Autocorrelation-Based Characterization of Industrial Process Dynamics for Automated Process Monitoring: A Benchmark-Informed Framework

Iliyan Vasilev

University of Chemical Technology and Metallurgy, Sofia, Bulgaria, (e-mail: i.vasilidis@uctm.edu)

Abstract **- Industrial control traces contain temporal dependencies that reflect regulation, storage, transport and response to disturbances. Yet correlations obtained under different sampling rates or annotation regimes are not directly comparable. This paper develops an interpretable, two-branch temporal autocorrelation descriptor and specifies its use with three public industrial benchmarks: HAI 23.05, BATADAL and the classical Tennessee Eastman Process (TEP) files. The method is based on comparison of the short-lag correlation profiles of signal levels and first differences against normal-operation references, using windows and thresholds calibrated separately for each dataset. The BATADAL training schedule contains 492 attack-hours, but only 219 (44.5%) are explicitly labeled. The unlabeled periods cannot be assumed to represent normal operation. An AR(1) example shows how autocorrelation varies between rapidly decaying, persistent and alternating signals. The study provides a reproducible method for comparing these patterns**.

Keywords **- Industrial process, process dynamics, automated process monitoring, anomaly diagnosis.**

INTRODUCTION

Industrial process variables are not independent observations. Feedback loops, material inventories, actuator limits and measurement delays create temporal structure even when the plant is functioning normally. A temperature measured by a slow sensor, a reservoir level integrating pump flow and an oscillatory pressure-control loop can have similar means and variances but substantially different autocorrelation functions (ACFs). This motivates the use of a time-lag correlation profile as a descriptive complement to conventional threshold monitoring [1].

Temporal correlation alone cannot identify the cause of a process change. High lag-one correlation may result from slow dynamics, drift or periodic loading, while low correlation may reflect sensor noise or strong control action. Dynamic PCA has shown the value of lagged data in process monitoring [2]. Here, simpler autocorrelation profiles are calculated for individual channels and combined across defined time windows.

The three datasets represent different industrial settings. HAI uses a hardware-in-the-loop control system with attack scenarios [3], [4]. BATADAL models disturbances in a water distribution network [5], [6], while Tennessee Eastman simulates a chemical process [7], [8]. Differences in sampling rates, labels and operating conditions make direct numerical comparisons difficult.

The study focus is on measuring correlation persistence and sign alternation, defining suitable analysis windows and examining what can be established from the available dataset documentation. The method is intended for process monitoring and disturbance analysis, without claiming to identify attack causes or prove control stability.

OPEN DATA AND STUDY DESIGN

Benchmark provenance and scope

The HAI 23.05 release combines boiler, turbine, water-treatment and HIL-model signals. Its repository describes 86 process data points per second, normal training files hai-train1 to hai-train4 and attack test files hai-test1 and hai-test2, with 52 attack cases in the latter files collectively [3]. Each recording file is a continuous series; consecutive files must not be concatenated without an explicit boundary. Attack labels are evaluation metadata and are excluded from the correlation calculation.

The BATADAL source provides hourly SCADA observations of the C-Town water network. Training Dataset 1 is attack-free; Training Dataset 2 spans approximately six months and contains attacks with incomplete labeling; the original competition test dataset was distributed without labels [5]. The tank-level channels $L_T1$to L_T7 provide one physically interpretable starting group. Binary pump states should be analyzed separately rather than pooled indiscriminately with analog signals.

The reference Tennessee Eastman files document 41 measurements and 11 manipulated variables per record, sampled every 180 s [7], [8]. For the classical 960-sample fault-test traces, the programmed fault begins after the first 160 observations (8 h); this timing must not be confused with other, later TEP datasets whose onset conventions differ [8], [9]. The pre-fault segment is too short for arbitrary, long reference windows, and each simulated fault run must remain separate.

Each collection is elaborated respectivelly on its native time grid. Time alignment follows actual timestamps where available and the missing HAI/BATADAL steps are not silently interpolated. The proposed primary sliding windows are 600 points with a 30-lag horizon for Each dataset is analyzed at its original sampling rate, using recorded timestamps without interpolating missing values. Sliding windows of 600 samples and 30 lags are proposed for HAI, 72 samples and 12 lags for BATADAL, and 120 samples and 12 lags for TEP. These

cover 10 min, 72 h and 6 h, respectively, with lag horizons of 30 s, 12 h and 36 min.

TABLE 1 BENCHMARK PROPERTIES AND RECOMMENDED ANALYTICAL UNIT

Benchmark	Sampling	Unit of analysis
HAI 23.05	1 s	One continuous CSV; analog channels
BATADAL	1 h	One source split; tank level/pressure groups
TEP classic	3 min	One simulated run; selected XMEAS

Sampling units are physical time steps; comparison is within a dataset, not across equal lag indices.

Event alignment, time scales and independence

Overlapping windows are useful for monitoring but cannot be treated as independent observations. Statistical estimates should therefore be based on separate events or data blocks. Windows covering attack onset are distinguished from those fully within an event. Since BATADAL includes 24-hour attacks, its 72-hour windows cannot cover an attack alone. Shorter windows are needed, although their correlation estimates may be less reliable.HAI, 72 points with 12 lags for BATADAL, and 120 points with 12 lags for TEP. These correspond to 10 min/30 s, 72 h/12 h and 6 h/36 min respectively. The aim is to leave enough pairs for each lag without pretending that identical sample counts represent identical dynamics.

TEMPORAL CORRELATION METHOD

Parallel level and increment profiles

Denoting $x(j,t)$ the value of analog signal j at native time step t in a window W of n observations. Screening removes channels with missing segments, non-finite values or effectively zero within-reference variance. Quantities derived from normal data (selection thresholds and calibration distributions) are frozen before reading attacked/faulted test segments. Two representations are considered: the observed level $x(j,t)$ and its first difference, $\Delta x(j,t) = x(j,t) - x(j,t-1)$.The second branch suppresses slow offsets, whereas the level branch retains shifts that differencing could otherwise hide.

For branch b in {level, difference}, a fixed window means is subtracted before computing the conventional biased sample autocorrelation for $k = 1,\ldots,L[1]$:

$$\hat{\rho}_{j,W}^{(b)}(k) = \frac{\sum_{t=k+1}^{n}(z_t - \bar{z})(z_{t-k} - \bar{z})}{\sum_{t=1}^{n}(z_t - \bar{z})^2}$$

Here z denotes the selected branch and L is strictly less than n. The same reference formula is applied to both branches. Profiles are undefined, not zero, for windows whose variance vanishes. No stationarity of all real plant channels is assumed: the local approximation is inspected empirically, and the difference branch is reported alongside the level branch.

Two elementary descriptors allow the mechanism to be read directly. Absolute persistence $A(j,W,b)$ summarizes how much lag dependence remains, while signed persistence $S(j,W,b)$ distinguishes predominantly positive from alternating correlations:

$$\hat{\rho}_{j,W}^{(b)}(k) = \frac{\sum_{t=k+1}^{n}(z_t - \bar{z})(z_{t-k} - \bar{z})}{\sum_{t=1}^{n}(z_t - \bar{z})^2}$$

A is between zero and one, and S between minus one and one for the selected estimator. A describes the finite declared lag horizon, not a physical time constant. In a new dataset the plotted horizontal axis is labeled both in lag steps and equivalent physical time.

Normal signature, drift score and threshold

Normal training files are first partitioned chronologically into a reference section and a separate calibration section. For each channel, branch and lag, the baseline signature is the median autocorrelation across non-overlapping reference windows. The aggregate deviation of a new window is then defined by a robust median over the retained channels J:

$$D(W) = \underset{j\in J}{\text{med}} \frac{1}{2L}\sum_{b=0}^{1}\left\| r_{j,W}^{(b)} - \tilde{r}_{j,N}^{(b)} \right\|_1$$

The threshold h is set at the 99th percentile of D, calculated from separate normal calibration windows within each dataset. A window is flagged when D(W) exceeds h. Attack labels are not required for calibration, but this threshold does not guarantee a 1% false-alarm rate under different operating conditions. With fewer than about 100 effectively independent windows, the percentile estimate becomes unreliable, requiring more normal data or a lower quantile. High scores should also be examined using the ACF curves of the contributing signals.

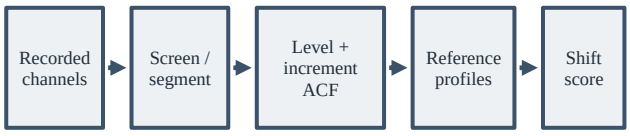

Fig. 1. Specified processing sequence. Selection and reference profiles are fitted on normal data; the comparison stage is evaluated on held-out segments.

Evaluation rules and failure modes

The three datasets require different treatment of event boundaries. In HAI, windows crossing an attack boundary are kept apart from those clearly belonging to normal or attack periods. BATADAL presents a different problem: the absence of an attack label does not always mean normal operation. Its published attack schedule is therefore considered when evaluating the results. For Tennessee Eastman, the fault onset is taken from each simulation run individually. Detection coverage, time to the first alarm and the fraction of normal windows producing alarms are calculated where reliable labels are available. Otherwise, the corresponding metric is left as not assessed.

The ACF estimates also need careful interpretation and analysis. Results from short windows can vary considerably, particularly when the process has periodic behavior. Constant sensor readings make the calculation

in (1) undefined because the variance is zero. Differencing introduces another difficulty: after a sudden change or saturation, the resulting signal may show little remaining correlation. We therefore examine the ACF together with signal means, variances, measured ranges and process alarms. Even a substantial correlation change does not, by itself, tell us whether the cause is a process fault, a change in operating conditions or an attack.

BENCHMARK METADATA RESULTS

Attack-label observability in BATADAL

The published BATADAL Training Dataset 2 schedule contains seven attack cases [5]. Together, they account for 492 attack-hours, but only 219 hours (44.5%) are labeled. Cases A2 and A7 have no labeled hours at all. These figures were calculated from the published attack schedule rather than the SCADA measurements. They therefore provide no direct evidence of changes in autocorrelation or detector performance.

TABLE 2 ANNOTATION COVERAGE IN BATADAL TRAINING DATASET 2

Attack	Exposure, h	Labeled, h	Coverage, %
A1	50	42	84.0
A2	24	0	0.0
A3	60	60	100.0
A4	94	37	39.4
A5	60	7	11.7
A6	94	73	77.7
A7	110	0	0.0
Total	492	219	44.5

Recalculated solely from the official event-duration and labeled-hours schedule [5].

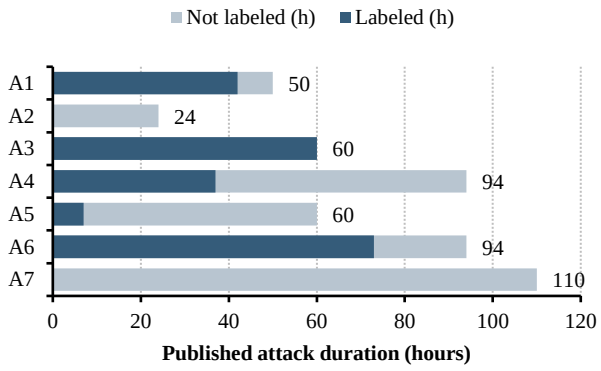

Fig. 2. Official BATADAL attack hours partitioned into labeled and unlabeled portions. Gray sections must not be interpreted as nonattack operation.

This creates a problem when evaluating detection results. Some attack hours have no labels, so treating all unlabeled records as normal could lead to incorrect false-alarm counts. An alarm during a documented but unlabeled attack might be classified as a false positive. To avoid this, alarms are checked against both the dataset labels and the complete attack periods given in the published schedule. Records that cannot be classified reliably are marked as “unknown”.

Analytical consistency check

The descriptor was first examined using a stationary AR(1) model, $z_t = \varphi z_{t-1} + e_t$, with $|\varphi| < 1$ and uncorrelated noise. Its theoretical autocorrelation at lag k is $\rho_k = \varphi^k$. We considered 12 lags for three values of φ: 0.20, 0.85 and −0.65. These represent rapid correlation decay, slow decay and alternating positive and negative correlations. The values in Table 3 were calculated from the AR(1) model and are not measurements from HAI, BATADAL or Tennessee Eastman.

TABLE 3 THEORETICAL TWELVE-LAG AUTOCORRELATION CHECK

AR(1) φ	ρ(1)	A(1:12)	S(1:12)
+0.20	+0.200	0.0208	+0.0208
+0.85	+0.850	0.4051	+0.4051
-0.65	-0.650	0.1539	-0.0326

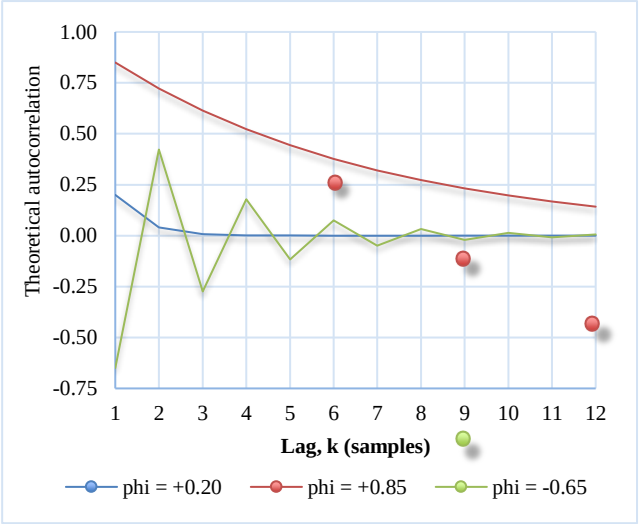

Fig. 3. Exact AR(1) population autocorrelation, illustrating decay rate and sign alternation.

The slowly decaying positive model gives a larger A than the fast model, while the negative-coefficient model retains noticeable absolute lag dependence despite a near-zero signed mean. Thus, A alone cannot distinguish oscillation from relaxation. Conversely, a large signed persistence would not establish sound regulation. A drifting or poorly controlled level can also generate strong positive serial dependence.

DISCUSSION

Autocorrelation profiles provide a way to examine how process behavior changes over time. The descriptors in (2) help identify whether a signal becomes more persistent, loses an oscillating pattern or behaves differently after differencing. This may be useful when examining slow process responses, sudden command changes or sensor irregularities. However, correlation alone cannot explain why these changes occur.

Similar patterns may result from attacks, normal operating changes, equipment faults or measurement problems. The method also examines each signal separately and may miss relationships between process variables. Lagged cross-correlation could provide additional information, although it would require further testing. The datasets also differ in sampling rates, noise

levels and operating conditions. Using the same number of lags would therefore cover different time periods. Parameters should be selected using normal data before testing, with results evaluated separately for each dataset and simulation run. A common detection threshold would not be appropriate without further validation.

The present analysis is limited to published dataset information and theoretical AR(1) calculations. These establish how the descriptors behave mathematically but do not show their performance on actual industrial measurements. Changes in ACF, detection accuracy and alert delays must be measured using the original HAI, BATADAL and Tennessee Eastman data before drawing experimental conclusions.

CONCLUSION

The paper specifies a reproducible way to treat temporal correlation structure as an operational descriptor of industrial dynamics. It joins level and increment autocorrelation, robust channel aggregation, source-specific time horizons and separate normal reference/calibration partitions. A documentary comparison of HAI 23.05, BATADAL and classical Tennessee Eastman shows why sampling and annotation conventions must shape the experiment. In particular, BATADAL Training Dataset 2 has 219 explicitly labeled hours out of 492 published attack-hours, precluding the automatic use of all unlabeled intervals as normal data. An analytical AR(1) check confirms that absolute and signed persistence encode distinct response characteristics. Measured performance on the three full archives remains to be established through a separately executed, auditable evaluation.

DATA AND CODE AVAILABILITY

The public data sources are HAI 23.05 [3], the BATADAL development data and attack schedule [5], and the classical Tennessee Eastman reference archive [8]. For HAI 23.05, the official repository requires Git LFS to retrieve the telemetry files. Values in Tables 2 and 3 follow from the published event schedule and the declared AR(1) identity. Raw HAI, BATADAL and TEP traces were not downloaded or processed for this draft. A companion source script supports the specified diagnostics once the original files are supplied.

REFERENCES

[1] Box, G. E. P., G. M. Jenkins, G. C. Reinsel, G. M. Ljung. Time Series Analysis: Forecasting and Control. 5th ed., Wiley, 2015. ISBN 978-1-118-67502-1.

[2] Ku, W., R. H. Storer, C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. Chemometrics and Intelligent Laboratory Systems, Vol. 30, No. 1, 1995, pp. 179-196. doi:10.1016/0169-7439(95)00076-3.

[3] Shin, H.-K., W. Lee, S. Choi, J.-H. Yun, B.-G. Min. HAI security datasets (HAI 23.05). Public project repository, 2023. https://github.com/icsdataset/hai.

[4] Shin, H.-K., W. Lee, J.-H. Yun, B.-G. Min. Two ICS security datasets and anomaly detection contest on the HIL-based augmented ICS testbed. CSET, 2021, pp. 36-40. doi:10.1145/3474718.3474719.

[5] BATADAL organizers. BATtle of the Attack Detection Algorithms: Development data, Training Dataset 2 attack schedule. 2016-2017. https://www.batadal.net/data.html.

[6] Taormina, R., S. Galelli, N. O. Tippenhauer, et al. Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks. Journal of Water Resources Planning and Management, Vol. 144, No. 8, 2018, 04018048. doi:10.1061/(ASCE)WR.1943-5452.0000969.

[7] Downs, J. J., E. F. Vogel. A plant-wide industrial process control problem. Computers & Chemical Engineering, Vol. 17, No. 3, 1993, pp. 245-255. doi:10.1016/0098-1354(93)80018-I.

[8] Large Scale Systems Research Laboratory, University of Illinois. Tennessee Eastman Process reference simulation data (d00, d01-d21 and test files), Prof. R. D. Braatz archive; public mirror. https://github.com/jkitchin/tennessee-eastman-profbraatz/tree/master/data.

[9] Kitchin, J. R. Tennessee Eastman process - hard faults notebook, published data and fault-onset convention. https://kitchingroup.cheme.cmu.edu/tep-rust/notebooks/03-hard-faults.html.