AVEVA PRiSM Overall Model Residual (OMR) Analysis in Equipment Health Monitoring

Core Algorithmic Approach of OMR in AVEVA PRiSM

Proprietary APR Modeling (OPTiCS Algorithm): AVEVA PRiSM (part of AVEVA Predictive Analytics, formerly Schneider/Avantis PRiSM) uses a proprietary advanced pattern recognition algorithm (branded **OPTiCS**) rather than a standard off-the-shelf PCA or PLS method [1]. The OPTiCS algorithm employs machine-learning techniques to learn an asset's **"unique operating profile"** across all normal operating regimes (covering different loads, ambient conditions, etc.) [2]. In essence, PRiSM ingests historical time-series data and automatically organizes it into *operational profiles* or clusters of normal behavior [3]. This data-driven model captures the typical interrelationships among all the sensor signals in the asset's system, effectively creating a multivariate signature of normal operation (often termed an "asset signature" by AVEVA) [4].

Sensor Interdependencies: Unlike simple threshold alarms on individual tags, PRiSM's model is multivariate. It leverages correlations between sensors – **each sensor's expected value is computed based on the others** under similar historical conditions [5]. In other words, PRiSM continually predicts what each sensor *should* read (the *Predicted Value*) given the current readings of all the other related signals in the model [5]. This predicted value is essentially drawn from patterns observed in historical data **at similar operating states** (e.g. same ambient temperature, load, process state, etc.), which is why PRiSM's approach inherently accounts for different regimes. If the current combination of inputs doesn't exactly match a known profile, the algorithm finds the closest matching profile/cluster of normal data [6] and uses that to estimate expected behavior. This approach is sometimes described as *pattern matching* on historical data (conceptually akin to a k-nearest-neighbor or clustering technique in multivariate space, rather than a single global regression equation).

Overall Model Residual (OMR) Calculation: The **OMR** is a single composite anomaly metric computed from all the individual sensor deviations. PRiSM calculates each sensor's **Absolute Deviation** as the difference between its actual value and the model-predicted value at that moment [7]. To normalize across sensors, a **Relative Deviation** is obtained (the absolute deviation divided by that sensor's training range) [8]. The OMR is then defined as the root-mean-square (RMS) of all sensors' relative deviations [8]. In formula form, if Δ_i are the relative deviations of each sensor *i*, then:

$$\mathrm{OMR} = \sqrt{\frac{\Delta_1^2 + \Delta_2^2 + \cdots + \Delta_n^2}{n}} \times 100\%,$$

essentially an aggregated "distance" of the asset's current state from its learned normal profile [9]. (This metric is analogous to the multivariate **squared prediction error (SPE)** or Q-statistic used in PCA-based anomaly detection, except PRiSM uses its proprietary modeling instead of explicit principal components.)

Use of PCA, PLS, or Regression: AVEVA's documentation does not explicitly mention using PCA or PLS algorithms. Instead, the term **"Advanced Pattern Recognition (APR)"** is used, implying a custom combination of techniques. In industry practice, APR often includes methods like auto-associative modeling or PCA-like analysis, but AVEVA's OPTiCS engine appears to go beyond a basic PCA. It likely performs nonlinear or clustered modeling of the normal data. For example, the clustering of historical data into operational profiles is built-in, whereas a pure PCA model would have to either include additional variables or separate models to handle different regimes. Likewise, PRiSM's approach of predicting each tag from others is conceptually similar to a set of multiple linear regressions or an autoencoder network that reconstructs inputs – indeed AVEVA offers a neural-network based plugin (called **KANN**) for cases with highly complex or noisy processes [10]. In summary, **PRiSM's OMR is generated via a proprietary multivariate model** (not a standard PCA/PLS library implementation) tailored to capture sensor interdependencies and normal patterns across all expected operating conditions [4]. This model continuously adapts to different modes (load changes, ambient changes) by recognizing the appropriate profile of normal behavior, rather than applying one static linear formula for all situations.

Regime-Based Profiles: A key strength of PRiSM is that it inherently handles **regime changes**. During training, the software learns the asset's behavior under various seasons, loads, and process states [2] [3]. The historical data is effectively segmented into clusters or "profiles" of similar operation. At run-time, PRiSM determines which profile the current conditions most closely match and uses the corresponding learned pattern to compute expected values [6]. This can be seen as *regime-based filtering*: rather than one global model trying to fit all conditions (which could smear or average out important differences), PRiSM's advanced pattern recognition organizes multiple normal operating regimes. This approach improves prediction accuracy and anomaly detection sensitivity, since the "normal baseline" dynamically shifts to match the context (e.g. what's normal at full load on a summer day may differ from normal at part load on a winter night). Many academic approaches address this by building separate PCA models for different regimes or adding regime variables; PRiSM accomplishes it within its automated modeling process by creating these operational profiles [3]. Thus, PRiSM's OMR algorithm explicitly incorporates regime-based training, which is a proprietary variant beyond classical PCA/PLS.

In summary, **PRiSM's OMR is a robust multivariate residual** indicating deviation from normal, powered by a proprietary APR algorithm. It is *not simply a PCA* or PLS model, though it serves a similar purpose to those techniques by computing a single health index from many sensor inputs. It indeed **models sensor interdependencies** (each sensor's normal is a function of the others) and **accounts for different operating regimes** via learned profiles [4] [3]. The OMR is essentially the combined error between actual and expected sensor values across the system – a higher OMR means the asset as a whole is behaving anomalously compared to its historical norm [11] [12].

OMR Outputs and Visualization in PRiSM

AVEVA PRiSM provides a rich web-based interface for monitoring models, with multiple visualization tools to display the OMR and related diagnostics. The **Overall Model Residual** itself is presented as a time-series trend and is the primary health indicator for each asset model [13]. Below are the key charts and visualization features related to OMR and anomalies:

- **OMR Trend Chart (Timeline):** The OMR is plotted over time as a single trend line (typically in percentage or normalized units) indicating the model's deviation from normal. This chart allows engineers to see the evolution of the asset's overall residual. When the OMR exceeds predefined thresholds, it triggers alerts and is visually highlighted. For example, if an OMR alert threshold is set, the chart will indicate when the OMR crosses that limit (often via color shading or markers on the timeline). In PRiSM's web UI, shaded bars above the OMR chart denote the alert status: no bar when OMR is below warning threshold, a blue bar when OMR exceeds a warning threshold, and purple if a critical threshold is exceeded [14]. This gives a quick visual cue of anomaly periods on the timeline. Only high (deviation) thresholds are typically used for OMR since it's a distance metric (low OMR simply means "normal"). Users can zoom and pan on this timeline to inspect when and how the residual started rising, helping to pinpoint onset of abnormal behavior.

- **Individual Sensor Trends (Actual vs Predicted):** For detailed analysis, PRiSM lets users open trend charts for each sensor (point) in the model. Each sensor's chart can display the **actual reading vs. the model's predicted value** over time [15]. The actual data is typically plotted in real-time (or near-real-time), and the predicted (expected) value is overlaid as a reference (often a smooth line reflecting what the sensor *should* be if the equipment were behaving normally) [15]. This visual overlay immediately shows which parameters are deviating from expected – a gap between actual and predicted indicates an anomaly in that parameter. Engineers use these plots to identify which sensor(s) are driving the high OMR. The UI allows toggling the predicted curves on/off for clarity [15].

- **Residual/Deviation Charts:** Directly under each sensor's trend, PRiSM can display the **deviation** of that sensor. Both **Absolute Deviation** (actual minus predicted in engineering units) and **Relative Deviation** (deviation normalized as a percentage of that sensor's normal range) can be shown as their own traces [16]. These are essentially the individual residual signals. Plotting deviations over time helps in spotting subtle drifts or intermittent spikes that may not be obvious in raw sensor trends but accumulate into the OMR. The relative deviation is effectively a normalized anomaly score per sensor – conceptually similar to a z-score, it tells how large the deviation is relative to historical variation (in PRiSM's case normalized by range) [8]. By examining which deviation trends are growing, users can identify the offending variable. PRiSM's interface allows you to hover on the deviation plot to see exact values at a timestamp [17], and you can zoom into specific time windows for a closer look at anomaly onset.

- **Signal Contribution Bar Chart:** A particularly powerful diagnostic view is the **contribution chart** for OMR. PRiSM can break down the overall residual at a given moment into contributions from each sensor (often computed as the squared normalized deviation of each signal as a fraction of the sum of squares). When the **"Signal Contributions"** toggle is activated, the interface displays a bar graph for each sensor (project point) showing how much that point is contributing to the current OMR value [18]. Essentially, it ranks the sensors by their percentage contribution to the total model error. This is typically shown as a horizontal bar for each tag, and one can hover over a bar to see the sensor name and its contribution percentage [19]. In practice, a single sensor with a large deviation will show a tall bar, immediately flagging it as a likely culprit. If multiple sensors each show moderate contributions, it could indicate a broader issue or multi-variable drift. Engineers use this contribution plot to quickly focus their investigation: for example, a high OMR alert might be predominantly due to one temperature reading drifting high (its bar might be, say, 60% of the OMR), whereas other sensors contribute less. PRiSM even allows clicking on a bar to isolate that sensor's trend for detailed analysis [20]. This contribution visualization is analogous to the **contribution plots** used in PCA-based fault diagnosis, providing a quick breakdown of which variables are driving the anomaly.

- **Dashboards and Alerts:** On a higher level, PRiSM's **Home/Alerts dashboard** gives an at-a-glance view of all assets and their status. Typically, the home screen shows a list of models with their current health (OK, warning, or alarm) and counts of alerts [21]. The OMR can serve as the "health index" displayed – for example, some implementations show a gauge or percentage for the current OMR against thresholds. Users performing daily monitoring can see if any model's OMR is yellow or red (exceeding limits) [22]. They can then drill down by clicking that asset to open the detailed Trends view. This serves as a dashboard for anomaly monitoring across the plant. In addition, PRiSM supports email or notification alerts when OMR thresholds are breached, but those are outside the visualization scope.

- **Historical Trends and Timelines:** PRiSM's trend view allows plotting lengthy time windows to see how anomalies develop over days/weeks. The OMR timeline coupled with individual sensor timelines can reveal patterns such as a slow deterioration (OMR gradually rising over weeks) or sudden events (an abrupt spike). Users can adjust time ranges, and PRiSM offers features like *event markers* on the OMR chart to note when model configuration changes or resets occurred [23]. This helps correlate any changes in the model or asset with anomaly trends.

- **Heatmaps:** While AVEVA's documentation doesn't explicitly mention a heatmap feature for PRiSM, the combination of contribution bars over time can conceptually be viewed similar to a heatmap. In some analytics tools, one might see a matrix of sensors vs. time with color intensity showing deviation. PRiSM's approach is more interactive: you examine contributions at specific times or over the current time span. If needed, an engineer could mentally assemble a heatmap by looking at contribution bars at successive timestamps. However, the primary visual tools remain line charts and bar plots rather than an automated heatmap. (AVEVA Insight, a different product, does have anomaly heatmaps, but in PRiSM the focus is on trends and contributions.)

- **Fault Diagnostics (Advanced Visualization):** In addition to raw residual analysis, PRiSM/Predictive Analytics offers a *Fault Diagnostics* module where specific failure modes are defined by patterns of deviations (fault signatures). When this is configured, the UI presents **doughnut charts** indicating the match of current data to known fault signatures, and a **fault tree visualization** linking sensor deviations to probable fault causes [24] [25]. For example, if a certain combination of sensor residuals matches a known "pump wear" signature, a colored doughnut chart will show a high match percentage. This goes beyond OMR by diagnosing likely failure modes. The fault tree view graphically shows relationships: it connects sensor metrics -> fault signatures -> fault diagnostics (problems) in a hierarchical chart [25]. While this is slightly beyond pure OMR, it illustrates how PRiSM visualizes not only that an anomaly occurred (via OMR) but also **why**, by mapping sensor patterns to fault libraries. These advanced visuals (signature match bars, percentages, and tree diagrams) greatly aid in root-cause analysis once an OMR alert is raised.

Overall, **PRiSM's interface is designed for quick anomaly detection and diagnosis**. The **OMR trend** gives a single-glance indication of asset health, the **sensor trend plots** and **deviation plots** show the detailed behavior versus expected, and the **contribution bar chart** highlights the likely trouble variables in real time [18]. All charts are interactive – users can zoom in, overlay data, and even export the views. The combination of timeline views and contributions effectively serves the same role as academic tools like SPE plots and contribution plots in PCA, but in a more user-friendly, integrated dashboard.

(Note: The user guide indicates that these charts are highly configurable; users can choose which ones appear by default. For instance, one can configure the Trends view to always show OMR and Signal Contributions on top, and Actual/Predicted/Deviation plots for individual sensors below [26] [27].)

Comparison to Academic Approaches (PCA/PLS) and Additional Capabilities

Methodology Parallels: The methodology used by AVEVA PRiSM for anomaly detection is broadly aligned with techniques well-known in academic literature, but with practical enhancements. In academia, a common approach is to use **Principal Component Analysis (PCA)** on historical normal data to build a model of correlation, then monitor the **Squared Prediction Error (SPE)** (also called Q statistic) as an anomaly indicator. PRiSM's OMR metric is conceptually analogous to the SPE – it is the aggregated residual between the high-dimensional sensor data and the model's expected output. In fact, researchers sometimes use the term "overall model residual" for the Mahalanobis-distance or PCA reconstruction error used to flag anomalies [28]. Both PCA-SPE and PRiSM-OMR signal an anomaly when the combination of sensor readings no longer fits the learned normal subspace.

However, PRiSM's implementation goes beyond a basic PCA in a few ways:

- **No Manual Modeling Required:** In PCA, engineers must often decide how many principal components to retain, possibly build separate models for different modes, and set statistical control limits. PRiSM automates the modeling – it uses its OPTiCS algorithm to internally determine the patterns. There is no need for the user to specify PCA vs PLS or tweak regression equations; the software's machine learning finds the optimal representation of normal behavior. AVEVA describes it as "no-code" modeling – users feed in historian data and the software builds the model automatically [29] [30].

- **Nonlinear/Regime Handling:** Classical PCA is linear and struggles with nonlinear relationships or multimodal operating data unless advanced techniques (kernel PCA, multiple models) are used. PRiSM inherently captures *all loading and ambient conditions* in one solution by creating those operational profiles [2]. This is akin to having an ensemble of PCA models for various regimes, or using clustering first – tasks an engineer would have to do manually in an academic setting. PRiSM's algorithm ensures the residual (OMR) remains sensitive even as conditions change, whereas a single PCA model might either alarm too often in one regime or miss anomalies in another if not tuned carefully.

- **PLS and Supervised Models:** Partial Least Squares (PLS) is typically used for predicting a specific output (e.g. product quality) from many inputs, rather than unsupervised anomaly detection. PRiSM's context is unsupervised (no specific "Y" to predict, just all sensors predicting each other). So PLS per se is not the technique here – PRiSM is more analogous to **auto-associative modeling** (where every sensor is predicted from the others). In academic terms, one might build multiple linear regression models or use an autoencoder neural network to reconstruct inputs; PRiSM's engine likely uses similar principles internally (indeed their *KANN* neural-net option confirms the use of auto-associative nets for certain cases [31]). It may also incorporate regularization (similar to Ridge Regression) to prevent overfitting the training data – while not stated explicitly, most industrial APR tools do something to ensure the model is robust to noise.

- **Real-Time Monitoring:** A critical difference is that PRiSM is a **real-time online system** out-of-the-box. The models run continuously on live data streaming from the plant historian. The software computes OMR and deviations at the chosen frequency (often one value per sensor per minute or hour, configurable) and generates alerts immediately when thresholds are breached [32]. Academic studies often apply PCA or machine learning to historical datasets or in simulated real-time; PRiSM is built into plant operations, analyzing data continuously in near-real-time. For example, as soon as a new sensor reading comes in, PRiSM updates the predicted values and OMR, so emerging issues are caught with minimal delay [33]. This real-time aspect is emphasized by AVEVA – the system provides **continuous monitoring and early warning** of equipment problems, often detecting subtle deviations weeks or months before failure [1].

- **Handling Missing or Bad Data:** Industrial datasets often have gaps or bad-quality readings. PRiSM has built-in data preprocessing to handle this. According to the user guide, the software can automatically **filter out "bad data"** (e.g. sensor offline, stuck, or flagged as invalid by the data source) so that these points do not skew the model residual [34]. It even attempts to recover short data dropouts (e.g. by interpolation) up to a certain time limit to avoid false alarms [35]. This is crucial for practical use – a PCA model in a paper might simply omit missing points or assume perfect data, but a live system must robustly deal with real-world data quality issues. PRiSM allows configuration of how to treat bad data (hide it, mark it on charts, etc.) [36]. By concealing bad inputs from the calculations, PRiSM ensures the OMR doesn't spike just because a sensor failed or transmitted a glitch; at the same time, it provides a separate **"Sensors" view to highlight sensor health issues**, so that what it filters out in OMR can still be seen by maintenance teams if needed [34]. In summary, missing data is handled gracefully: small gaps are bridged, longer gaps lead to points being ignored (and possibly an alert that the sensor itself has an issue), rather than triggering false positive OMR alarms.

- **Model Updating and Retraining:** Over time, an asset's normal behavior can change (due to aging, upgrades, etc.), so the predictive model should be updateable. PRiSM supports both manual and assisted retraining of models. Users can initiate a **"Profile Retrain"** using more recent historical data if they believe the definition of "normal" needs refreshing [37]. For instance, if an equipment has been overhauled or the process conditions have shifted, the engineer can select a new date range of healthy operation and retrain the model (either replacing or adding to the existing profiles). This is done via a UI workflow with no coding: one selects the project and specifies the training time window and data frequency, then the software rebuilds the model profiles with the new data [37]. PRiSM even supports **scheduled retraining** or continuous learning modes in some versions, though typically changes in the model are controlled by the user to avoid "learning" an abnormal state by mistake. By contrast, academic implementations of PCA might assume a fixed model or periodically recompute it offline. PRiSM provides a structured way to keep the model accurate as the asset evolves, including versioning of profiles and marking on the OMR trend when a model revision occurred (those event markers noted earlier) [23]. This ensures traceability – one can see if a drop in OMR is because the model was retrained to new baseline, etc.

- **Diagnostic Guidance:** Pure academic methods like PCA will tell you an anomaly occurred and maybe which variables contribute, but interpreting the root cause is left to the engineer. PRiSM augments the process with its **Case management and Fault Diagnostics libraries**. When an OMR alert is generated, users can log it as a case, add observations, and if configured, see probable failure modes identified via the fault signature matching (the doughnut charts mentioned above). This prescriptive guidance ("what does this combination of residuals likely mean and what actions to take") is a value-added layer on top of the raw analytics. It reflects industry best practices to not only detect a problem but also expedite the diagnosis and response.

In summary, **PRiSM's approach aligns with multivariate anomaly detection principles from academia** (model normal, detect deviations), but it is implemented in a robust, automated way for industrial use. It incorporates the strengths of PCA/PLS (capturing correlations, providing a global residual and contribution metrics) while addressing their limitations (nonlinear profiles, data noise, operationalization in real-time). One can think of OMR as analogous to the **PCA Q-statistic (SPE)** or an **autoencoder's reconstruction error** [28] – a single metric for "distance from normal", and the contribution bars analogous to PCA's contribution plot for fault variables. PRiSM supports real-time scoring (continuous monitoring) by integrating with historians and providing live updates on the OMR and sensor deviations [33]. It handles missing or bad data through preprocessing filters to ensure reliability of alerts [34]. And it allows model maintenance through retraining on new data when necessary to keep the "normal" baseline current [37]. These capabilities make it a comprehensive industrial solution, building upon the foundation that has been well-established in academic research, but adding layers of practicality and user-friendliness (dashboards, alerts, case management, etc.). As a result, companies using AVEVA PRiSM have reported detecting faults **days to months in advance** with high confidence, something that hinges on the accurate OMR analytics combined with effective visualization and diagnostics [38] [39].

Sources:

- AVEVA (Schneider Electric) Whitepapers and Brochures on Predictive Analytics and PRiSM [40] [10] [41]
- AVEVA Predictive Analytics Training Materials (User Concepts and Monitoring Guide) [42] [8] [18]
- AVEVA Predictive Analytics Web User Guide (2022) [15] [16] [37]
- Academic reference on anomaly detection using model residuals (for conceptual comparison) [28].

---

[1] [2] [4] [39] [40] gulf.avevaselect.com
https://gulf.avevaselect.com/wp-content/uploads/2023/03/WhitePaper-AVEVA-Predictive-Analytics-In-Oil-and-Gas.pdf

[3] [5] [6] [7] [8] [9] [12] [13] [32] [33] [42] 1.AVEVA Predictive Analytics Training - Intro To User Concepts | PDF | Analytics | Predictive Analytics
https://www.scribd.com/document/606244620/1-AVEVA-Predictive-Analytics-Training-Intro-to-User-Concepts

[10] [31] 3 & 8. AVEVA Predictive Asset Analytics | PDF | Analytics | Reliability Engineering
https://www.scribd.com/document/579887241/3-8-AVEVA-Predictive-Asset-Analytics

[11] [38] [41] APM 4.0 with Predictive and Prescriptive Analytics
https://engage.aveva.com/rs/986-YIS-805/images/WhitePaper_AVEVA_APM-PredictiveandPrescriptiveAnalytics_22-08.pdf

[14] [15] [16] [17] [18] [19] [20] [23] [24] [25] [26] [27] [34] [35] [36] [37] AVEVA Predictive Analytics Web User Guide | PDF
https://www.scribd.com/document/768161438/AVEVA-Predictive-Analytics-Web-User-Guide

[21] [22] 4.AVEVA Predictive Analytics Training - Monitoring | PDF | Predictive Analytics | Analytics
https://www.scribd.com/document/606244623/4-AVEVA-Predictive-Analytics-Training-Monitoring

[28] TF_Template_Word_Windows_2016
https://arxiv.org/pdf/2509.12372

[29] [30] AVEVA™ Predictive Analytics for AVEVA™ PI System™
https://gulf.avevaselect.com/wp-content/uploads/2023/03/Brochure-AVEVA-Predictive-Analytics.pdf