Real Time Empirical Synchronization of IoT Signals for Improved AI Prognostics

Guang Chao Wang
Oracle Physical Sciences Research Center
OracleLabs, Oracle Corp.
San Diego, CA
guang.wang@oracle.com

Kenny Gross
Oracle Physical Sciences Research Center
OracleLabs, Oracle Corp.
San Diego, CA
kenny.gross@oracle.com

***Abstract*— A significant challenge for Machine Learning (ML) prognostic analyses of large-scale time series databases is variable clock skew between/among multiple data acquisition (DAQ) systems across assets in a fleet of monitored assets, and even inside individual assets, where the sheer numbers of sensors being deployed are so large that multiple individual DAQs, each with their own internal clocks, can create significant clock-mismatch issues. For Big Data prognostic anomaly detection, we have discovered and amply demonstrated that variable clock skew issues in the timestamps for time series telemetry signatures cause poor performance for ML prognostics, resulting in high false-alarm and missed-alarm probabilities (FAPs and MAPs). This paper describes a new Analytical Resampling Process (ARP) that embodies novel techniques in the time domain and frequency domain for interpolative online normalization and optimal phase coherence so that all system telemetry time series outputs are available in a uniform format and aligned with a common sampling frequency. More importantly, the "optimality" of the proposed technique gives end users the ability to select between "ultimate accuracy" or "lowest overhead compute cost", for automated coherence synchronization of collections of time series signatures, whether from a few sensors, or hundreds of thousands of sensors, and regardless of the sampling rates and signal-to-noise (S/N) ratios for those sensors.**

Keywords—Cross Power Spectral Density, Genetic Algorithm, Signal Synchronization

I. INTRODUCTION

The expansions of Internet-of-Things (IoT) dense-sensor applications across many industrial segments are fast growing in the past decade. For example, a modern oil refinery these days has 1M sensors recording time series signals generating observations 24x7x365. A typical large commercial airplane has 75,000 sensors these days, and a medium size enterprise or cloud data center can have 1M sensors. Very many dense-sensor IoT applications have distributed data-acquisition (DAQ) modules across their fleet of assets. Moreover, it is not uncommon for there to be multiple DAQ modules inside each large asset. However, it is most often the case that the clocks in the DAQ modules are generating the timestamps for the packets of observations being aggregated by the DAQ, and humans set up the clocks for the DAQ modules. Whether the distributed DAQ clocks are out of synch because of human errors, or from a variety of long-term clock skew mechanisms, the consequence for big-data Machine Learning (ML) anomaly discovery can be very poor prognostic performance. Variable clock skews can cause the best ML pattern recognition algorithm to "blur" the patterns of correlation across large-scale collections of time-series signatures. In addition, variable clock skews across a collection of digitized sensor time series result in excessive False-Alarm Probabilities and Missed-Alarm Probabilities (FAPs and MAPs) for the prognostic surveillance ML algorithm.

One hardware-based solution is to subscribe to commercial clock-synchronization hardware/software systems that will periodically synchronize a network of distributed clocks to a remote highly-accurate atomic clock. This approach became popular in the previous decade, in spite of the significant cost to design in clocks with either network or wireless capability for periodic re-sync updates on the fly. However, this clock-synch methodology has fallen out of favor for many business-critical facilities after some well-publicized hacks occurred when hackers discovered they could tap into facility critical assets through the clock-synch connections (which require a penetration through the facility-network firewall).

Oracle previously developed a separate solution for empirical re-synchronization [1-2], which requires no hardware modifications anywhere in the IoT assets or networks, nor any penetration through the firewall, in the form of machine-learning-based empirical synchronization of all types of IoT digitized time series signatures. This innovative technique, called the Analytical Resampling Process (ARP), has demonstrated significant ROI for autonomous prognostics over the years. ARP embodies novel techniques in the time domain and frequency domain for optimal phase coherence so that all system telemetry time series outputs are available in a uniform format and aligned with a common sampling frequency. ARP synthesizes data from multiple, disparate-format sources and has become an indispensable tool for producing synchronized data streams suitable for use in the design, testing, and performance evaluation of real-time prognostic monitoring techniques. Over the last 15 years, three completely different approaches for empirical re-synchronization of time series signals have been developed. These 3 earlier versions of ARP include:

1. Correlogram: One signal is picked as the 'reference anchor' signal, meaning its timestamps will be assumed to be correct. All other signals in the asset or fleet of assets will be empirically aligned to the "reference anchor" signal by computing pairwise

cross correlation coefficients, then systematically "adjusting" the lags for the individual signals to optimize the correlation coefficient with respect to the "reference anchor".

2. CPSD: Cross Power Spectral Density technique [1]. A bivariate frequency-domain technique that uses a sophisticated FFT computation that infers with high accuracy the "phase angle" (in the frequency domain) between two timeseries. An algorithm is then employed to compute an optimal estimate of the lag time from the phase angle. Pairwise computations are performed for all signals in the collection, adjusting signals to bring the empirical lag times to zero.

3. Genetic Algorithm (GA): A metaheuristic-type optimization technique wherein signals are given random "kicks" in positive or negative directions, after which the overall synchronization score for the population is evaluated. Signal kicks that result in an improvement are retained for the next "generation", whereas kicks that worsen the degree of correlation are moved back to the positions they held in the previous generation. Kick sizes are systematically reduced in each generation (to prevent oscillatory "hopping over" the optimum synchronization value) [3].

Although each of the above earlier ARP approaches does the job for empirical re-synchronization of time series sensor signals, it is by no means obvious, even to senior-level data scientists, which technique is the "best" for any given prognostic monitoring use case in the various industries starting to adopt dense-sensor prognostic solutions (which include Utilities, Oil-and-Gas, Manufacturing, Transportation, and of course Datacenters). However, it is a significant amount of work to decide in advance which of the above 3 earlier ARP approaches may be "best" for any given use case. The reasons are twofold:

1) For some IoT prognostic surveillance use cases, "best" means highest possible accuracy for realignment of signals for which time-stamps are out of sync due to variable clock-skew issues in DAQs. But for other IoT prognostic use cases, ball-park alignment of signals (e.g. reducing clock-skews from minutes to ~10 seconds) is more than adequate to achieve Prognostic Functional Requirements (PFRs), but overhead compute cost (CC) is the gating factor for whether ML prognostics are economically feasible. This is the case for large-scale streaming analytics where the overhead of empirical re-sync computations performed "upstream" of the ML prognostics can exceed the available CPU cycles for the computer (or VM in a cloud architecture), without requiring a more powerful prognostic platform (or more VMs).

2) Both the re-sync accuracy, and the overhead CC, are complex nonlinear functions of the number of monitored signals, the sampling rates for those signals, and the signal-to-noise ratios (SNRs) for those signals.

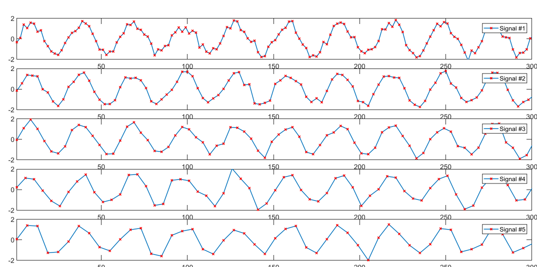
Because of the heretofore intractable complexity in knowing which of 3 drastically different mathematical implementations of ARP might be "best" for any given use case, we have in the past used a very human-intensive seat-of-the-pants approach in selecting which one of the 3 ARP algorithms to use for a prognostic challenge at hand: A human data scientist knowledgeable with Oracle's 3 ARP approaches would take a test data set, and "try out" a ML prognostic evaluation using each ARP algorithm separately, then compile results, and see which technique produced the best prognostic accuracy (if that is the top priority), or produced the lowest overhead compute cost (if that is the top priority), then deploy that algorithm.

In this paper, we teach an automated parametric framework that systematically evaluates any example dataset of target time series signatures, lets the end customer specify whether top accuracy or low overhead compute cost (CC) is the most desirable functional requirement, and then uses an innovative adaptive machine-learning approach to select for the end user the optimal algorithmic implementation to achieve ARP functional requirements. This automates the process so that novice users at companies subscribing to Oracle Cloud autonomous prognostic offerings across multiple industrial segments served by Oracle, will always have optimal phase-synchronization of signals (and hence highest-sensitivity prognostic performance for ML anomaly discovery, with lowest false-alarm and missed-alarm probabilities).

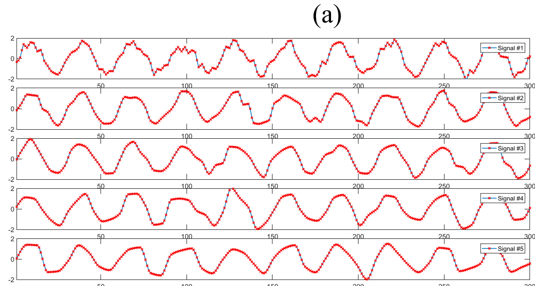
II. METHODOLOGY

A. ARP-resampling:

First of all, the various signals like power, CPU utilization, performance, and temperature need to be analytically resampled (i.e. they may be physically sampled at different frequencies, but in this step they are analytically upsampled/downsampled as necessary to produce uniform sampling intervals). One common sampling frequency is picked and all the signals are resampled so that they have common time stamps. One common sampling frequency is picked and all the signals are upsampled or downsampled through an innovative imputation algorithm so that they have common time intervals (but still may be out of phase alignment).



(a)



(b)

Figure 1: Use case of ARP upsampling

Figure 1a and Figure 1b illustrate the performance of ARP resampling through a use case where 5 telemetry signals in different sampling rate are presented. The algorithm first detects the fastest sampling rate (or user-specified sampling rate of 1s in this case), and then upsamples those slower signals using spline interpolation for purposes of illustration. The output are signals with uniform 1s samples.

B. ARP-Correlegram Resynchronization

The most basic approach in ARP is called the correlegram technique, which performs an analysis of correlation coefficient vs. lag in time domain, to determine the optimum lag at which two signals are aligned the closest, i.e., the lag at which the correlation (absolute value) is the highest. This step is required even though the signals may their individual time stamps, due to various clock-mismatch mechanisms mentioned above. Examples below show the correlegram for power and utilization signals when the highest correlation occurs at a positive lag, at a negative lag, and after the appropriate phase shift has been performed. This new innovation works even when all clocks for the hardware power meter, the hardware external temperature DAQ, the operating system (OS), and (if applicable) the Service Processor (SP) are totally out of sync.

If the highest correlation occurs at a non-zero lag, one of the signals is shifted by that corresponding lag so that with this newly shifted signal, the highest correlation occurs at zero lag. The correlogram analysis is performed for every pair of signals. This is not very resource intensive since we have only a few number of signals for this illustrative example. Subsequent ML prognostics are now generated using the processed set of signals. This results in an accurate analysis.

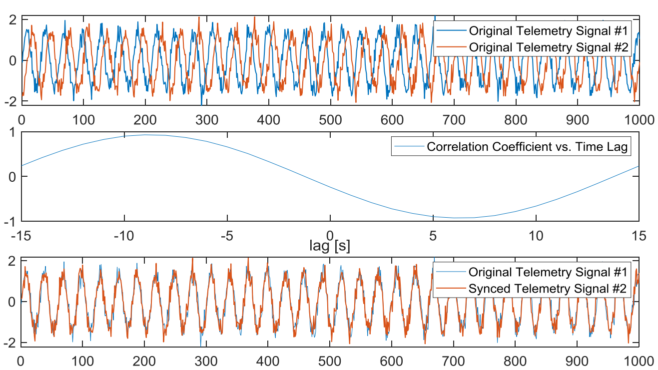


Figure 2: Use case of ARP Correlegram

Figure 2 illustrates a use case of ARP correlegram on two out of sync telemetry signals. The correlogram analysis is performed for matching the signal #2 to signal #1 by iteratively measuring the correlation coefficient between signal #1 and shifted copies of signal #2 (middle plot). The non-zero lag corresponding to the highest correlation is captured and used to shift the signal #2, yielding the synced two signals (bottom plot).

C. **ARP-CPSD Resynchronization**

Some business-critical applications need the finest resolution possible on alignment of signals, thus we developed a second technique that uses a cross power spectral density (CPSD) method in the frequency domain. The signals are transformed into the frequency domain using Fourier decomposition and a cross power spectral density (CPSD) analysis is performed to determine the relative phase shift (and hence time lag) between all possible pairwise combinations of telemetry signals. In this analysis, the phase angle is plotted against frequency, then the slope of the phase-vs-frequency curve indicates whether the original telemetry signals are aligned or not. A non-zero slope in the phase angle vs frequency line indicates that the two signals are not aligned. The phase angle is treated as an adjustable parameter in a systematic stepwise iterative algorithm that is recursively applied until a zero slope indicates that the signals are perfectly aligned.

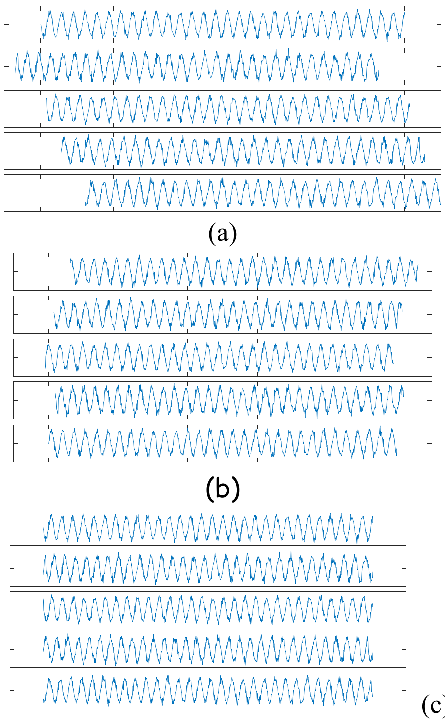
If there is a non-zero slope, one of the signals is shifted by the corresponding lag so that with this newly shifted signal, the phase angle vs frequency plot has zero slope. The CPSD analysis is performed for every pairwise combination of telemetry signals.

The novel frequency-domain technique has been developed for analytical resampling and phase shift optimization of telemetry signals coming from server power measurements that are being required by new EPA guidelines for all future computer servers. The advantage that this technique brings is that power-versus-utilization monitoring can be performed more efficiently and accurately, even when the signals are dynamically varying and even when those signals are contaminated with significant degrees of quantization from low-resolution A/D chips used in most enterprise computing systems. Conventional time-domain signal synchronization techniques can presently only measure EPA power efficiency metrics while all signals are held constant with time. If the signals are varying dynamically, then any small time shifts due to clock skews between the external hardware power meter, the temperature measurements, and the OS throughput and performance variables, causes very large uncertainties with conventional methods).

D. ARP-Genetic Algorithm (GA) Resynchronization

Both of the above techniques have a reasonable compute cost for up to dozens of signals. But the compute cost goes up geometrically with the number of signals. For large-scale telemetry databases that may contain many dozens or hundreds of time series signals, we also devised a third technique that incorporates a GA algorithm and that scales extremely well with low compute cost even for huge numbers of signals:

Shift each signal with random phase kicks in turn and compute the objective function. If the objective function value increases, roll back and continue process until the objective function reaches a pre-assigned threshold or does not increase for several iterations of random kicks.



(a) (b) (c)

Figure 3: use case of GA at intermediate iterations

Figure 3(a)-(c) illustrates a use case for the GA phase optimization based technique. The clock-mismatch issues in distributed data-acquisition modules cause correlated processes to be out of alignment when consumed by ML Pattern Recognition algorithms. Our proposed technique goes through three stages and optimally aligns the signals between each other.

III. EVALUATION

For any given customer with a set of signals to analyze with machine learning (ML) in Oracle's prognostic cloud, that customer's set of signals will embody a given number of signals, number of samples (equivalent to saying a certain sampling rate), and characteristic signal-to-noise ratio for the signals. Again, the substantial challenge for a customer to set up an ARP pre-processing algorithm to optimally synchronize her signals (correcting for clock-skew issues in the data acquisition systems) lies in the fact that it takes a very involved manual investigation to find the best ARP approach that achieves her goals of either:

[A] minimizing overhead compute cost [common for real-time streaming prognostics, where latencies have to be minimized]; or

[B] maximizing the accuracy of the time synchronization [common for "batch-wise" prognostics, where the signals from the customer's critical assets are periodically (e.g. once per day, once per week) analyzed to assess aging-degradation of the assets. For this use case compute cost is no issue; or

[C] a dual-optimization objective of hitting the best sweet-spot between lowest compute cost with best accuracy [common for cases where the ML algorithm can easily keep up with real-time, so both high accuracy and low compute cost are desirable].

The technique reported in this paper conducts the entire investigation automatically. The customer user can simply submit an example dataset of all the signals she will wish to monitor with ML prognostics, and her specified top "goodness" metric/objective for automated clock mismatch synchronization: e.g. lowest possible compute cost, vs highest possible synchronization accuracy, vs bi-variate optimal accuracy-vs-compute cost.

Of course the analyses and selection of optimal algorithm are completely different for any individual customer use case (because both compute cost and accuracy are complex nonlinear functions of the number of signals, sampling rates for the signals, and signal-to-noise ratios for the signals). We discussed in the introduction that how Oracle's new Automated Optimal ARP Supervisor solves this challenge for any collection of end-customer signals, and "takes the human out of the loop" so that tedious replicated manual experiments are no longer necessary, literally creating a fully automated Oracle tool that can be operated by a beginning level data technician who doesn't have to know anything about empirical phase synchronization…just supplies a test dataset and the new Oracle automated optimal ARP supervisor performs a systematic parametric nonlinear tradeoff evaluation and sets up an optimal ARP algorithm for the customer's specific use case.

We illustrate in this section how we approach this with a systematic parametric evaluation that varies the number of signals, the number of samples for the signals (and hence the sampling rates for the signals), and the signal-to-noise ratios (S/N Ratios) for the signals, and computes the overhead compute cost (CC) and the synchronization accuracy (measured in RMSE, in the same units as the customer's signals), then picks a customized ARP algorithm that optimally achieves the customer's most important performance metric.

Figure 4 below illustrates how synchronization accuracy varies for the Correlegram technique as a function of number of signals and number of observations (equivalently, sampling rate). We can see that for the Correlogram method, the Accuracy is only mildly related to the number of signals and sampling rate, whereas Figure 5 shows that the overhead compute cost varies substantially with the number of signals and sampling rates.

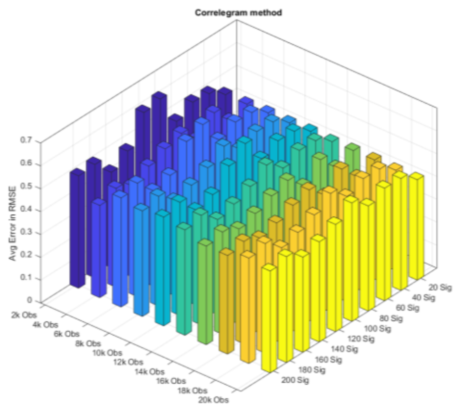


Figure 4: Evaluation of accuracy for Correlegram method.

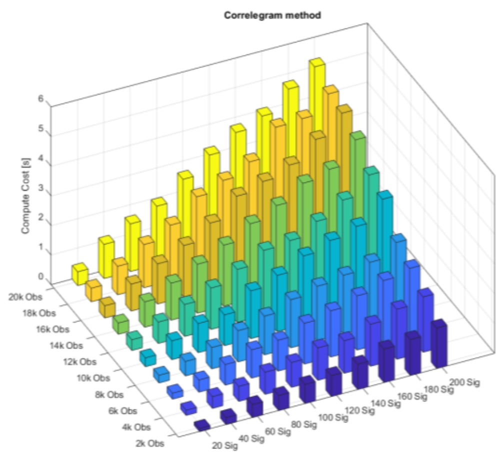


Figure 5: Evaluation of compute cost for Correlegram method.

Figure 6 and Figure 7 illustrate example parametric evaluations of accuracy and compute cost for the CPSD method. It is important to note in Figure 7 that the compute cost is relatively invariant to the sampling rate for the signals, but very sensitive to the number of signals for the customer's specific use case.

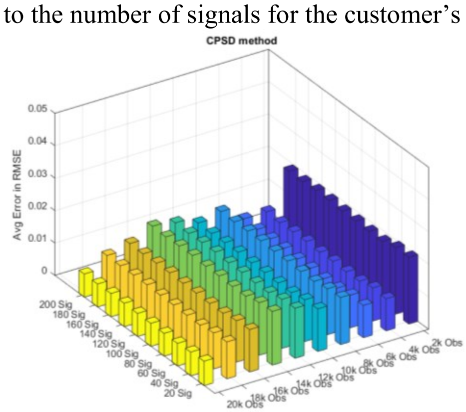


Figure 6: Evaluation of accuracy for CPSD method.

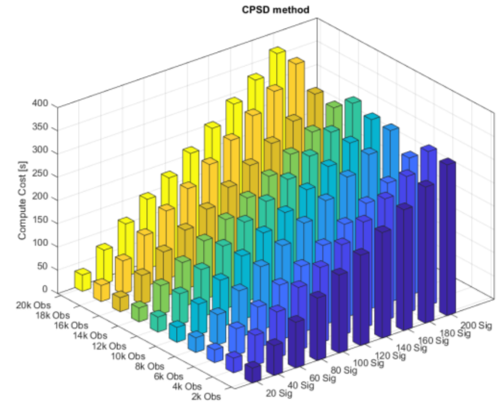


Figure 7: Evaluation of compute cost for CPSD method.

Figure 8 and Figure 9 illustrate the corresponding parametric empirical functional relationships between compute cost (CC) and synchronization accuracy (RMSE) for the Genetic Algorithm (GA) approach.

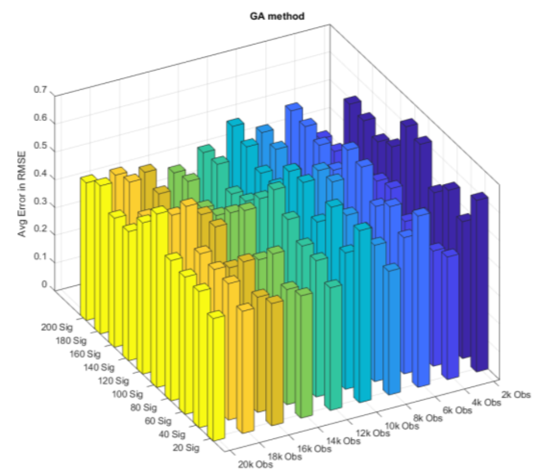


Figure 8: Evaluation of accuracy for GA method.

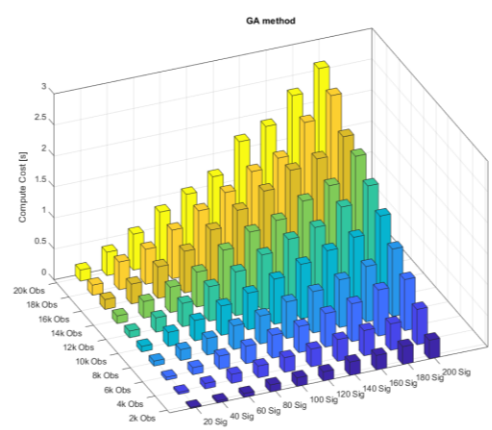
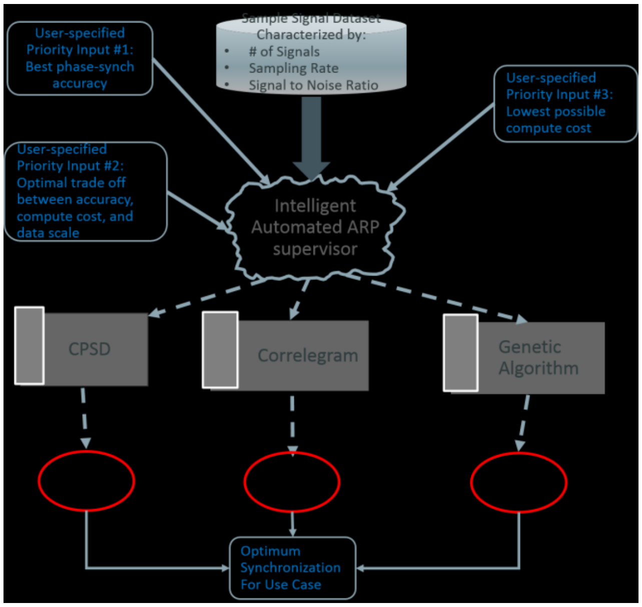
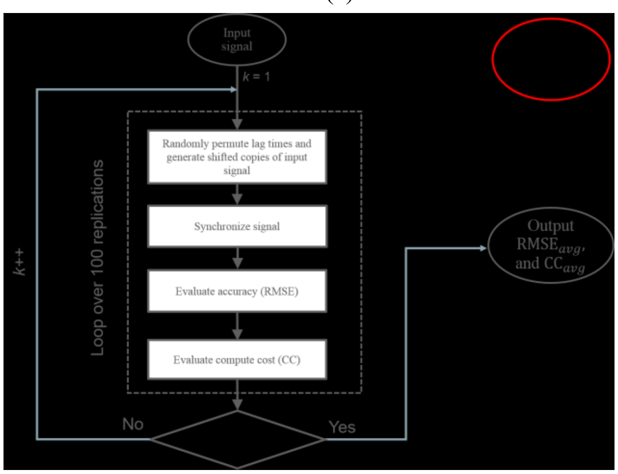


Figure 9: Evaluation of compute cost for GA method.

These systematic parametric optimization techniques have been embodied into an Automated Optimal ARP Supervisor algorithm as illustrated schematically in Figure 10.



(a)



(b)

Figure 10: Flowchart A and B of the automated optimal ARP.

Note that if a data scientist has a use case with some number of signals "N" (a number that is different depending on his/her system, machine, or use case): let's say she has 33 signals captured in a dataset with a fixed lead/lag relationship captured in the dataset. We synchronize that set of 33 signals, then decide by the procedure in A which algorithm had the lowest compute cost and/or best accuracy for that one "snapshot" dataset. However, with just that one snapshot use case, the results and conclusion could be very misleading. It could be that an hour later or day later the leads/lags could be completely different. [This is especially relevant where the leads/lags are due to variable flow processes, and the flowrates can change due to many factors.] It would be imprudent for us to pick an optimum algorithm on the basis of just one "snapshot" of customer data. Instead, in our autonomic implementation of this Optimal ARP Supervisory algorithm we take the original measured signals (which define the # of signals available and give a good representation of the sampling rate and "noisiness" of the signals) and we do 100 replicated iterations [100 is adequate, in our experiments we get good asymptotic accuracy and compute cost by averaging across 100 replications] in which we randomly permute the lead/lag times in each iteration, invoke the 3 candidate ARP sync algorithms [in the preferred embodiment, but allow for the possibility that additional algorithms can be plugged in in future versions, and Automated ARP still yields an optimum sync algorithm for any given set of measured signals for any use case], then select the sync algorithm that best complies with the customer's most important performance criteria [(a) balance between compute cost and accuracy, (b) lowest possible overhead compute cost during real time streaming applications, vs (c) highest possible sync accuracy regardless of compute cost).]

IV. Conclusion

This paper describes a new Optimal Analytical Resampling Process for autonomic dense-sensor IoT prognostic use cases that embodies novel techniques in the time domain and frequency domain for interpolative online normalization and optimal phase coherence so that all system telemetry time series outputs are available in a uniform format and aligned with a common sampling frequency. ARP synthesizes data from multiple, disparate-format sources and has become an indispensable tool for producing synchronized data streams suitable for use in the design, testing, and performance evaluation of dynamic power monitoring techniques for server components and subsystems with important spinoff applications for proactive fault monitoring tools that catch incipient problems in enterprise computing servers. Analytical re-synchronization for avoidance of variable clock skews tremendously improves prognostic performance for all types of ML-based big-data surveillance use cases.

References

[1] "Improved Energy Monitoring and Prognostics of Servers via High-Accuracy Real-Time Synchronization of Internal Telemetry Signals," K. C. Gross and K. Vaidyanathan, *Proc. IEEE World Congress in Computer Science, Computer Engineering, and Applied Computing (WorldComp2010)*, Las Vegas, NV (Aug 2010).

[2] "Improved Methods for Early Fault Detection in Enterprise Computing Servers Using SAS Tools," K. C. Gross and K. Mishra, *2004 SAS Users Group International (SUGI 29)*, Montreal, Canada. (May 9 – 12, 2004).

[3] "Genetic Algorithm Approach for Optimal Phase Shift Synchronization of Telemetry Signals," K. C. Gross and Y. Bao, U.S. Patent 7,391,835 (6/24/2008).