AI-01971 Yaskawa Robotics

Owner	Ⓡ Rigel Arcayan
Tags	Project
Created time	@August 15, 2025 10:04 PM

A Blueprint for Data-Driven Maintenance: Harnessing Motor Performance from Yaskawa Robots

Part I: The Strategic Imperative of Data-Driven Maintenance for Yaskawa Robots

1.1 The Shift from Preventative to Predictive Maintenance

The operational efficiency of industrial automation hinges on minimizing downtime and maximizing asset lifespan. For decades, the industry has relied on time-based preventative maintenance schedules, such as servicing a robot's reducers every 6,000 servo hours, a practice specifically recommended for Yaskawa robots. This approach, while effective at a basic level, often leads to unnecessary maintenance, which can be costly and disruptive, or fails to prevent an impending breakdown that occurs outside of the scheduled interval. Modern industrial strategy is shifting from this reactive or interval-based model to a data-driven, predictive model. This more advanced approach, often referred to as condition-based maintenance, leverages real-time operational data to forecast potential failures and enables maintenance activities to be conducted precisely when they are needed, rather than on a rigid schedule.

This transition is a core component of Yaskawa's "i^3-Mechatronics" vision, which combines traditional mechatronics with information, communication, and digital technologies such as the Internet of Things (IoT). This strategy is designed to achieve "zero downtime" by anticipating equipment malfunctions due to wear and tear. By adopting this data-centric philosophy, manufacturers can achieve significant business advantages, including reduced unplanned downtime, optimized spare parts inventory, and a prolonged operational life for high-value robotic assets. The user's request for a system to collect and analyze motor data aligns directly with this strategic evolution, positioning it as a fundamental step toward achieving a more resilient and efficient manufacturing operation.

1.2 The Role of Motor Data as a Leading Health Indicator

The motors within a Yaskawa robot are not merely power sources; they are a direct conduit to the mechanical health of the manipulator's joints and systems. Key metrics such as motor torque, speed, and temperature are leading health indicators that provide critical foresight. For instance, servo motors continuously monitor sensing data like vibration, disturbance, and temperature to detect real-time signs of failure, such as equipment aging or changes in the operating environment.

A gradual increase in a motor's steady-state torque, while the robot performs a consistent task, can signal increased friction and wear in the corresponding joint's reducer. Similarly, a rise in motor temperature may indicate increased load or inefficient operation. By systematically monitoring these parameters over time, subtle changes and trends can be identified long before an operational alarm is triggered on the controller. This allows for proactive intervention, such as a torque analysis or a timely reducer replacement, which prevents catastrophic failure and avoids the associated costs of unplanned downtime. Visualizing this data, as demonstrated by the Yaskawa Cockpit's ability to compare normal and abnormal values for root cause analysis , transforms raw metrics into actionable intelligence for improved quality and maintenance.

Part II: Foundational Methods for Yaskawa Robot Data Access

2.1 The In-situ Approach: Native Teach Pendant Diagnostics

The Yaskawa teach pendant (TP) is the most immediate and accessible tool for interacting with a robot and performing on-the-spot diagnostics. It provides real-time monitoring of various data points, including each axis's current position, I/O status, and system alarms. For a technician troubleshooting a specific, localized issue, the teach pendant's built-in "Oscilloscope Function" is a powerful native feature. This function allows the user to view real-time waveforms for motor performance, including servo status and other data, directly on the pendant's screen.

While invaluable for manual diagnosis, the teach pendant has fundamental limitations that prevent it from being a solution for continuous, fleet-wide monitoring. The physical tether of the pendant cable restricts the user's mobility to a limited area around the controller. Furthermore, for safety reasons, a technician cannot use the remote pendant function to turn on the servo power or start a job. These limitations necessitate a networked solution that can extract data remotely and continuously, which is the core driver for the user's inquiry.

2.2 The Commercial Pathway: Yaskawa's Software Suites

Yaskawa has developed its own suite of commercial software to meet the demand for advanced monitoring and maintenance. **Yaskawa Cockpit** is the company's proprietary Industrial Internet of Things (IIoT) platform, designed to provide a "360-degree view of your robot operations". This central software platform connects to multiple devices and visualizes their health, status, and performance in real time. It is built on a scalable database that collects and analyzes data to identify preventive maintenance needs, predict component lifespans, and automatically distribute system alarms. The platform is a direct response to the market's need for a fleet-wide monitoring solution, a need that the teach pendant cannot fulfill.

Another proprietary solution is **RobotPro**, a PC-based expert maintenance package. While it also supports robot maintenance, its focus is on providing guided, step-by-step repair and maintenance procedures, including preventive maintenance schedules, checklists, and parts lists. It serves more as a comprehensive repair manual and troubleshooting tool, whereas Yaskawa Cockpit is a real-time data collection and visualization platform. The existence of these paid, proprietary solutions confirms that Yaskawa has made significant investments in data-driven asset management, validating the user's goal and demonstrating that the company recognizes the value of this data.

2.3 The IIoT Gateway: Industrial Communication Protocols

To move beyond the limitations of the teach pendant and the vendor-specific solutions, a standardized, robust communication protocol is required to extract robot data remotely. OPC Unified Architecture (OPC UA) has emerged as the leading standard for this purpose in industrial environments. OPC UA is a cross-platform, open-source protocol that enables secure, end-to-end communication between industrial machines and the cloud.

A key advantage of OPC UA is its use of a structured "information model," which is central to its utility. Instead of simply transferring raw, uncontextualized data, the protocol uses a standardized, semantically defined data format. For example, the protocol specifies how to interpret and expose data points for "motor temperature," "torque," or "run time" in a consistent manner, regardless of the specific robot model or manufacturer. This significantly simplifies integration, as a third-party client application can connect to a Yaskawa robot's OPC UA server and know precisely what data it is receiving without complex, custom data mapping. The OPC UA server on Yaskawa controllers, which is a paid software option, allows clients to read variables, access state data such as motor temperatures, and even control the robot remotely. This is the recommended pathway for building a custom, vendor-agnostic IIoT solution.

While Yaskawa controllers also support other protocols like EtherNet/IP and Modbus TCP/IP , these are primarily for general device communication. They typically lack the rich, granular, and semantically-defined data structure provided by the OPC UA information model, making them less suitable for sophisticated, data-driven applications. The choice of OPC UA as the foundational protocol is a critical architectural decision that simplifies the entire downstream data management and analysis process.

Part III: The End-to-End IIoT System Architecture

3.1 The Reference Architecture: A Data-Centric Blueprint

Developing a system for continuous data charting and analysis requires a structured approach. The following reference architecture outlines the necessary components, from the data source to the user interface:

1. **Data Source:** The Yaskawa robot controller (e.g., YRC1000, DX200) serves as the origin of all motor performance and operational data.

2. **Protocol & Gateway:** The OPC UA Server acts as the secure, standardized gateway for extracting this data from the controller.
3. **Data Ingestion:** An OPC UA client application or middleware connects to the robot controller to pull the data.
4. **Database:** A Time-Series Database (TSDB) is the optimal choice for storing the high-volume, time-stamped data from the robots.
5. **Visualization:** A dashboarding platform provides the user interface for charting, analyzing, and alerting based on the data in the TSDB.

3.2 Phase 1: Controller-Side Preparation

The initial phase involves preparing the Yaskawa controller to expose its data via OPC UA. The OPC UA server is a paid software option that can be obtained from Yaskawa. Once acquired, the process is as follows:

1. **Installation:** Install the provided MotoPlus app and license onto the robot controller. This software enables the controller to function as an OPC UA server.
2. **Configuration:** Utilize the Yaskawa OPCUA Config Editor, a separate utility, to define which specific data points (e.g., motor torque, speed, run times) will be exposed by the server. The editor generates a `.JBI` configuration file.
3. **Deployment:** Load the `.JBI` file onto the controller to apply the configuration.
4. **Connectivity:** Ensure the controller's network settings are properly configured with a static IP address and subnet mask to enable communication with the network.

3.3 Phase 2: Data Ingestion and ETL

With the controller configured as an OPC UA server, the next step is to get the data into a database. This requires an ingestion layer, which is typically a third-party OPC UA client application. The process follows a standard Extract, Transform, Load (ETL) model:

- **Extract:** The client application connects to the robot's OPC UA server and continuously pulls the defined data points. This process can be configured to pull data at a specific frequency to balance the need for real-time information with network bandwidth.

- **Transform:** As the data is ingested, it is prepared for storage. This may involve adding a universal timestamp, normalizing units, or adding metadata (e.g., robot serial number, cell name) to each data point for easier analysis later.
- **Load:** The transformed data is then loaded into the selected time-series database.

3.4 Phase 3: Database Selection and Implementation

For high-volume, time-stamped, and sequential data from a robotic fleet, a specialized Time-Series Database (TSDB) is the optimal choice over a traditional SQL database. TSDBs are engineered for this specific data model, offering superior ingestion speed, storage efficiency through compression, and query performance for time-based analysis. The choice of TSDB is a critical architectural decision that impacts scalability and the complexity of future analysis.

3.5 Phase 4: Visualization and Dashboarding

The final phase transforms the raw data into actionable intelligence. A visualization platform connects directly to the TSDB to create a user-friendly interface. This interface can be used to generate real-time charts showing current motor performance, historical trend lines to identify gradual degradation, and dashboards that display key operational KPIs, such as run times. These platforms also offer powerful alerting capabilities that can notify personnel when a predefined threshold is exceeded, enabling a proactive response to potential issues.

Part IV: Comparative Analysis of Core Technologies

4.1 The Time-Series Database (TSDB) Decision

The market offers several powerful TSDBs, but a detailed comparison of two leading open-source options, TimescaleDB and InfluxDB, reveals key differences in their data models and performance characteristics. The choice between them often depends on a team's existing skill set and the anticipated complexity of their analytical queries.

TimescaleDB is built as an extension to PostgreSQL, which means it uses a familiar and powerful relational data model and is fully compatible with SQL. For a team with existing SQL expertise, this significantly reduces the learning curve and allows them to leverage a vast ecosystem of third-party tools and connectors. The relational model is more versatile, supporting complex queries like joins and window functions that are critical for analyzing data across multiple robots or correlating robot data with other manufacturing variables. TimescaleDB has also demonstrated superior performance with high-cardinality datasets (many devices or metrics) and complex queries.

InfluxDB utilizes a custom, non-relational NoSQL data model. It is optimized for high-write throughput and simple query scenarios with low data cardinality. While this may be easier for some to get started with, it uses a proprietary query language called Flux, which can introduce a learning curve for teams accustomed to SQL. It has also been shown to experience performance degradation and stability issues when dealing with high-cardinality datasets.

The following table provides a concise comparison to assist in the architectural decision-making process:

Table 4.1: Time-Series Database Comparison Matrix

Feature / DB	TimescaleDB	InfluxDB
Data Model	Relational, based on PostgreSQL	Custom NoSQL (tagset model)
Query Language	Full SQL	Proprietary (Flux)
High Cardinality	Excellent performance scaling	Performance can deteriorate
Complex Queries	Generally outperforms, supports joins and window functions	Supports a narrower range of queries, slower performance
Ecosystem	Inherits vast PostgreSQL ecosystem	Limited, fewer compatible tools
Reliability	Inherits PostgreSQL's 25+ years of engineering for fault tolerance	Must implement its own reliability features from scratch

4.2 The Visualization and Analytics Platform

After selecting a database, the final component is a visualization platform to transform the stored data into meaningful dashboards. The choice here is a strategic decision, often dictated by an organization's existing software ecosystem and budget.

Grafana is the de facto open-source standard for visualizing time-series data. Its key strength lies in its flexibility and extensive library of data source plugins, which allow it to connect to virtually any database, including both TimescaleDB and InfluxDB. Grafana provides a centralized, single-pane-of-glass view of data from a wide range of disparate sources, without the need to combine the data in a backend. This makes it the ideal choice for a team building a custom, open-source stack.

Power BI is an enterprise-grade solution that is particularly well-suited for organizations already invested in the Microsoft ecosystem. It offers seamless, native integration with Azure IoT Hub and Azure Stream Analytics, which can be configured to route data from an IoT source directly into Power BI dashboards. Power BI's drag-and-drop interface is user-friendly and enables non-technical users to easily create reports and dashboards.

Table 4.2: Visualization Platform Comparison

Platform	Licensing	Key Integrations	Ease of Use	Customization
Grafana	Open-source	Wide range of databases via plugins (TimescaleDB, InfluxDB, Prometheus, etc.)	Moderate; requires query language knowledge	High; extensive dashboard and panel options
Power BI	Commercial (Subscription)	Microsoft ecosystem (Azure IoT Hub, Stream Analytics, etc.)	High; drag-and-drop interface	Moderate; constrained by platform features

4.3 Real-World Application: The Yaskawa-Miller Integration Case Study

The integration of Yaskawa robots with Miller's Insight Centerpoint™ software provides a real-world example of a custom data-driven solution. This partnership demonstrates that Yaskawa robots are an open and capable platform for advanced data integration. In a welding application, Miller's software was used to monitor and log critical weld parameters such as amperage and voltage, and this data was stored in an SQL server database. This system enabled the robotic cell to track non-conforming parts, ensuring that safety-critical welds were within specified limits. This successful implementation confirms the viability of the proposed architecture and serves as a powerful precedent. The choice of an SQL database in this case also demonstrates that while TSDBs are ideal for performance, a traditional relational database can be a sufficient solution for certain applications, especially when integrated with existing enterprise systems.

Part V: Operationalizing the Data for Business Impact

5.1 From Data to Dollars: The Predictive Maintenance Model

The culmination of building an IIoT system is the transition from raw data to a valuable business asset. The data becomes a tool for proactive management rather than a historical record. The following steps outline how to operationalize the collected data for preventive maintenance:

1. **Baseline Data Collection:** After commissioning the system, a period of continuous operation should be used to collect "normal" operational data. This establishes a baseline for healthy performance against which future data can be compared.
2. **KPIs and Thresholds:** Key Performance Indicators (KPIs) should be defined and monitored. For motor performance, this includes measuring average motor torque, peak torque, and motor temperature over time. Gradual deviations from the established baseline, such as a steady increase in torque at a constant speed, can be a sign of impending mechanical issues like reducer wear. Thresholds can be set to flag these deviations automatically.
3. **Alerting:** The visualization platform's alerting capabilities should be configured to notify maintenance personnel when a KPI exceeds a predefined

threshold. This enables a proactive response, allowing a technician to investigate an issue before it leads to a total system shutdown.

5.2 Anomaly Detection and Proactive Troubleshooting

The dashboards created from the collected data can be a powerful tool for real-time diagnosis and root-cause analysis. Yaskawa Cockpit's ability to "visualize" equipment status and compare normal and abnormal data values demonstrates this capability. If a robot begins to produce defective parts or experiences inconsistent motion, a technician can immediately consult the dashboards. By analyzing a real-time graph of motor speeds, temperatures, or torque values, they can quickly diagnose a potential mechanical issue or a process variable drift that may have otherwise been difficult to identify. This empowers teams to perform data-driven troubleshooting, reducing the time spent on problem-solving, which accounts for a significant portion of industrial downtime.

5.3 Scaling and Enterprise Integration

The IIoT architecture outlined is inherently scalable and can be expanded from a single robot to an entire factory fleet. The use of a standard protocol like OPC UA simplifies the process of adding new devices and integrating their data streams. The final strategic step is to integrate the data from the TSDB into higher-level enterprise systems, such as Manufacturing Execution Systems (MES) or Enterprise Resource Planning (ERP) systems. Yaskawa Cockpit's ability to make all data available to the "extended enterprise" via OPC UA validates this as a critical strategic goal. This final integration unifies data from the shop floor with business-level systems, enabling comprehensive business optimization, from production scheduling and quality control to inventory management and supply chain logistics.

Conclusion

The pursuit of motor performance data from Yaskawa robots is a strategic move that aligns with the modern imperative of a data-driven manufacturing environment. The analysis confirms that a custom IIoT solution for continuous data extraction and analysis is not only feasible but also a powerful pathway to transitioning from reactive to predictive maintenance.

The most effective and scalable approach involves bypassing the limitations of native teach pendant diagnostics and leveraging a standardized, secure industrial protocol. OPC UA stands out as the superior choice due to its robust architecture and standardized information model, which simplifies the integration of data from Yaskawa robot controllers.

The system architecture can be developed by sequentially implementing four key phases: controller-side preparation with the OPC UA server, data ingestion, database implementation with a purpose-built Time-Series Database, and a visualization platform. For a team with existing SQL knowledge and a need for complex, fleet-wide analysis, a robust solution would involve pairing TimescaleDB for data storage with Grafana for visualization. For organizations already embedded in the Microsoft ecosystem, an alternative would be to leverage Power BI and its seamless Azure integration.

Ultimately, the goal is to transform motor data into actionable intelligence. By establishing a performance baseline, setting critical thresholds, and configuring automated alerts, a company can proactively identify and address potential issues. This prevents costly, unplanned downtime, extends the operational life of assets, and provides a clear competitive advantage in the modern industrial landscape.

Sources used in the report

robotsdoneright.comMaintaining Your Yaskawa Motoman Robot Opens in a new window metalworkingnews.infoYaskawa Cockpit software platform – Industry 4.0 in real time | Metalworking News Opens in a new window yaskawa-global.comPredictive Failure Diagnosis of Equipment | Yaskawa Global Site Opens in a new window yaskawa.comTeach Pendant - FOR MULTIAXES MOTION CONTROLLER PROGIC-8 - Yaskawa Opens in a new window inrobots.shopWhat is A Robot Teach Pendant - inRobots Opens in a new window uploads.strikinglycdn.comDx200 Operator'S Manual Opens in a new window knowledge.motoman.comRemote Operations with Yaskawa Motoman Robots Opens in a new window motoman.comYaskawa Cockpit - Industrial Monitoring IIoT Software for Robots ... Opens in a new window motoman.comRobot Maintenance & Troubleshooting Software - RobotPro Opens in a new window motoman.comRobot Technical Support - Yaskawa Support Services Opens in a new window knowledge.motoman.comOPCUA Server on Yaskawa Motoman

Controllers – Yaskawa ... Opens in a new window yaskawa.comEtherNet-IP - Yaskawa Opens in a new window yaskawa.comEtherNet-IP - Drives - Yaskawa Opens in a new window yaskawa.comModbus TCP-IP - Yaskawa Opens in a new window yaskawa.comModbus TCP-IP - Yaskawa Opens in a new window tdengine.comBest Time-Series Databases in 2025 | TDengine Comparison Opens in a new window iunera.comWhat are Open Source Time Series Databases? | iunera Opens in a new window grafana.comProduction node exporter | Grafana Labs Opens in a new window iiot-world.comReal-Time Visualization for IIoT Data Opens in a new window millerwelds.comInsight Welding Arc Data Monitoring Solutions | MillerWelds Opens in a new window tigerdata.comTimescaleDB vs. InfluxDB: Purpose-built for time-series data ... Opens in a new window influxdata.comCompare InfluxDB vs TimescaleDB Opens in a new window grafana.comIndustrial IoT visualization: How Grafana powers industrial automation and IIoT Opens in a new window learn.microsoft.comTutorial - IoT data visualization with Power BI - Azure IoT Hub ... Opens in a new window abbacustechnologies.comPower BI and IoT: Analyzing Sensor Data - Abbacus Technologies Opens in a new window grafana.comCreate a dashboard | Grafana documentation Opens in a new window last9.ioIntegrating OpenTelemetry with Grafana for Better Observability - Last9 Opens in a new window motoman.comRobotic Welding System Helps Chassis Supplier Exceed OEM Requirements Opens in a new window millerwelds.comCenterpoint Arc Data Monitoring Software | MillerWelds - Miller Welding

YASKAWA

YASKAWA

YASKAWA

IIoT World

Connecting the SMARTS!

YASKAWA