched to be the best case for MKSK, and Random-MKSK represents the worst case. If we find that Ideal-MKSK remarkably outperforms Random-MKSK, we can empirically confirm the non-negligible impact of P and L. Meanwhile, the performance gap between these results quantifies the maximum influence induced by the preliminary periodicity-analyzing procedures of MKSK. Through our extensive Monte Carlo simulations in Sections V-C and V-E, Random-MKSK is verified to be 37.96%, and 44.17% inferior to Ideal-MKSK, respectively towards detection power, and estimation accuracy of degradation trends, which proves and quantifies the sensitiveness of MKSK upon periodic behaviors of performance measurements.

Problem 4. Should periodicity analysis techniques, e.g., Fourier analysis, be applied to test P and estimate L for MKSK preliminarily?

In the following 2 subsections we will indicate that a preliminary procedure for periodic component analysis (e.g., Fourier analysis and Wavelet analysis) has remarkable disadvantages in practical application.

B. Troublesome Periodicity Test, and Period Estimation

Visual checking is the most direct, common approach to identify the periodic components of a performance time series. However, for large-scale degradation analysis where tremendous performance metrics are involved, manually visual check will become unaffordable. Therefore, researchers must resort to an automatic, adaptive procedure to achieve automatic periodicity (P) detection and period length (L) estimation. In the signal processing field, *Power Spectral Density* (*PSD*) analysis based on *Discrete Fourier transform* (*DFT*) is a well-acknowledged solution towards automatic periodicity anatomy [31]. Nevertheless, as indicated by many researchers, Fourier analysis makes rather idealistic assumptions, and is highly unreliable for time series that are non-stationary, noisy, and without quasi-sinusoid periods, all of which are quite common in performance measurements.

Therefore, uniformly using Fourier analysis for preliminary periodicity analysis is highly likely to degrade MKSK performance severely. To prove and quantify this point, we set another configuration of MKSK, the *Fourier-MKSK*. Fourier-MKSK utilizes the Fourier PSD analysis techniques we mentioned above to test the statistical significance of periodicity (P), and estimate the period length (L) of the input time series. Through evaluation in Sections V-C and V-E, we discover that Fourier-MKSK outperforms Random-MKSK, but exhibits much lower performance than Ideal-MKSK. The performance gap between Fourier-MKSK and Ideal-MKSK quantifies the negative influences of incorrect P and L that are directly induced by Fourier analysis. Via Monte Carlo simulations, Fourier-MKSK is proved to be 28.96%, and 24.81% inferior to Ideal-MKSK respectively for degradation detection power, and degradation characterization accuracy.

C. Indispensable Human Aid for Sophisticated Periodicity Analysis Techniques

To overcome the limitations of Fourier analysis, Wavelets is the most acknowledged alternative. Wavelets based spectral analysis is more advantageous and robust for analyzing periodic properties of a time series that are non-stationary and noisy, e.g., internet traffic. However, the most severe challenges for wavelet analysis are 1) selection of a mother wavelet function (e.g., Haar, Daubachies, Coiflet, Mexican Hat, or Morlet wavelet), and 2) determination of the decomposition level.

Wavelets-based interpretation of a time series is only meaningful relative to the selected mother wavelet and selected decomposition level. Adaptively determining applicable wavelets basis functions and the optimal decomposition level both is still a daunting problem that many researchers are endeavoring to

solve. In practical applications, these two intricate problems are usually resolved by tentative experiments in a trial-and-error manner, and domain knowledge or expert experiences are often indispensable. We also investigate other sophisticated techniques, including *Short Time Fourier Transform* (*STFT*) [29], and *Smoothed Pseudo Wigner Ville Distribution* (*SPWVD*) [30], which are superior to Fourier analysis. However, STFT and SPWVD both require determining the optimal size of the sliding window, which is also difficult. Consequently, for automatic, large-scale analysis of degradation, where immense performance metrics are involved, sophisticated techniques like Wavelet and SPWVD are infeasible to provide P and L for MKSK because they are commonly used in ad-hoc ways, and require manual aided configuration in most cases.

D. Oversimplified Characterization of Degradation Process

Sen's Slope Estimator, and Seasonal Sen's Slope Estimator actually estimate the slope of the assumed linear trend. However, such a linear trend is too oversimplified to characterize the degradation process in detail towards more in-depth analysis. It cannot provide diagnostic knowledge to help answer questions such as the following three.

1) When does system performance start to degrade? This question is important because the exceptional events around this time point should be more carefully inspected.
2) Does the degradation trend of Throughput resemble the degradation trend detected over the Mean Latency of SQL Query? Do they have causal relations?
3) What time-frame does degradation exhibit most serious effect on system QoS? This question is important because the log messages recorded within this time-frame should most importantly be watched. But if the estimated degradation trend is nonlinear, we can answer all the questions above.

IV. CSHP Framework

A. Introduction to Modified Cox-Stuart Test

In [20], Cox and Stuart introduced a trend test based on the sign test. The rationale of this test is 1) a series of observations is said to exhibit an upward trend if the later observations tend to be larger than the earlier observations, and 2) a series of observations is said to exhibit a downtrend if the earlier observations tend to be larger than the later observations. The routine of the standard Cox-Stuart test follows three steps.

1) Pair the i-th observation of the first half with the i-th observation of the second half of the time series tested.
2) Carry out a sign test based on the "+" or "−" computed from each pair.
3) Via the sign test, a downtrend is detected if the first half has significantly bigger s-medians than the second half, and an uptrend is detected if the first half has significantly smaller s-medians than the second half.

The Cox-Stuart test is widely used, but cannot be applied to time series with periodic components. We modify the standard routine of the Cox-Stuart test, shown in Fig. 3.

Modified Cox-Stuart Test

$T= [T_1,T_2,...,T_n]$ is the input time series for degrading trend.
α is the significance level for detecting degrading trend

1: *Divide T into two halves:*
If n is an even number,
$T_{first}=[T_1,T_2,...,T_{n/2}]$, $T_{second}=[T_{n/2+1},T_{n/2+2},...,T_n]$, $n^*=n/2$;
If n is an odd number,
$T_{first}=[T_1,T_2,...,T_{(n-1)/2}]$, $T_{second}=[T_{(n+1)/2},T_{(n+1)/2+1},...,T_n]$, $n^*=(n-1)/2$.

2: *If n is an even number, pair each element T_i (i=1, 2... n/2) in T_{first} with $T_{i+n/2}$ in T_{second} :*
$(T_1,T_{n/2+1}), (T_2,T_{n/2+2}),..., (T_{n/2},T_n)$;
If n is an odd number, pair each element T_i (i=1, 2... (n-1)/2) in T_{first} with $T_{i+(n-1)/2}$ in T_{second} :
$(T_1,T_{(n+1)/2}), (T_2,T_{(n+1)/2+1}),..., (T_{(n-1)/2},T_n)$.

3: *For the ith pair, let T_{i1} represents the first element and T_{i2} represents the second element then compute $S_i=sgn(T_{i2}-T_{i1})$ for each pair, where i=1,2,...,n/2 when n is an even number and i=1,2,..., (n-1)/2 when n is an odd number.*

4. *Synthetize all S_i into a cumulative statistic $S=\sum S_i$.*

5: *Transform S into a normalized statistic z, which follows N(0,1):*
$z=(S+1)/sqrt(n/2)$.

6: *Compute one sided p-value of z according to cumulative distribution function of standard normal distribution.*

7: *If p-value<α, degrading trend is detected, else there is no degrading trend.*

Fig. 3. Modified Cox-Stuart test.

B. Introduction to Iterative Hodrick-Prescott Filter

Hodrick-Prescott filter (HPF) [21] is introduced in this subsection to improve degradation trend estimation. Unlike Sen's (or Seasonal Sen's) Slope Estimator assuming linear trend of time series, HPF estimates trend in a data-driven and unstructured manner, with no presumed trend structures. Thus, when the degradation process exhibits nonlinear patterns, HPF can characterize the nonlinear dynamics accurately. Furthermore, HPF is insusceptible to periodic behaviors of input time series. The principles of HPF are introduced as below.

$$Input: Y = [y_1, y_2, \ldots, y_n]\ Output: X = [x_1, x_2, \ldots, x_n] \tag{1}$$

$\{y_t\}$ represents the raw time series, and $\{x_t\}$ represents the trend component estimated from $\{y_t\}$. HPF can extract the trend component, completely insensitive to whether periodic patterns are present or not within raw time series. The trend component $\{x_t\}$ is the optimal solution of the following dynamic programming problem.

$$\min\{\sum_{t=1}^{n}(y_t - x_t)^2 + \lambda \sum_{t=2}^{n-1}(x_{t-1} - 2x_t + x_{t+1})^2\}. \tag{2}$$

Utilizing a least squares method, the trend can be estimated as

$$X = (IN + \lambda M^T M)^{-1} Y,$$

$$M = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1-2 & 1 & \end{bmatrix}. \quad (3)$$

In fact, the trend component extracted by HPF is a penalized spline model, fitting raw time series to be a more smoothened representation. Parameter λ is the smooth parameter, which penalizes the variability of the trend component, controlling the trade-off between the goodness of fit and smoothness of X. The larger the value of λ is, the smoother the trend component is. In the extreme case, when λ approaches infinity, the limit of X is the affine fit of Y. Adaptively determining the optimal value of λ is critical to the quality of trend estimation. Schlicht [22], and Dermoune [26] provide consistent estimators of optimal λ , while unfortunately the λ calculated from the estimators are usually suboptimal. In our practical use, the λ estimators often yield a λ too small to obtain a relatively smooth trend estimation. Therefore, we design an iterative filtering procedure with λ initialized by Dermoune's estimator as λ_0. We double λ in each loop, and repeat until the change in the smoothness of the estimated degradation trend is negligible. The smoothness in each iteration is measured by the 90th percentile of an array V, where V contains all of the second order difference of X (equals $M * X$). We provide a Cauchy-type stopping criterion:

$$\frac{|Q_{90}[V(t+1)] - Q_{90}[V(t)]|}{|Q_{90}[V(t)]|} < Cauchy_{stop}. \quad (4)$$

$Q_{90}(\cdot)$ means the 90th percentile, $V(t)$ represents the second order difference of the trend component X in the i-th loop, and $Cauchy_{stop}$ is set to 0.0005. This is a common value for a Cauchy-type stopping criterion. The maximum number of iterations is set to be 50 because the λ at the 50th iteration (equals to $\lambda_0 * 2^{50}$) approximately equals infinity, which will be definitely larger than the optimal λ. In such an enumerative way, a quasi-optimal λ can be deduced. Finally, we name our modified filter *Iterative Hodrick-Prescott filter (IHPF)*, an adaptive filter that can automatically extract the degradition trend, with no need for assumptions of the structure and the periodic patterns of the performance time series.

V. Monte Carlo Experiment for Evaluation of CSHP and MKSK

A. Evaluation Methodology

1) 1) Evaluation Metrics: The power of a statistical test is the probability that the test will reject the null hypothesis when the null hypothesis is false (i.e., True Positive Rate). In statistical tests for trend, the null hypothesis and the alternative hypothesis are always as follows.

$$H_0 : The\ tested\ time\ series\ shows\ no\ trend$$
$$H_1 : The\ tested\ time\ series\ shows\ a\ significant\ trend. \quad (5)$$

Thus, the *Trend Test Power* (TTP) is the statistical power of a statistical trend test at detecting the trend present in a time series. We make TTP the *first* metric to evaluate and compare the performance of degradation detection. In the next subsection, we will estimate the TTP of our modified Cox-Stuart test, and the Mann-Kendall (or Seasonal Kendall) Test, via Monte Carlo simulations. For either trend test, after testing a batch of synthesized time series, a TTP instance can be estimated as

$$TTP = \frac{N_{reject}}{N_{trend}}. \quad (6)$$

N_{trend} is the total number of separate time series with a trend. And N_{reject} is the total number of separate time series upon which the null hypothesis has been rejected.

Similarly, we can estimate an instance of *False Positive Rate (FPR)*, which is the *Type I* error rate, by

$$FPR = 1 - \frac{N_{accepted}}{N_{notrend}}. \quad (7)$$

$N_{notrend}$ is the total number of time series which actually include no trend. N_{accept} is the number of times that the null hypothesis was accepted. For both our modified Cox-Stuart test in CSHP, and the Mann-Kendall (or Seasonal Kendall) Test in MKSK, a higher TTP and a lower FPR mean superior performance in degradation detection.

Finally, to evaluate the performance of quantitative degradation characterization, we actually evaluate the accuracy of reconstructing degradation trends from raw time series, which is contaminated by noise and multiple periodic components. During Monte Carlo experiments, we can compute the *Sum of Squared Error* (*SSE*) between the degradation trend estimated and the original degradation trend recorded when the time series is synthesized. The smaller the SSE, the better a degradation progress is characterized.

$$SSE = \sum_{i=1}^{M} |T_t - T_i^*|^2. \quad (8)$$

In (8), T_i* represents the original degradation trend synthesized into the *i-th* time series. T_i represents the degradation trend estimated from this time series. M represents the total number of time series.

2) 2) Evaluation Targets: As we have specified in Section III, the performance (i.e., TTP, FPR, and SSE) of MKSK also relies on the correctness of two additional input parameters P and L. To guarantee the comprehensiveness of evaluation and comparison between CSHP and MKSK, we incorporate three different configurations of MKSK, which respectively utilize different schemes to achieve P and L preliminarily. Totally, we have four targets for performance evaluation and comparison, listed below.

CSHP: With no need for input parameters P and L, CSHP is insensitive to complex periodic patterns of input time series for degradation detection and quantitative characterization.

Ideal-MKSK: The correct values of P and L are manually provided to MKSK.

Random-MKSK: Random values are assigned to P and L each time MKSK is executed. The values of P and L are respectively sampled from stochastic distributions. More specifi-

cally, P follows a discrete 0–1 distribution with $Pr\{P=1\} = Pr\{P=0\} = 0.5$, and L follows a discrete uniform distribution between $[2, N/2]$, where N represents the total number of observations within a single input time series.

Fourier-MKSK: P and L are both estimated via a preliminary periodicity analysis procedure, i.e., harmonic analysis based on Fourier Power Spectral Density [24], [31].

B. Monte Carlo Experiments Design

1) 1) Models for Generating Performance (or Resource Usage) Time Series: We use a general additive model to generate simulated time series:

$$\begin{aligned} Y &= T + C + E \\ Y_t &= T_t + C_t + E_t (t = 1, 2, \ldots N). \end{aligned} \tag{9}$$

It is a universal model widely used for time series synthesis and decomposition. Y is composed of three components $T, C, and\ E$.T represents the trend component (i.e., the simulated degradation trend), C represents the periodic component, and E utilizes a Moving Average (MA) stochastic process to construct the noise component. We generate $T, C, and\ E$ according to diverse models. In particular, the generated periodic component and the noise component are both trend-free. Hence, whether a synthetic time series Y shows a trend is completely determined by the trend component T.

To not be biased by one certain type of degradation trend, we use 5 types of trends to generate performance time series. Trend component T can be classified into two categories: *Trend-Free*, and *Non-Trend-Free*. The former can be used to estimate the FPR of a certain trend test, and the later can be used to estimate TTP. The types of trends and their generating models (10)∼ (14) are listed below.

The *Moving Average* trend model of T:

$$T_{t+1} = \epsilon_{t+1} + 0.1\epsilon_t (t = 0, 1, \cdots N-1). \tag{10}$$

The *Linear* trend model of T:

$$T_t = \alpha t + 0.1 (t = 0, 1, \ldots N-1). \tag{11}$$

The *Quadratic* trend model of T:

$$T_t = \frac{\alpha t^2}{N-1} (t = 0, 1, \ldots N-1). \tag{12}$$

The *Exponential* trend model of T:

$$T_t = \frac{\alpha(N-1)}{(e^{(N-1)/10} - 1)} (e^{t/10} - 1)(t = 0, 1, \ldots N-1). \tag{13}$$

The *Sigmoid* trend model of T:

$$T_t = \frac{\alpha(N-1)}{1 + e^{-10/N(t-N/2)}} (t = 0, 1, \ldots N-1). \tag{14}$$

In (10)–(14), ϵ is a randomly generated white noise series. α is a variable controlling the strength of trend. We call α the *Trend Strength*. In the following content, we will change α to evaluate the TTP and FPR on time series with trends in different intensities. All types of trend components except Moving Average are rescaled to the same range $[0, \alpha(N-1)]$. This rescaling ensures that the TTPs of different types of trends are comparable, because all of the trend components on average increase from 0 to $\alpha(N-1)$ over a time range from 1 to N. That is to say, they all get an approximately equal trend strength, on average $\alpha(N-1)/N$.

For periodic component C, to simulate the complex oscillating patterns in a real-world time series, we also design different types of simulated periodic components; their generating models (i.e., (15)–(19)) are listed below.

The *Non-periodic* model of periodic component C:

$$C_t = 0 (t = 0, 1, \ldots N-1). \tag{15}$$

The *Sinusoidal* model of periodic component C:

$$C_t = \frac{A}{2} \sin(\frac{2\pi}{k} x) (t = 0, 1, \ldots N-1). \tag{16}$$

The *Unimodal* model of periodic component C:

$$C_t = A|\sin(\frac{\pi}{k} x)| (t = 0, 1, \ldots N-1). \tag{17}$$

The *Bimodal* model of periodic component C:

$$\begin{aligned} C_t = {} & A|\sin(\frac{\pi}{k} x)| \\ & + A|\sin(\frac{2\pi}{k} x + \frac{\pi}{10})| (t = 0, 1, \ldots N-1). \end{aligned} \tag{18}$$

The *Multimodal* model of periodic component C:

$$\begin{aligned} C_t = {} & A|\sin(\frac{\pi}{k} x)| + A|\sin(\frac{2\pi}{k} x + \frac{\pi}{10})| \\ & + A|\sin(\frac{4\pi}{k} x + \frac{\pi}{10})|, (t = 0, 1, \ldots, N-1). \end{aligned} \tag{19}$$

In (15)–(19), A is a variable controlling the Oscillation Intensity of the periodic component in the generated time series. For the noise component, we use a *MA(5)* process model:

$$E_t = \mu + \frac{\beta}{5} \sum_{j=1}^{5} \theta_j \epsilon_{t-j} (t = 0, 1, \ldots N-1) \tag{20}$$

ϵ represents a randomly generated white noise time series. β is a parameter controlling the intensity of noise, we call it Noise Strength. By changing β in the next two subsections, we can evaluate whether the TTP and FPR of a trend test is sensitive to noise intensity. Parameter μ is the mean of this *MA(5)* process, which is always set to zero in the Monte Carlo experiments. Parameter θ_j is a random variable following a continuous uniform distribution between 0 and 1.

2) 2) Experimental Plan: To evaluate CSHP and MKSK in an objective way, we are required to carry out evaluation over adequate and heterogeneous time series. To maximize heterogeneity, we select 6 heterogeneity factors (as much as we can imagine, cf. Table I), and design five levels (cf. Table II) for each factor. As a result, we are able to generate 15,625 (combinations of all factors and all levels) distinct types of simulated time series. On the other hand, for each type of time series, we generate 1,000 homogeneous samples to constitute a group. The

TABLE I
HETEROGENEITY FACTORS

Number	Heterogeneity factor	Corresponding Parameter	Official Notation
1	Trend strength	α	Trend Strength
2	Time series length	N	Series Length
3	Style of trend	Trend Type	Trend Type
4	Style of period	Period Type	Period Type
5	Oscillation intensity of periods	A	Period Amplitude
6	Noise strength	β	Noise Strength

TABLE II
LEVELS OF EACH HETEROGENEITY FACTOR

Level	Trend Strength	Length	Trend Type	Period Type	Amplitude	Noise
1	0.001	60	Moving Average	Non-periodic	2	0.01
2	0.003	80	Linear	Sinusoidal	4	0.03
3	0.005	100	Quadratic	Unimodal	6	0.05
4	0.007	120	Exponential	Bimodal	8	0.07
5	0.009	140	Sigmoid	Multimodal	10	0.09

samples within the same group are identical, except for the difference induced by the random noise component. In total, we generate 15,625,000 simulated time series samples (i.e., 15,625 sample groups, each group with 1,000 samples). The samples are heterogeneous among different groups, while homogenous within the same group.

For each of the four targets, i.e., CSHP, Ideal-MKSK, Random-MKSK, and Fourier-MKSK, we execute the modified Cox Stuart test or the Mann-Kendall (or Seasonal Kendall) Test on every sample group under a significance level 0.05. When the 1,000 time series within a certain group are trend-free (i.e., having a Moving Average style trend), we estimate a FPR instances on this group. When the time series in a group are not trend-free, we estimate a TTP instance. Finally, we got 3,215 instances of FPR, and 12,500 instances of TTP for each evaluated target. For SSE, we execute Sen's (or Seasonal Sen's) Slope estimator or our IHPF on each group, and finally we get 15,625 SSE instances for each target.

C. Evaluation of Trend Test Power

We apply a box-whisker plot (Fig. 4) to compare the detection power (over degradation trends) of the four targets. The box-whisker plot utilizes Minimum, Lower quartile (25th percentile), Median (50th percentile), Higher Quartile (75th percentile), Maximum, and s-Mean to quantify the difference between two sample sets, which is more comprehensive and unbiased than just a single metric (e.g., median or mean). Hence, we define an average measure $Diff$ to integrate multiple statistical metrics of a box-whisker plot, where A and B are two sets of TTP, FPR, or SSE instances; Q_{low} means the lower quartile, and Q_{high} means the higher quartile. See equation (21) at the bottom of the page.

1) 1) Verifying and Quantifying the Influence of P and L Over TTP: From Fig. 4, we can quantify the difference of degradation detection power between Ideal-MKSK and Random-MKSK, with $\mathrm{Diff}_{\mathrm{TTP}}$ (Ideal-MKSK, Random-MKSK)$=0.3796$. This difference verifies that the detection power of MKSK is highly susceptible to the correctness of the two prerequisite procedures, i.e., periodicity test (P), and period length estimation (L). *This result answers Question 3*. In practical application of MKSK, if we cannot guarantee absolute accuracy of the preliminary periodicity analysis (viz, P and L) of MKSK, the degradation detection power may be severely deflated by as much as 37.96% (relative to perfect detection power 1.0).

2) 2) Unreliable Fourier Analysis: From Fig. 4, we can also evaluate the degradation detection power of Fourier-MKSK, $\mathrm{Diff}_{\mathrm{TTP}}$ (Ideal-MKSK, Fourier-MKSK)$=0.2896$, and $\mathrm{Diff}_{\mathrm{TTP}}$ (Fourier-MKSK, Random-MKSK)$=0.1090$. The degradation detection power of Fourier-MKSK is inferior to Ideal-MKSK (28.96% power reduction, relative to perfect

$$Diff(A,B)=\frac{\{[Q_{low}(A)-Q_{low}(B)]+[median(A)-median(B)]+[Q_{high}(A)-Q_{high}(B)]+3[mean(A)-mean(B)]\}}{6} \quad (21)$$

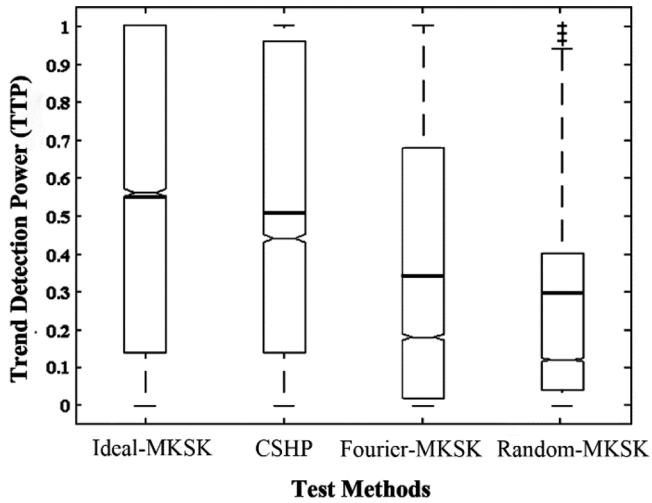

Fig. 4. Box-Whisker plot for TTP comparison in degradation detection, where the heavy black solid line represents the s-mean value.

TABLE III
ANOVA OF TTP FOR IDEAL-MKSK

Source of Variance	Sum of Squares	Degrees of Freedom	Mean of Squares	Effect Size (η2)
Trend Strength	572.021	4	143.005	40.2%
Series Length	315.156	4	78.789	22.1%
Trend Type	2.398	3	0.799	0.2%
Period Type	0.945	4	0.236	0.0%
Period Amplitude	0.000768	4	0.000192	0.0%
Noise Strength	234.693	4	58.673	16.5%
Residual Error	298.714	12476	0.0239	21.0%
Total	1423.928	12499		

power 1.0). This result verifies that Fourier analysis is unreliable, and likely to degrade the detection power of MKSK, due to incorrect periodicity test or inaccurate period length estimation. Among the simulated performance time series, some of them are remarkably non-stationary with strong trends (time series with large value of α), some with complex interlaced and non-sinusoid shaped oscillation components (e.g., time series with Multimodal shaped periodic components), and some with strong noise. In such situations, Fourier analysis provides distorted P and L to MKSK, which remarkably deflates the detection power. Therefore, although Fourier analysis is the most feasible preliminary procedure to automate MKSK, it is highly problematic in practical degradation detection. *This result answers Question 4*.

However, the degradation detection power of Fourier-MKSK is higher than Random-MKSK (10.9% higher, relative to perfect power 1.0), which means that even if Fourier analysis induces a negative impact on MKSK, it works in few cases. It still works for a small fraction of time series that are approximately stationary, and with periodic components resembling sinusoid-like waves.

3) 3) Evaluation of TTP for CSHP: From Fig. 4, we can evaluate the degradation detection power of CSHP from $\mathrm{Diff}_{\mathrm{TTP}}$ (Ideal-MKSK, CSHP)$= 0.0572$, and $\mathrm{Diff}_{\mathrm{TTP}}$ (CSHP, Fourier-MKSK)$= 0.2324$. The detection power of CSHP approaches Ideal-MKSK with a narrow gap 5.72%, which means CSHP can approach the upper-limit performance of MKSK with no need for human-assisted knowledge about the periodicity properties of the performance time series. In contrast, MKSK can only achieve this power level when it is manually provided with always correct values of P and L. Moreover, CSHP remarkably outperforms Fourier-MKSK with a huge gap 23.24%, which means CSHP is remarkably superior to the most feasible automatic version of MKSK.

4) 4) CSHP is Insusceptible to Periodic Patterns and Robust to Noise: Analysis of Variance (ANOVA) techniques can determine the individual effects of factors that affect the response variable. In this subsection, the TTP of CSHP is treated as the response variable, and the affecting factors are Period Type, Period Amplitude, Noise Strength, Trend Type, Series Length, and Trend Strength (cf. Table I). The TTP instances of CSHP enable a 6-way ANOVA with five levels for each factor. However, Moving Average is not used for estimating TTP (used for FPR), thus as an exception there are not five but four levels for the factor Trend Type. We also execute an identical ANOVA procedure on the TTP instances of Ideal-MKSK for comparison. The result of Ideal-MKSK is shown in Table III, and the result of CSHP is shown in Table IV. The last column of the two tables is the *Effect Size* $(\eta)^2$ of each factor. It measures the relative magnitude of impact induced by a factor towards the response variable. Jacob Cohen [32] offers a rule for interpreting η^2, where 0.0099 constitutes a small effect, 0.0588 a medium effect, and 0.1379 a large effect.

From the results, we can conclude two key points. 1) According to Cohen's rule, Period Type, and Period Amplitude have negligible effects on CSHP. This result supports that CSHP is intrinsically invulnerable to periodic behaviors of performance time series, in degradation detection. Seemingly, Ideal MKSK also achieves this merit. However, it is a facade, just due to the human-assisted knowledge of periodicity. 2) CSHP is more robust to noise than MKSK. According to Cohen's rule, Noise Strength, with an effect size of 16.5%, exhibits a large effect on the TTP of Ideal-MKSK. But for CSHP, Noise Strength constitutes only a medium effect, with an effect size of 8.7%.

D. Evaluation of False Positive Rate (FPR)

All FPR instances for the four targets are integrated into a box-whisker plot (cf. Fig. 5). CSHP exhibits remarkably lower FPR than MKSK. $\mathrm{Diff}_{\mathrm{FPR}}$ (Ideal-MKSK, CSHP)$= 0.0181$, $\mathrm{Diff}_{\mathrm{FPR}}$ (Fourier-MKSK, CSHP) $= 0.0117$, $\mathrm{Diff}_{\mathrm{FPR}}$ (Random-MKSK, CSHP)$= 0.0116$. Relative to the s-mean FPR of Ideal-MKSK, i.e., 0.05, CSHP reduces 37.7%. Further, CSHP exhibits more stable FPR than MKSK, which is generally bounded by the significance level 0.05

TABLE IV
ANOVA OF TTP FOR CSHP

Source of Variance	Sum of Squares	Degrees of Freedom	Mean of Squares	Effect Size (η2)
Trend Strength	486.956	4	121.739	43.3%
Series Length	267.499	4	66.875	23.7%
Trend Type	3.660	3	1.220	0.3%
Period Type	0.0139	4	0.00348	0.0%
Periods Amplitude	0.0154	4	0.00385	0.0%
Noise Strength	97.639	4	24.410	8.7%
Residual Error	269.529	12476	0.0216	24.0%
Total	1125.314	12499		

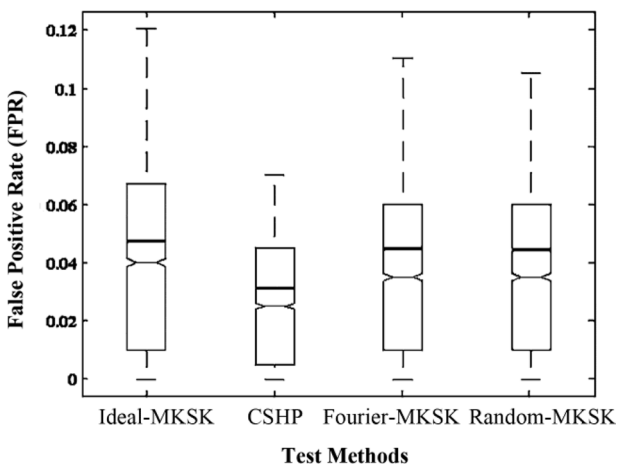

Fig. 5. Box-Whisker plot for FPR comparison in degradation detection.

(i.e., the expected FPR upper-limit). In contrast, the FPRs of Ideal-MKSK, Fourier-MKSK, and Random-MKSK are highly uncontrollable, respectively with 53.7%, 51.2%, and 52.8% of FRP instances beyond the expected limit.

The FPR of Ideal-MKSK, Fourier-MKSK, and Random-MKSK are approximately within the same level, with Ideal-MKSK slightly higher. This result means that, though incorrect periodicity test and inaccurate period estimation cause severe detection power loss, they do not have much effect on FPR. That result is true because in all statistical tests, including either the Mann-Kendall Test or the Seasonal Kendall Test, controlling false alarms is the foremost consideration.

E. Evaluation of Quantitative Degradation Characterization

1) 1) Comparison of SSE: From Fig. 6, Ideal-MKSK exhibits remarkably lower errors in degradation trend estimation than Random-MKSK, with $\mathrm{Diff}_{\mathrm{SSE}}$ (Ideal-MKSK, Random-MKSK) $= -1.85$. A distorted P or L may incur

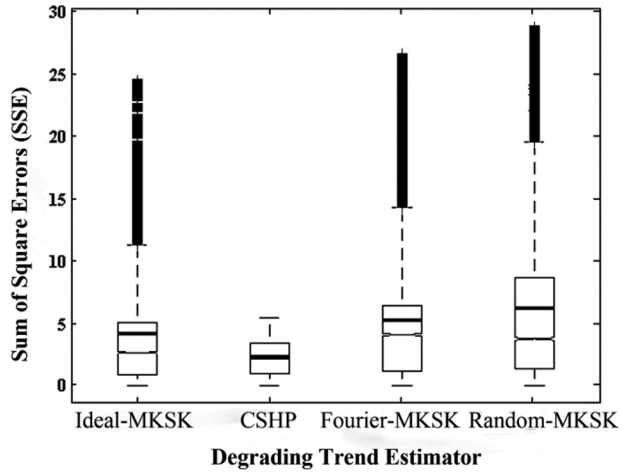

Fig. 6. Box-Whisker plot for SSE comparison in quantitative degradation characterization.

up to 44.17% loss of estimation accuracy, compared with the s-mean SSE of Ideal-MKSK. Hence, the estimation accuracy of MKSK upon performance degradation trends is highly susceptible to the preliminary periodicity analysis as well, strengthening the conclusions of TTP in Section V-C. *This result is also a complementary answer to Question 3*. The estimation accuracy of Fourier-MKSK is remarkably inferior to Ideal-MKSK (relatively 24.81%), and Fourier-MKSK is more likely to incur exceptionally large errors. This result means that applying Fourier analysis as a preliminary procedure for MKSK is highly problematic in degradation trend estimation, which reinforces the conclusions of TTP. *This result is also a complementary answer to Question 4*.

The SSE of CSHP is remarkably lower than MKSK, with $\mathrm{Diff}_{\mathrm{SSE}}$ (CSHP, Ideal-MKSK) $= -1.29$, $\mathrm{Diff}_{\mathrm{SSE}}$ (CSHP, Fourier-MKSK) $= -2.34$, and $\mathrm{Diff}_{\mathrm{SSE}}$ (CSHP, Random-MKSK) $= -3.15$. Beyond that, CSHP is more stable in controlling outliers of SSE. If we set 10 as an expected SSE upper-bound, then the percents of exceptional errors are 0.00%, 11.03%, 15.06%, and 21.38%, respectively for CSHP, Ideal-MKSK, Fourier-MKSK, and Random-MKSK. To sum up, MKSK is incapable at characterizing the process of degradation in an accurate way. In contrast, IHPF in CSHP can estimate degradation trends with low errors.

2) 2) Linear and Nonlinear Trend Estimation: Similar to the two ANOVA experiments of the TTP in Section V-C, two more are carried out in this subsection, respectively with response variable SSE of CSHP, and SSE of Ideal-MKSK. The result tables are omitted due to limited space. Trend Type (cf. Table II) is the factor having most significant effect on the SSE of MKSK with an effect size of 19.8%, while for CSHP the effect size of Trend Type is negligible (0.136%). To delve into the effects of Trend Type, we respectively divide the 15,625 SSE instances of MKSK and CSHP into 5 groups, grouped by trend types. Then we recreate an integrated box-whisker plot to compare them (cf. Fig. 7). For MKSK, the groups with quadratic, exponential, and sigmoid trends have remarkably larger SSE than the groups with linear trend and no trend. Furthermore, the nonlinear groups of MKSK show a lot of SSE outliers, while the linear groups don't.

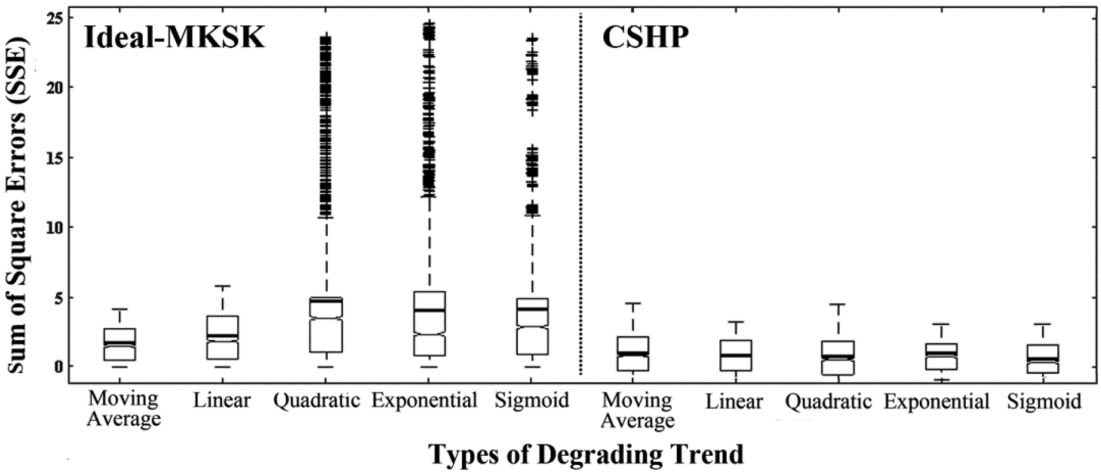

Fig. 7. Comparison of SSE over different types of degradation trends for Ideal-MKSK and CSHP.

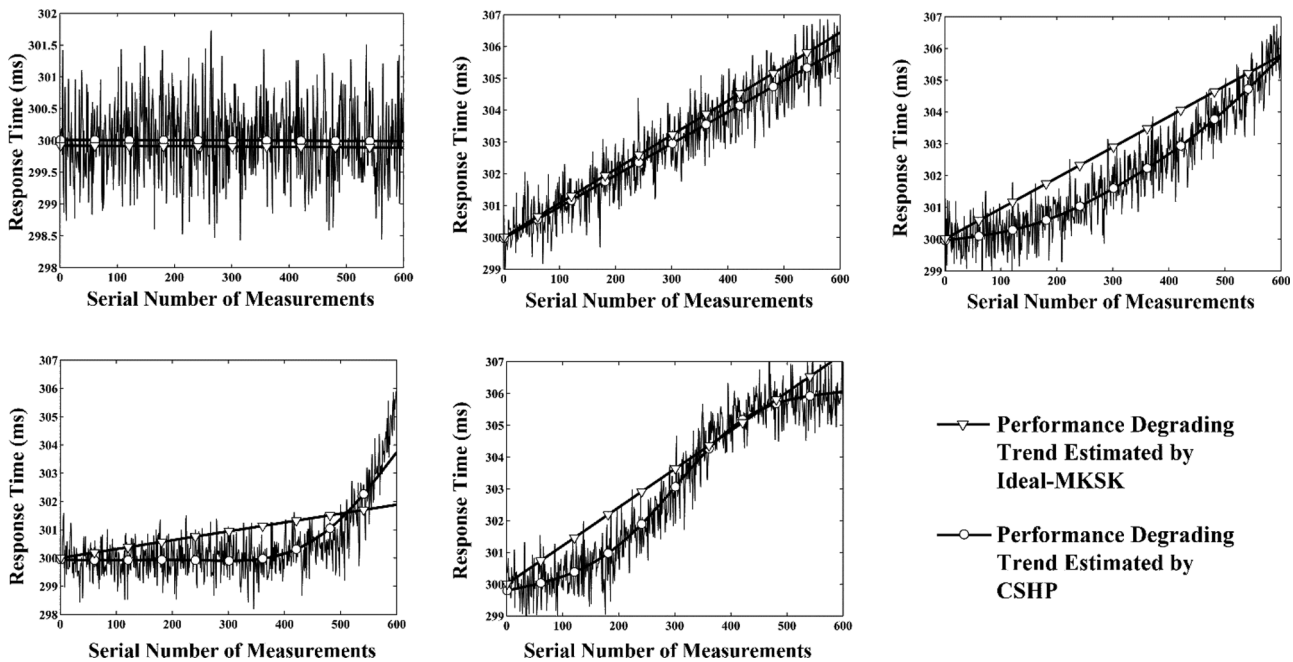

Fig. 8. Comparison of degradation trends estimated by Ideal-MKSK and CSHP on time series without periodic patterns.

The result confirms that MKSK only works for linear degradation trend estimation. In contrast, no matter whether it is a linear trend or a nonlinear trend, CSHP can estimate the trend with a small SSE. This result confirms that CSHP could characterize the nonlinear patterns of performance degradation in an adaptive way. We provide some examples to illustrate degradation trends estimated by CSHP and Ideal-MKSK in Fig. 8.

The effect size of Noise Strength for MKSK is 3.19%, and just 0.67% for CSHP. This result verifies that MKSK is more sensitive to noise than CSHP, reinforcing the conclusions in Section V-C about the TTP and noise. Period Type, and Period Amplitude both exhibit negligible effects on CSHP, which verifies that CSHP is invulnerable to the complex periodic behaviors of performance time series (examples in Fig. 9). To sum up, CSHP is superior to MKSK in quantitatively characterizing degradation.

F. Characterization of Multi-Stage, and Multi-Pattern Degradation Processes

1) 1) Multi-Stage Degradation: Commonly, degradation is an intermittent process with multiple discrete stages, because degradation-related anomalies are occasionally activated by uncertain conditions. For instance, an abnormal service thread, which repeatedly leaks temporary message buffers during execution, can keep depleting memory only when it is scheduled.

2) 2) Multi-Pattern Degradation: The pattern (e.g., shape) of the degradation trend for a certain degradation process may not be monotonous. Exposed to different levels of workload intensities, the same degradation problem can exhibit different orders of severity, and thus can incur different patterns of degradation trend. Moreover, the root causes of degradation phenomena are diverse, also causing various types of degradation trend.

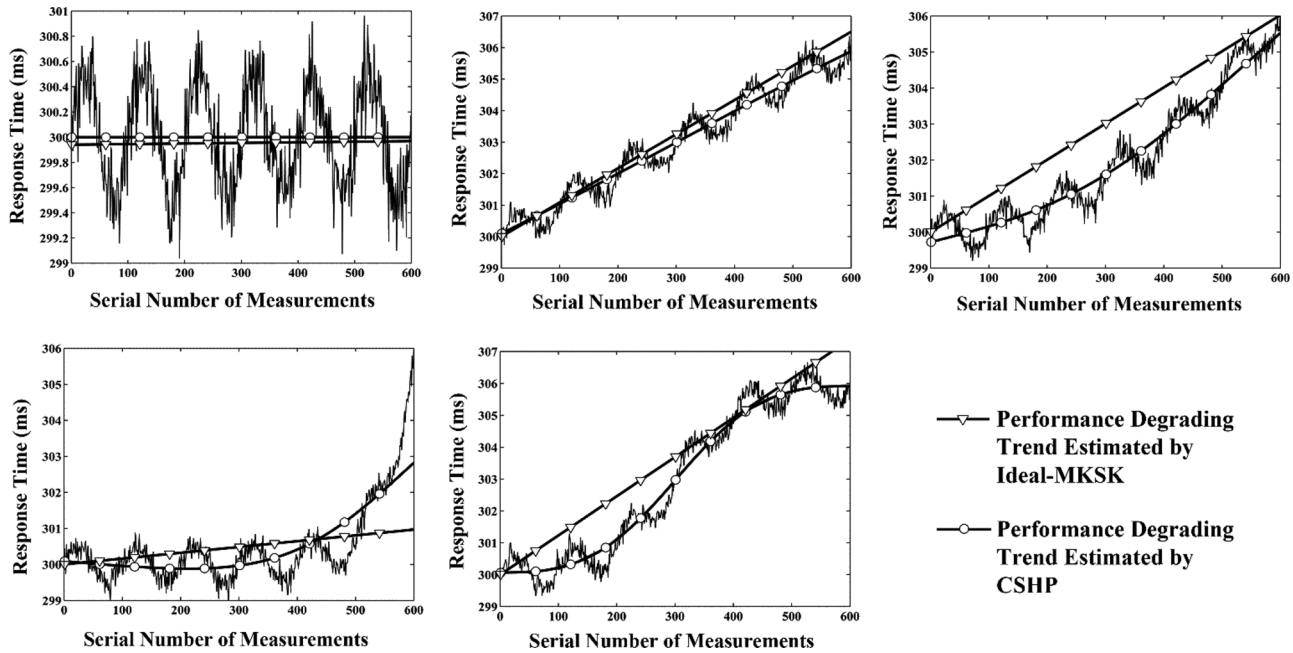

Fig. 9. Comparison of degradation trends estimated by Ideal-MKSK and CSHP on time series with periodic patterns.

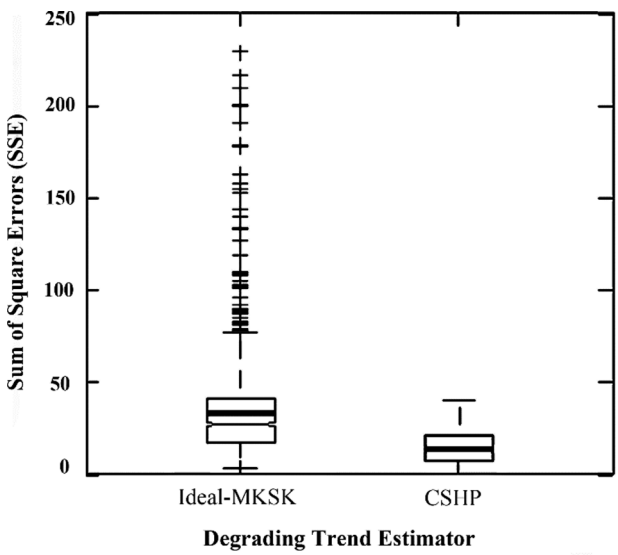

Fig. 10. Comparison of SSE over various types of degradation trends.

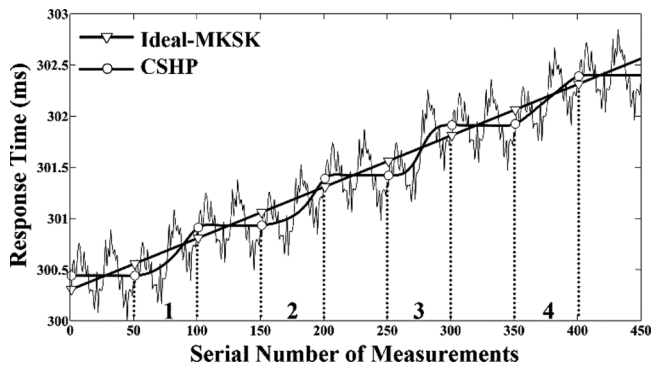

Fig. 11. Examples for comparison of CSHP and MKSK in characterizing periodic degradation process with multiple stages and multiple patterns, 1 is quadratic, 2 is exponential, 3 is sigmoid, and 4 is linear.

Hence, we incorporate these two complexities to further evaluate the performance of CSHP in degradation trend estimation. To simulate Multi-stage degradation, we concatenate multiple degradation processes within a single simulated time series. Each sub-process of degradation is separated from one another, and a non-degradation process (trend-free process) is interposed between. To simulate Multi-pattern degradation, we simultaneously incorporate five types of degradation trend within a single simulated time series. Each trend type corresponds to a certain sub-process of degradation described above. To sum up, for a single simulated time series, in total, 9 sub-processes are involved, where four are degradation sub-processes with distinct trend types (Linear, Quadratic, Exponential, and Sigmoid), and the other five are non-degradation sub-processes (all with trend type Moving Average).

We carry out Monte Carlo experiments similar to those of Section V-E, except that Trend Type is not a heterogeneous factor any longer, and the length of the time series are increased due to degradation process concatenation. The numerical results are illustrated in Fig. 10, which verifies the superiority of CSHP in characterizing more complicated degradation processes. An example is illustrated in Fig. 11, from which we can realize that 1) CSHP accurately characterizes each degradation phase no matter what types of degradation trend each phase exhibits; 2) CSHP can reflect the onset of each degradation stage with a relatively high resolution, which is very important for degradation diagnosis and log analysis (*which answers the three questions we have mentioned in* Section III-D); and 3) in comparison, MKSK is oversimplified and, incapable to provide degradation details for further analysis.

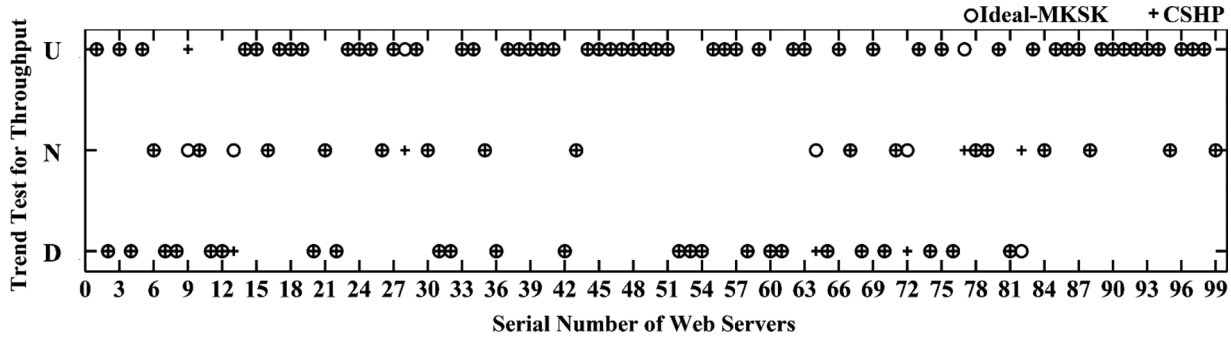

Fig. 12. Detection of degradation trends over daily throughput of 99 web servers.

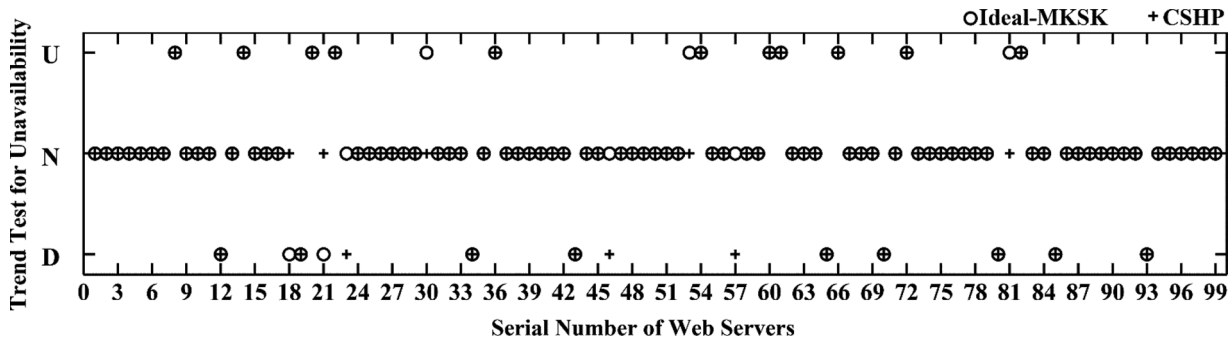

Fig. 13. Detection of degradation trends over daily unavailability of 99 web servers.

VI. Evaluation of CSHP on a Real Dataset

A. Degradation Trend Detection on Real Data

The dataset in [23] is a collection of probe traces, where the probes are sent from a client machine (Intel 600MHZ and Linux) in CMU-PDL to 99 public web servers all around the world. The servers include a variety of domains, such as com, edu, gov, and org. The goal of these probes is to monitor the availability and throughput of web servers in the long term (about 3 months in fact). Each probe carries several points of information: 1) whether the target server is available (whether the homepage is valid), 2) the time elapsed to load the homepage, and 3) the amount of bytes transmitted within this probe. The probe is launched every 10 minutes. In this simple manner, the service availability and the throughput (loading time divided by bytes transmitted) of web servers are periodically measured and collected into time series.

Unlike previous simulation experiments, the intrinsic trend components of real-world throughput and unavailability time series are unachievable. Thus TTP and FPR cannot be estimated for evaluation and comparison any more. In this situation, we used Ideal-MKSK (i.e., human-assisted MKSK) as the evaluation baseline because it is widely validated and acknowledged in previous research. The detection results for throughput, and unavailability are respectively shown in Fig. 12, and Fig. 13. The vertical axes represent the detection results: 1) U represents an uptrend detected (e.g., degradation trend for unavailability), 2) N represents no trend detected, and 3) D represents a downtrend detected (e.g., degradation trend for throughput). When a cross and a circle overlap in the figures, it means CSHP yields consistent results with Ideal-MKSK. For the 99 web servers, Ideal-MKSK and CSHP make 92 consistent results in detecting trends of throughput, and 91 consistent results in detecting trend of unavailability. Thus, for degradation detection, the probability CSHP can perform just like human-assisted MKSK is roughly 92%. However, it should be noted that CSHP achieves this idealized performance of MKSK with no need for human assistance.

B. Degradation Trend Estimation on Real Data

In this subsection, we apply CSHP and MKSK to quantitatively characterize the degradation trends of throughput and unavailability. Because the ground-truth trend components of the real-world time series are unachievable, goodness-of-fit measures, i.e., SSE, cannot be utilized for evaluating estimation accuracy any more. Here we utilize an alternative approach. For a trend estimation technique, such as IHP or Sen's (or Seasonal Sen's) Slope estimator, the residual component represents the raw time series, subtracting the estimated trend component. Thus, the more accurately the trend component is estimated, the more trendless the residual will be. In other words, because the trend component of a time series represents the variation of its s-mean value over time, if the trend component is perfectly estimated, the remaining components (i.e., the residual) must include no variation of s-mean at all, only containing trendless fluctuations (e.g., cyclical oscillations or noises) around a fixed value. In this sense, measuring invariability of the s-mean value of the residual can serve as a good approach to measure trend estimation accuracy. Here we adopt a sliding window t-test to measure the invariability of the s-mean for the residual components. The result of the sliding window t-test (between 0 and 1) represents the probability of s-mean-invariability, where 0 indicates the worst estimation accuracy, and 1 indicates the best (i.e., 100% accuracy of estimation).

TABLE V
THEORETICAL TIME COMPLEXITY

Framework	Procedure	Best-case Time Complexity	Average-case Time Complexity	Worst-Case Time Complexity
MKSK	Mann-Kendall Test	$O(n^2)$	$O(n^2)$	$O(n^2)$
	Seasonal Kendall Test	$O(n \cdot log(n))$	$O(n \cdot m)$	$O(n^2)$
	Sen's Slope Estimator	$O(n^2 log(n))$	$O(n^2 log(n))$	$O(n^2 log(n))$
	Seasonal Sen's Slope Estimator	$O(n \cdot log(n))$	$O(n \cdot m \cdot log(n))$	$O(n^2 \cdot log(n))$
CSHP	Modified Cox-Stuart Test	$O(n)$	$O(n)$	$O(n)$
	Iterative Hodrick-Prescott Filter	$O(n)$	$O(n)$	$O(n)$

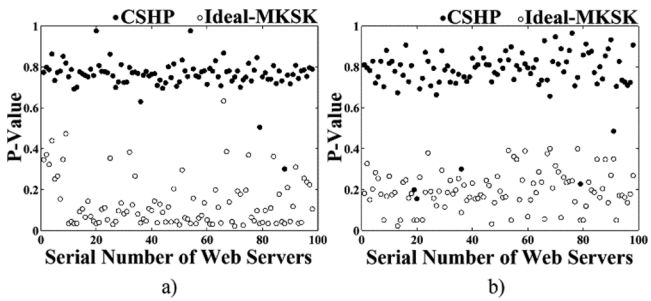

Fig. 14. Sliding window t-test for residual components: a) throughput, average p-value of CSHP: 0.812, MKSK: 0.124; b) unavailability, CSHP: 0.807, MKSK: 0.185.

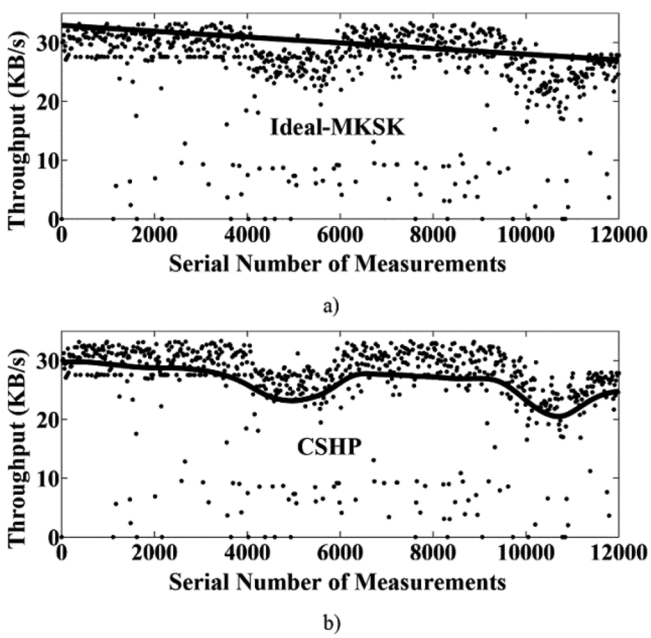

Fig. 15. Real degradation trend of throughput estimated over a random web server for MKSK and CSHP comparison.

We apply CSHP and Ideal-MKSK to estimate the trends of throughput and unavailability for all 99 web servers. Then we carry out a sliding window t-test over the residual components thereof. The results are illustrated in Fig. 14. We can conclude that the residual component of CSHP is approximately trend-free, meaning that the trend component of throughput or unavailability time series is extracted by CSHP accurately. In contrast, the residual component of MKSK is impossibly trend-free, meaning that the trend estimated by MKSK must be highly inaccurate. In summary, upon real-world performance datasets, CSHP is superior to MKSK in degradation trend estimation. This result consolidates the conclusions derived from previous Monte Carlo simulations in Sections V-E and V-F. An example of degradation trends estimated by MKSK and CSHP on this real dataset is illustrated in Fig. 15. Particularly in Fig. 15, the throughput measurements exhibit multiple degradation phases. CSHP can accurately capture such details of the degradation process, while MKSK cannot.

VII. ANALYSIS OF COMPUTATIONAL COMPLEXITY

A. Theoretical Analysis of Computational Complexity

In this subsection, the theoretical time complexity of all constituent parts for MKSK (i.e., Mann-Kendall Test, Seasonal Kendall Test, Sen's Slope Estimator, and Seasonal Sen's Slope Estimator), and CSHP (i.e., modified Cox-Stuart Test, and IHP) are calculated respectively, based on the number of arithmetic instructions (CMP, ADD, SUB, MUL, and DIV) involved. Then each of the complexity expressions is transformed into Big O Notation form (Big O notation [33], also called Landau's symbol, is a symbolism used in complexity theory to describe the asymptotic behavior of functions. Basically, it tells you how fast a function grows or declines.), cf. Table V. n represents the length of the time series, and m represents the number of periods within a time series. The time complexity of the Seasonal Kendall Test and the Seasonal Sen's Slope estimator are both affected by m, where a tiny m (relative to the magnitude of n) represents the best case, a medium m represents the average case, and a large m represents the worst case. The demands for memory space of the six procedures are listed in Table VI.

The modified Cox-Stuart test reduces time complexity to an $O(n)$ level, because its hypothesis test is simpler than that of the Mann-Kendall Test. The time complexity of IHPF is also of $O(n)$ time complexity. Solving the standard HPF (3) is actually solving a system of linear equations. Fortunately, the matrix M is a sparse matrix, and by applying the Gauss Elimination algorithm it can be quickly solved within $O(n)$ level of time complexity. Thus, the total time complexity of IHPF is $O(K^* n)$,

TABLE VI
THEORETICAL SPACE COMPLEXITY

Framework	Procedure	Best-case Space Complexity	Average-case Space Complexity	Worst-case Space Complexity
MKSK	Mann-Kendall Test	*O(1)*	*O(1)*	*O(1)*
	Seasonal Kendall Test	*O(1)*	*O(1)*	*O(1)*
	Sen's Slope Estimator	$O(n^2)$	$O(n^2)$	$O(n^2)$
	Seasonal Sen's Slope Estimator	*O(n)*	$O(n \cdot m)$	$O(n^2)$
CSHP	Modified Cox-Stuart Test	*O(1)*	*O(1)*	*O(1)*
	Iterative Hodrick-Prescott Filter	$O(n^2)$	$O(n^2)$	$O(n^2)$

TABLE VII
EXPERIMENTAL TIME COMPLEXITY

Framework	Procedures	*n*	*n·log(n)*	n^2	$n^2 \cdot log(n)$	n^3	n^4
MKSK	Mann-Kendall Test	2.065	0.5466	*0.3172*	0.3206	0.3188	0.3291
	Seasonal Kendall Test	2.075	0.4612	*0.1198*	0.1204	0.1221	0.1245
	Sen's Slope Estimator	14.16	3.807	3.359	*3.228*	3.242	3.279
	Seasonal Sen's Slope Estimator	8.295	2.469	1.969	*1.9612*	1.937	1.948
CSHP	Modified Cox-Stuart Test	*0.0404*	0.0417	0.0427	0.0429	0.0436	0.0442
	Iterative Hodrick-Prescott Filter	*0.8056*	0.8087	0.8139	0.8247	0.8355	0.8413

where K represents the loops of iterations. The iterations of IHPF are at most 50, because 2^{50} is virtually an infinite maximum for HPF. Actually, in our experiments, the iterations are below 35 in most cases. Therefore K is just a constant compared to n, and the IHPF is in $O(n)$ level of time complexity.

B. Experimental Verification of Computational Complexity

To validate the time complexity for MKSK and CSHP empirically, we utilize a CPU-time profiler to measure the execution time for the six procedures. n is set from 50 to 2000 with a step of 50. For a certain value of n, each procedure is repeated 1000 times, and the average execution times are calculated to represent the time complexity of each procedure (at size n). Moreover, because the worst-case execution time is of particular concern in complexity analysis, we set m large enough (m equals $n/2$). Then polynomial regression (via LMS) is carried out to estimate the polynomial order of time complexity for each procedure. RMSE (Root of Mean Square Error) is utilized as the criterion to select the optimal-fitting order. RMSEs are respectively calculated on the training set (70% points in the front, input size from 100 to 1450), and the validation set (30% points in the end, input size from 1500 to 2000). The final RMSE for each procedure is listed in Table VII, where the optimal polynomial orders are in italic type. The results validate the theoretical analysis in Section VII-A.

The experiment over space complexity validation is also executed. The result also testifies the theoretical analysis in Section VII-A. The experimental results are omitted due to limited page space. The memory demand when n is 2000 is provided: Mann-Kendall Test 56 Bytes, Seasonal-Kendall Test 88 Bytes, Sen's Slope Estimator 15.25 MB, Seasonal Sen's Slope Estimator 7.62 MB, modified Cox-Stuart test 72 Bytes, and IHPF 30.50 MB.

To sum up, the time complexity of CSHP $(O(n))$ is much lower than that of MKSK $(O(n^2 \log(n)))$. Thus, CSHP exhibits highly superior scalability than MKSK in practical applications. The space complexity of CSHP and MKSK are both in quadratic level $O(n^2)$, with a little more memory consumption for CSHP.

VIII. LIMITATIONS AND FUTURE WORK

In fact, the Sen's (or Seasonal Sen's) Slope estimator in MKSK can also be applied as a linear-model to forecast the Time-To-Failure (TTF) or the Time-To-Exhaustion (TTE) of degradation. Specifically, the slope estimated by MKSK is roughly the exhausting rate of available resources, or the decreasing rate of service level. Then the time point of failure (e.g., *Out of Memory* or *Service Level Violation*) can be linearly extrapolated. Aiming at TTE and TTF prediction, in addition to MKSK, Alonso [6] introduces the M5P regression tree; Li [7] introduces the ARMA/ARX model; Cassidy [8] introduces the MSET pattern recognition technique. In this sense, CSHP cannot predict TTE or TTF itself, which is a limitation. Nevertheless, CSHP can server as a useful preprocessing-tool for TTF and TTE prediction. For instance, the models in [6]–[8] must be *ad-hoc adjusted* according to periodic patterns of performance time series, or the prediction accuracy may be largely reduced.

The problem is that accurately anatomizing periodic patterns is not an easy job (cf. Section III-B), which in practice makes the model adjustment troublesome. In this situation, CSHP can estimate the degradation trend preliminarily with periodic patterns eliminated, and then standard TTE and TTF prediction models can be trained with no need for ad-hoc adjustment anymore.

CSHP extracts the representative features of a degradation process to facilitate deeper degradation analysis, e.g., post-mortem degradation diagnosis. Specifically, we can analyze the correlations among degradation trends of different system metrics, and deduce the causal path towards root-cause analysis.

IX. Conclusion

In this paper, we propose an innovative approach, the CSHP framework, which is based on our modified Cox-Stuart test, and our Iterative Hodrick-Prescott Filter, to improve the state-of-the-art approach (we name it MKSK) in analyzing performance degradation problems of software systems. Several natural advantages of CSHP correspondingly overcome the intrinsic drawbacks of MKSK. Thus, CSHP is more practical and efficient in degradation detection and quantitative degradation characterization, especially for automatic large-scale degradation analysis. We evaluate the performance of CSHP and MKSK through extensive Monte Carlo experiments over 15,625,000 simulated time series, and over a real dataset containing throughput and availability time series of 99 web servers. Moreover, we technically prove the rationality and objectivity of our experiment design during the evaluation process. To summarize all our work, in consideration of detection power, false positive rate, degradation trend estimation accuracy, level of automation, robustness to noise, and time complexity, our approach CSHP is proved superior to the state-of-the-art approach MKSK in analyzing performance degradation problems of software systems.

References

[1] A. Avritzer and E. J. Weyuker, "Monitoring smoothly degrading systems for increased dependability," *Empirical Software Eng.*, vol. 77, pp. 59–77, 1997.

[2] K. Vaidyanathan, R. E. Harper, S. W. Hunter, and K. S. Trivedi, "Analysis and implementation of software rejuvenation in cluster systems," in *Proc. 2001 ACM SIGMETRICS Int. Conf. Measurement and Modeling of Comput. Syst.*, 2001, pp. 62–71.

[3] S. P. Kavulya, S. Daniels, K. Joshi, M. Hiltunen, R. Gandhi, and P. Narasimhan, "Draco: Statistical diagnosis of chronic problems in large distributed systems," in *2012 IEEE/IFIP Int. Conf. Dependable Syst. and Networks*, 2012, pp. 1–12.

[4] M. Grottke, L. Li, K. Vaidyanathan, and K. S. Trivedi, "Analysis of software aging in a web server," *IEEE Trans. Rel.*, vol. 55, no. 3, pp. 411–420, 2006.

[5] L. M. Silva, J. Alonso, and J. Torres, "Using virtualization to improve software rejuvenation," *IEEE Trans. Comput.*, vol. 58, no. 11, pp. 1525–1538, 2009.

[6] J. Alonso, J. Torres, J. Berral, and R. Gavalda, "Adaptive online software aging prediction based on machine learning," in *Proc. 2010 IEEE/IFIP Int. Conf. Dependable Syst. and Networks*, 2010, pp. 507–516.

[7] L. Li, K. Vaidyanathan, and K. S. Trivedi, "An approach for estimation of software aging in a web server," in *Proc. 2002 Int. Symp. Empirical Software Eng.*, 2002, pp. 91–100.

[8] K. J. Cassidy, K. C. Gross, and A. Malekpour, "Advanced pattern recognition for detection of complex software aging phenomena in online transaction processing servers," in *Proc. 2002 IEEE/IFIP Int. Conf. Dependable Syst. and Networks*, 2002, pp. 478–482.

[9] S. Garg, A. Puliafito, M. Telek, and K. S. Trivedi, "Analysis of preventive maintenance in transaction processing systems," *IEEE Trans. Comput.*, vol. 47, no. 1, pp. 96–107, 1998.

[10] D. Cotroneo, R. Natella, R. Pietrantuono, and S. Russo, "Software aging analysis of the Linux operating system," in *Proc. IEEE 21st Int. Symp. Software Rel. Eng.*, 2010, pp. 71–80.

[11] E. Marshall, "Fatal error: How Patriot overlooked a Scud," *Science*, p. 1347, 1992.

[12] T. Lou and J. Tang, "Solving performance degradation problems in WebSphere applications," IBM, 2007 [Online]. Available: http://www.ibm.com/developerworks/websphere/library/techarticles/0706_lou/0706_lou.html

[13] Oracle Database 2 Day + Performance Tuning Guide, 11g Release 1 (11.1), "Resolving performance degradation over time," 2007 [Online]. Available: http://docs.oracle.com/cd/B28359_01/server.111/b28275/tdppt_degrade.htm

[14] M. Gabel, A. Schuster, R. Bachrach, and N. Bjorner, "Latent fault detection in large scale services," in *Proc. 2012 IEEE/IFIP Int. Conf. Dependable Syst. and Networks*, 2012, pp. 1–12.

[15] K. Ren, J. López, and G. Gibson, "Otus: Resource attribution in data-intensive clusters," in *Proc. 2nd Int. Workshop MapReduce and its Applications*, 2011, pp. 1–8.

[16] J. Sun and J. Tang, "Solving memory problems in WebSphere applications," IBM, 2007 [Online]. Available: http://www.ibm.com/developerworks/websphere/library/techarticles/0706_sun/0706_sun.html

[17] S. Garg, A. van Moorsel, K. Vaidyanathan, and K. S. Trivedi, "A methodology for detection and estimation of software aging," in *Proc. 9th Int. Symp. Software Rel. Eng.*, 1998, pp. 282–292.

[18] A. Bovenzi, D. Cotroneo, R. Pietrantuono, and S. Russo, "Workload characterization for software aging analysis," in *Proc. IEEE 22nd Int. Symp. Software Rel. Eng.*, 2011, pp. 240–249.

[19] K. S. Trivedi, K. Vaidyanathan, and K. Goseva, "Modeling and analysis of software aging and rejuvenation," in *Proc. 33rd Annu. Simulation Symp.*, 2000, p. 270.

[20] D. R. Cox and A. Stuart, "Some quick sign tests for trend in location and dispersion," *Biometrika*, vol. 42, pp. 80–95, 1955.

[21] R. Hodrick and E. C. Prescott, "Post-war U.S. business cycles: An empirical investigation," *J. Money*, vol. 29, no. 1, 1980.

[22] E. Schlicht, "Estimating the smoothing parameter in the so-called Hodrick-Prescott filter," *J. Japan Statistical Assoc.*, vol. 35, no. 1, pp. 99–119, 2005.

[23] M. Bakkaloglu, J. Wylie, C. Wang, and G. R. Ganger, "On correlated failures in survivable storage systems," Parallel Data Laboratory, CMU, (CMU-CS-02–129), Paper 107, 2002.

[24] M. Yarmohammadi, "A filter based Fisher g-test approach for periodicity detection in time series analysis," *Scientific Res. Essays*, vol. 6, no. 17, pp. 3717–3723, 2011.

[25] A. Bovenzi, D. Cotroneo, R. Pietrantuono, and S. Russo, "On the aging effects due to concurrency bugs: a case study on MySQL," in *Proc. IEEE 23nd Int. Symp. Software Rel. Eng.*, 2012.

[26] A. Dermoune, B. Djehiche, and N. Rahmania, "A consistent estimator of the smoothing parameter in the Hodrick-Prescott filter," *J. Japan Statistics Soc.*, vol. 38, no. 2, pp. 225–241, 2008.

[27] D. Cotroneo, S. Orlando, R. Pietrantuono, and S. Russo, "A measurement-based ageing analysis of the JVM," *Software Test., Verification and Rel.*, vol. 23, no. 3, pp. 199–239, 2011.

[28] D. Cotroneo, S. Orlando, and S. Russo, "Characterizing aging phenomena of the Java virtual machine," in *Proc. 26th IEEE Int. Symp. Reliable Distributed Syst.*, 2007, pp. 127–136.

[29] J. B. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-25, no. 3, pp. 235–238, 1977.

[30] G. Andria and M. Savino, "Interpolated smoothed pseudo Wigner-Ville distribution for accurate spectrum analysis," *IEEE Trans. Instrum. Meas.*, vol. 45, no. 4, pp. 818–823, Aug. 1996.

[31] F. J. Harris, "On the use of Windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[32] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 1988.

[33] Big O Notation, MIT, 2003 [Online]. Available: http://web.mit.edu/16.070/www/lecture/big_o.pdf

Pengfei Zheng is currently working toward the M.S. degree in the Department of Computer Science and Technology at Xi'an Jiaotong University, China.

His research interests include dependable computing, operating systems, and software engineering.

Yong Qi received the Ph.D. degree from Xi'an Jiaotong University, China.

He is currently a full Professor in the Department of Computer Science and Technology at Xi'an Jiaotong University. His research interests include operating systems, distributed systems, and cloud computing.

Yangfan Zhou (M'14) received the Ph.D. degree from The Chinese University of Hong Kong (CUHK) in 2009.

He is currently a research staff member with the Shenzhen Research Institute at CUHK. His research interests include mobile computing, cloud computing, and the Internet of Things, with a focus on their software reliability issues.

Pengfei Chen is currently working toward a M.S.–Ph.D. joint degree at Xi'an Jiaotong University, China.

His research interests include dependable computing, cloud computing, distributed computing, and software engineering.

Jianfeng Zhan (M'13) received the Ph.D. degree in computer engineering from the Chinese Academy of Sciences, Beijing, China, in 2002.

He is currently a Professor of computer science with the Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include distributed and parallel systems.

Michael Rung-Tsong Lyu (F'04) received the Ph.D. degree in computer science from the University of California, Los Angeles, CA, USA, in 1988.

He is now a Professor in the Department of Computer Science and Engineering at The Chinese University of Hong Kong.

Dr. Lyu is an IEEE Fellow and an AAAS Fellow, and received the IEEE Reliability Society 2010 Engineer of the Year Award.