SOLUTION BROCHURE

Turbine Engine Test Systems

Modern approaches for testing increasingly complex engine systems

Turbine Engine Test Solution Advantages

- Modular architecture ensures the system can grow and evolve over time with minimal disruptions to test facilities
- Recommended measurement zone approach provides the right equipment for each signal type
- Data abstraction reduces the risk of being locked into any single solution or vendor
- Facility monitoring improves facility availability and efficiency
- Integrated approach to engine testing increases the value of the data obtained from the engine, measurement system, and facility

Contents

Application Overview

Test engineers tasked with testing new engine technologies face increasingly complex challenges. Engine technologies rapidly evolve; test engineers must ensure that their test capabilities keep up with these changes. This requires new distributed measurement technologies to get closer to the measurement, more capable instrumentation, and more complex software architectures. These technologies increase the value of the test data and ensure that test facilities operate at their maximum efficiency.

Turbine Engine Test Types

Whether testing engines for early research, for development, or for production, engine tests likely include some or all of these signals:

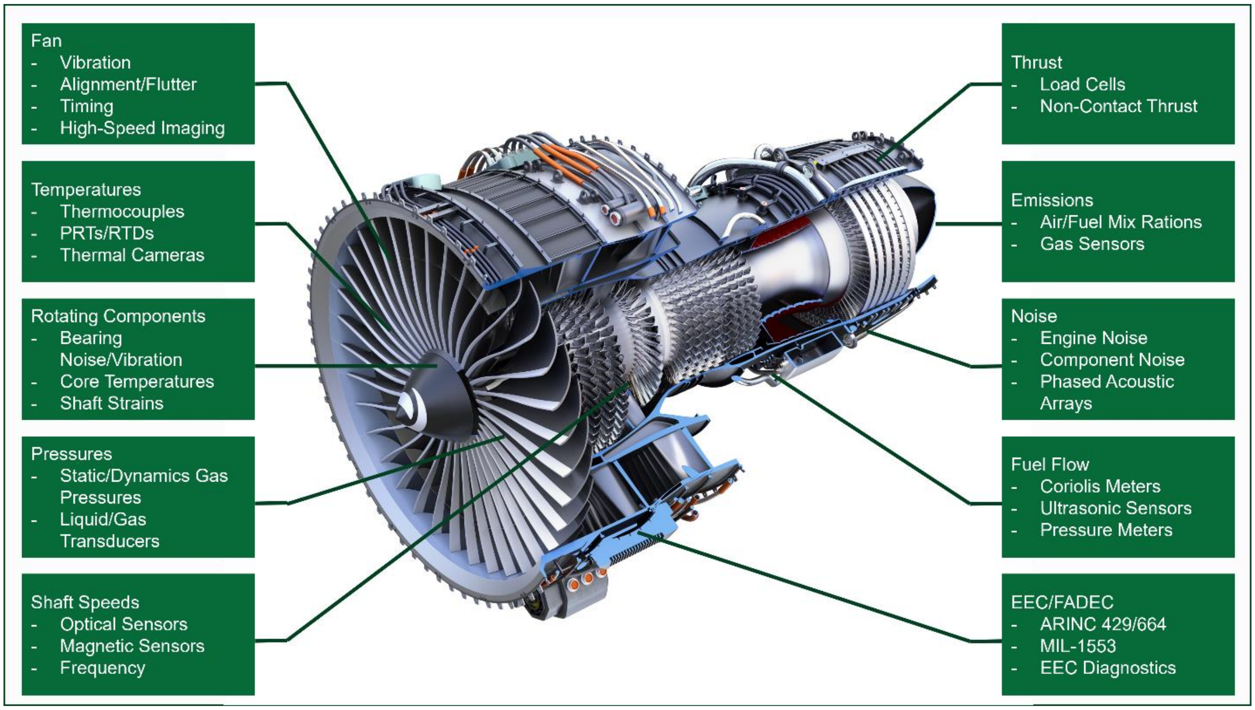


Figure 1: Engine Test Signals

Depending on the complexity of the engine, and the purpose of the tests, the number of channels in an engine test can range from a dozen to several thousand. Managing the test plans, facility capabilities, sensors, instrumentation, and control systems requires careful planning and test design.

Turbine Engine Test Facilities

A major challenge of testing these signals in an engine is to replicate the environment in which the engine will be used. The test facility must provide adequate air flow, fuel supply, and mounting resistance. When designing an engine test, engineers select from three primary test environments: indoor facilities, outdoor facilities, and flying test beds.

Indoor Engine Test Facility

Indoor facilities provide a controlled environment for the engine test. Engineers can control the air flow, the temperature, the fuel flow and mix, and many other details of the test environment. Measurement systems can track thousands of details of the engine during test. Indoor facilities vary in design and capabilities, but a standard facility may include these components:

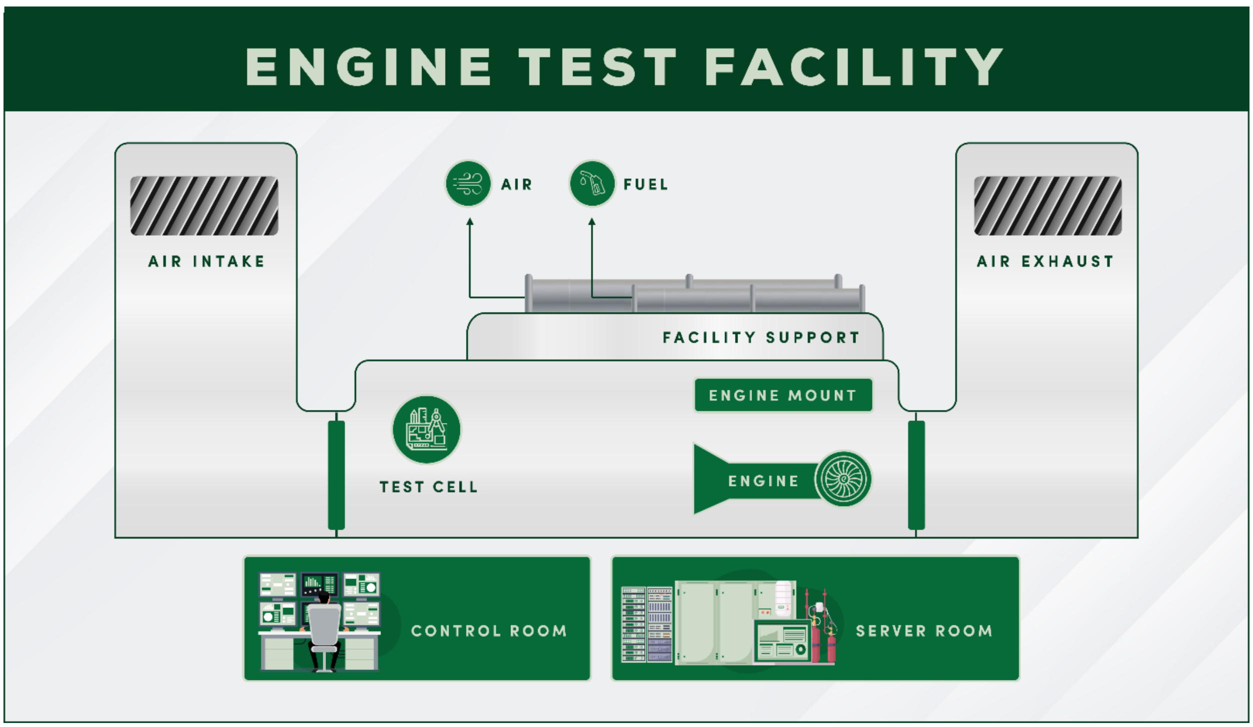


Figure 2: Indoor Engine Test Facility

An engine is mounted to a retention fixture for testing. The test cell provides air flow, intake, and exhaust. The facility provides air, fuel, and other resources to run the engine. A control room provides test operators a safe working environment during the test. Instruments are placed in various locations around the facility to collect data from the engine test.

Indoor Engine Test Measurement Zones

An important part of the test planning process is determining where to place instrumentation equipment. The engine creates a difficult environment for instrumentation—high vibrations and temperatures challenge even the most rugged test equipment. But the electrical noise of the engine degrades certain signals, so engineers must balance proximity of the test equipment to the sensors with the environmental hazards.

One way to simplify testing is to consider measurement zones within the test facility. Test engineers can match the signal needs and measurement equipment to each zone. This zone approach allows engineers to manage needs and capabilities within each zone, optimizing their test equipment within a fixed set of requirements. The following figure provides one approach to defining zones in a test facility.

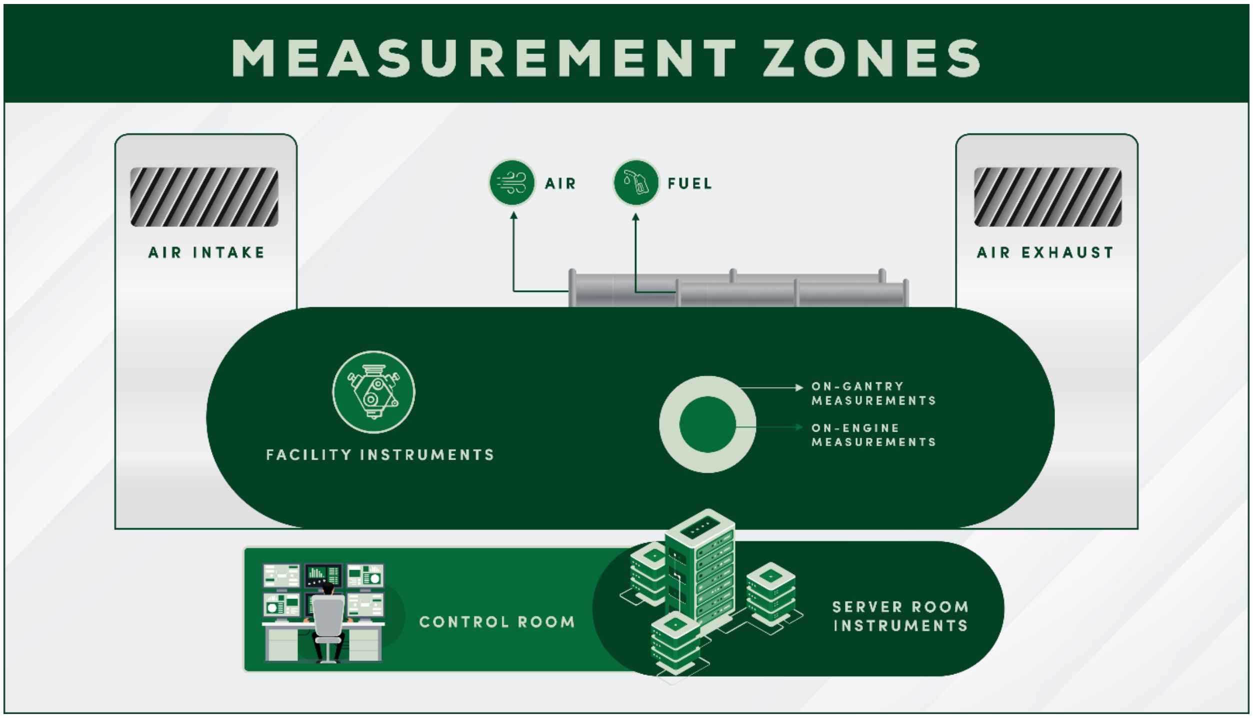


Figure 3: Indoor Engine Test Measurement Zones

On-engine measurements provide the shortest path from the instruments to the sensors, reducing electrical noise in the signal. Certain signals, like thermocouples and pressure, degrade quickly, so engineers try to place the measurement equipment as close to possible. To be mounted on the engine, test equipment must be rugged to survive the vibration, noise, and temperature of the engine.

Off-engine measurements are ideal for signals which have limited distances between the sensor and the instrumentation equipment. Vibration, acoustics, resistance measurements, and voltages fall into this category. Moving a measurement system from the engine to the engine mount, gantry, or test fixture eases the requirements for ruggedization—but does not eliminate them. The engine still generates significant vibration and noise to the surrounding area.

Within the test cell, and around the test facility, instruments are used to collect data from the engine, the test cell, and the facility. These instruments record wind speed, test cell temperatures, and signals from the engine. With instruments placed in and around the facility, environmental requirements are not as difficult. System reliability is important to avoid sending technicians into the facility for repairs during a test.

Finally, some signals may be run to a central server room for acquisition. These signals need to run long distances without losing signal quality or picking up environmental noise; low current (4-20mA) signals are well suited to this type of measurement. The advantage of using a server room for measurements is that the environment can be ideally suited to computer needs, so low-cost, high-density equipment can be used for the instrumentation.

Outdoor Engine Test Facility

Outdoor facilities provide additional test capabilities. These facilities can be constructed in extreme environments—including cold, heat, or humidity, for example—which can be expensive to replicate indoors. These facilities can also be used for destructive or potentially destructive tests, with less risk of damage to a facility.

Figure 4: Outdoor Engine Test Facility (Image courtesy of MDS Aero Support)

As with indoor facilities, a zone approach helps engineers manage test signals and measurement equipment in an outdoor facility. The zones are similar but with important differences. On-engine and off-engine signals have the added complexity of being outdoors, where environmental conditions may include rain, wind, snow, heat, and cold. Facility measurements may be located outside or in a nearby building. Server room conditions may be available in a nearby building—or may be impossible to construct. Test engineers must examine the outdoor facility options and create a zone approach that fits the situation.

Flying Engine Test Bed

Figure 5: Flying Engine Test Bed

Some flight conditions may be impossible to fully replicate in an engine test cell. Flight regulators may require testing in actual conditions on a vehicle before being deployed for commercial use. For this testing, a flying engine test bed provides the test facility.

A flying test bed is an aircraft outfitted for engine testing. In a four-engine aircraft, three engines will be standard engines, with a fourth test engine installed. The engine and the aircraft are outfitted with sensors which are run into the cabin to the measurement equipment.

Installing measurements into a flying test bed requires an additional level of expertise, and the details are beyond the scope of this document. One key to designing an appropriate test is understanding the flight conditions and the environmental conditions for the measurement equipment.

Turbine Engine Test System Solution Overview

Engine test approaches have evolved over time. What was once a monolithic test system has evolved to include a modular approach. Based on best practices identified at testing facilities across multiple industries, NI recommends this approach to turbine engine testing.

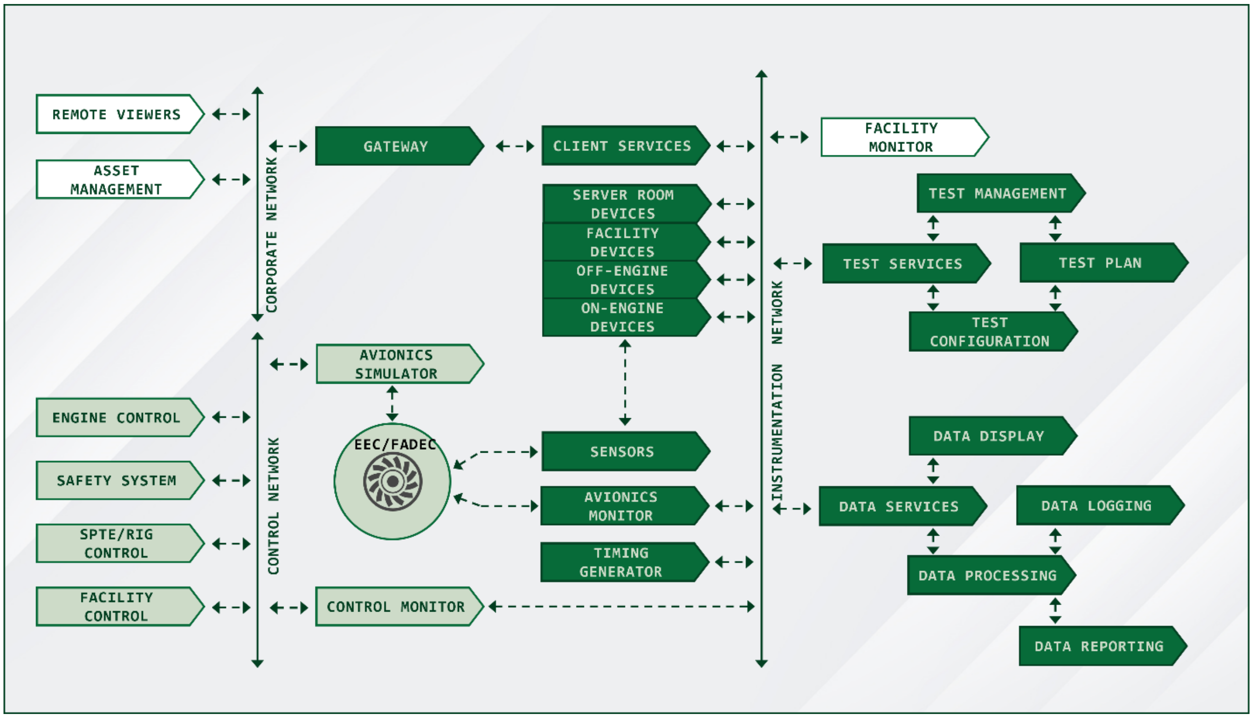


Figure 6: Turbine Engine Test Solution Architecture

The NI Turbine Engine Test Solution Architecture relies on three network systems. Separating these networks protects system components and prevents collisions within the network.

Control Network

The control network manages control signals to and from the engine. The control network connects to these subsystems:

Engine Control—The test operator executes the test from the engine control station. Depending on the nature of the test, the operator can manually run the engine from the station, or the station may automate the engine operation.

Avionics Simulator—The engine in the test fixture expects to receive certain control and monitoring signals from the aircraft. The avionics simulator provides these signals, based on information from the engine control system.

EEC/FADEC—The engine is controlled by the electronic engine controller (EEC) or the full authority digital engine control (FADEC). This controller takes information from the aircraft (through the avionics simulator) and the engine control system to generate the detailed operational commands for the engine.

Safety System—To protect the engine, the facility, and the operators, a safety system constantly monitors the test conditions and shuts the system down in the event of any anomalies. The safety system also provides a mechanism for the operator or test controller to shut down the test if necessary.

SPTE/Rig Control—Some engine tests require special-purpose test equipment (SPTE) or special test rigs; equipment may provide water injection to simulate rain conditions, for example. This system controls the specialized equipment.

Facility Control—The facility is an important part of the engine test; the facility controller provides air flow, fuel flow, engine cooling, engine mounts, and other equipment necessary for the execution of the test. This control must be closely coordinated with the rest of the engine control system.

Control Monitor—Throughout the test, control signals must be logged. The instrumentation system often relies on these signals as well, so the control systems connect to the instrument network through a control monitor system. This system publishes data to the instrumentation network in a format that the instrumentation systems can subscribe to the control data.

Instrumentation Network

The instrumentation network connects all the instrumentation systems together, ensuring that the data is captured, logged, and reported to the test engineers.

Sensors—The engine is outfitted with sensors, as requested by the design and test engineers as part of the test. For a successful test, engineers must understand each of the sensors to be used in the sensor system—including the signal type, excitation, frequency, range, and engineering units of each sensor.

Avionics Monitor—Many measurement signals are provided by the sensors attached to the engine. Other signals are acquired within the engine as well; these are made available on the avionics bus of the engine. The avionics monitor captures these signals from the engine and makes them available to the instrumentation system.

Devices—Devices capture measurement signals from the sensors and publish them to the instrumentation network. Devices must be matched to the sensor output signals to provide proper excitation and conditioning, they must acquire the data at appropriate speeds and accuracy, and they must meet the environmental requirements of their measurement zone.

Timing Generator/Master Clock—As engine technologies grow increasingly complex, the acquisition timing becomes more critical to success of the measurement system. The timing generator serves as master clock to provide a timing signal that is used by the instrumentation system to ensure accurate timing. Engineers must decide if relative timing is sufficient, using technologies like IEEE 1588 Precision Time Protocol (PTP), or if absolute timing is necessary using a technology similar IRIG-B in GPS or GNSS. Accuracy, cabling, and distance must all be considered in selecting a timing generator.

Test System—The test system is composed of several components described as follows. Engineers use these components to plan, configure, and execute the test.

Test Plan—Test engineers take test requirements from design engineers to plan the test. The test plan includes the signals targeted for the test, the sensors to be used, the device capabilities required for the test, the layout of the instruments, the facility and special-purpose test equipment to be used, and the steps the test will follow. The test plan defines the tests that will be run, the specific order of the tests, and the data to be recorded in each test. The test plan may include specific dates and facility information for the test.

Planning tools can range from simple documents to complex requirements tracking systems and may be manual or automated. Good planning tools help engineers quickly and completely translate requirements into a complete plan.

Test Configuration—The test configuration component takes the test plan and translates this into specific configuration instructions for the instrumentation and engine control components. The configuration may include acquisition rates, ranges, excitation levels, and control signals. The test configuration tools may be manual or automated.

Test Execution—The test execution tools take the test plan and step through the actual execution of the test. The execution tool may include a setup phase, execution phase, and cleanup phase, or the test plan may be more detailed. The execution tool signals the system to put the engine into a specific run state, then signals the measurement devices to acquire the signals; this may manage the auto-throttle control of the engine during the test.

Test Services—The test system needs to interface with the rest of the instrumentation network and the engine test architecture. The test service component translates the output of the test execution tools to communicate in a way the rest of the instrumentation system can understand. This communication may rely on industry standards like iDDS or OPC UA, or it may be defined by the engineers within a group or company.

Data System—The data system components collect, display, log, and report the data generated during the test.

Data Services—Data available on the instrumentation network is collected by the data services component. This component collects the data from locations on the network and converts the data-to-data formats used by the other data components. The data services may be built into other components, depending on the complexity of the data types and the sophistication of the data tools.

Data Display—Test controllers, operators, and monitors need to see the data to make test decisions. The data display tools make the data available to for real-time viewing and analysis.

Data Processing—Data processing tools apply mathematical algorithms to the data. This may apply engineering units to the data, process data through filters or other algorithms, or combine data points to generate new information (such as combining fuel flow and thrust to determine efficiency). Data may be processed by other data tools or may be offloaded to a dedicated processor to accelerate analysis.

Data Logging—Data logging tools move data to storage locations, including local hard drives or network drives. They may store raw data, processed data, or both. In addition to sensor data, data logging may include control signals and operator actions. Data logs may be stored in binary formats to streamline data movement and storage, or they may be stored in more readable formats, including text.

Data Reporting—Reporting tools collect insights from the data logs and make the data available in human-readable reports. Report generation may be manual or automated; these reports are used by the engineers or customers who request the test data, and different versions may be made available to different roles. For short-term tests, reports are usually generated after the completion of the test. For longer-running tests, reports may be generated periodically to update stakeholders on the progress of the tests.

Client Services—Client services take data from the instrumentation network and make it available to outside services. Engineers determine which data is made available through the service as well as how the data is made available. Data may be streamed out continuously, or it may be made available upon request through a defined API. It is important that the client services expose data correctly—without putting the instrumentation network at risk to accidental overuse or cyberattacks.

Corporate Network

Gateway—The purpose of a gateway is to protect the instrumentation and control networks from outside attacks and network issues. Controlling which information gets through the gateway ensures that corporate users have access to the data they need, without the ability to pass network requests which could interfere with the performance of the other networks. A gateway could be a properly configured firewall, or something more extreme like a data diode.

Remote Viewers—Data made available by the client services module is accessed by engineers, managers, customers, and other interested stakeholders through viewers. These may be thin-client web pages requiring no local installation on the viewing computer and may allow the viewer to configure what they see, or they may be standardized dashboards available with specialized viewing software.

Asset Management—Engineers may spend months planning tests, including which test equipment will be available for each test. An asset management tool helps engineers manage the equipment they have available, reserve that equipment for testing, ensure the equipment is properly configured, and is available at the time of the test. It is common for asset management tools to track calibration status of test equipment so that engineers trust that tests will not be delayed in order to calibrate the equipment they rely on for the tests.

Facility Monitoring

Facility Monitoring—An engine test facility is a complex set of equipment, networks, personnel, and processes. When any part of the facility fails, it can cause expensive delays in the engine test while a team finds, identifies, diagnoses, and repairs the issue. Monitoring the health and readiness of all parts of a test facility requires access to thousands of data points across the facility.

Good facility monitoring software collects data from many different systems, including data from the test fixture, equipment rigs, the engine, and the sensors on the engine. It collects information from the control system and the safety systems. It collects data from the facility—the rooms, the doors, and the environmental controls, as well as data from the network and the attached systems about network bandwidth, processor time, and storage systems.

Together, all this data can provide a picture of the facility's health. It can identify existing issues in the facility or predict issues that are expected to happen soon. It can report facility efficiency and show performance trends over time. The facility monitoring tool can collect this data, automate the analysis, and present data to facility managers through remote dashboards.

Communication System and Data Abstraction

The recommended turbine engine test solution architecture allows for more modular components in the engine test system. Test engineers can manage each component separately, selecting the right technology for each part of the system. The modular approach creates added value in expanded capabilities. It also introduces risk that some components will not be able to connect to other components.

Communication System

When designing a turbine engine test architecture, engineers must determine a strategy for connecting the components of the system together. When considering a technology for communication among the modules, engineers must consider several important factors:

Physical Connections—Does the technology run over existing network lines, or does it require a proprietary connection system?

Bandwidth—Does the communication support the data rates—per channel, and in aggregate, of the connected modules?

Timing—Does the communication system support the timing requirements of the engine test? Do the modules require an absolute timing reference, or a relative one?

Metadata—Does the communication system convey information about the data, along with the data?

Modules Support—Is the communication compatible with all the components in the system?

Future Capability—Will the communication system support future expansion as the test system evolves?

A communication system may be proprietary to a particular vendor. Systems built on proprietary communications systems may lock engineers into a single solution, raising the risk of increased cost, obsolescence, and limited upgrade options. However, a proprietary system may enable a vendor to create and evolve technologies to provide unique value to their customers. Test engineers who design a system based on a proprietary vendor technology should balance the benefits of the single solution with the risks involved.

A company or organization may design a custom communication system for the test architecture. This has the advantage of being suited to the needs of that organization and the ability to be adjusted over time to evolve with the needs of the organization. It can be adapted to work with technologies from multiple vendors. But designing a robust, responsive, and capable communication system is no easy task; doing so requires an experienced team of developers and sufficient development resources.

Data Abstraction

A third option is to implement a communication system based on industry-standard technologies. The right industry standards provide the capabilities needed to support the test architecture, are open to the community to evolve with changing technologies and are fully tested for robustness and reliability.

One approach of these industry standards is data abstraction—isolating various modules from the communication system. A good data abstraction approach makes it possible to mix modules from multiple vendors. This protects the system for future enhancements by ensuring that new technologies can be inserted in place of aging ones. The data abstraction system must consider all the considerations listed above.
There are several data abstraction technologies on the market today, with varying degrees of adoption and use. This paper will not evaluate all these; instead, it will introduce a new standard emerging in engine test applications—iDDS.

iDDS—Instrumentation Data Distribution Service

Data Distribution Service (DDS) is networking middleware that handles the core tasks of a communication system, including message distribution, reliability, and interoperability. Instrumentation Data Distribution Service (iDDS) adds message formatting specific to large scale instrumentation systems, including device configuration, acquisition rates, data formatting, and error codes.

iDDS is managed by the iDDS working group and is progressing toward adoption as an open standard. As iDDS becomes a more mainstream standard, it brings these benefits to turbine engine test facilities:

- Interoperability of components from multiple vendors
- Support for low- and high-speed signals typical in engine test applications
- Communication reliability
- Optimization of network communication

For more information about iDDS, visit https://mdsaero.com/test-solutions/data-acquisition-systems/

Turbine Engine Test Solution Summary

This section provided an overview of a turbine engine test architecture based on a modular approach. It described three separate yet connected networks: the Control Network, Instrumentation Network, and Corporate Network. Each of these networks is connected to various modules and subsystems, which must be connected using a communication system. The following section describes some of the technologies that are used in these engine test modules.

Turbine Engine Test Technologies

Software and hardware technologies continue to evolve to provide more options to engineers designing engine test systems. This section provides an overview of some of the modern technologies available to engineers today.

Turbine Engine Software

In this solution architecture, software is used in multiple subsystems. Software is used as application tools as well as embedded interfaces in the hardware. In designing a turbine engine test architecture, engineers must choose from tools made specifically for engine testing, tools made for general purpose applications, and tools made by the engineering team.

NI Software for Turbine Engine Test

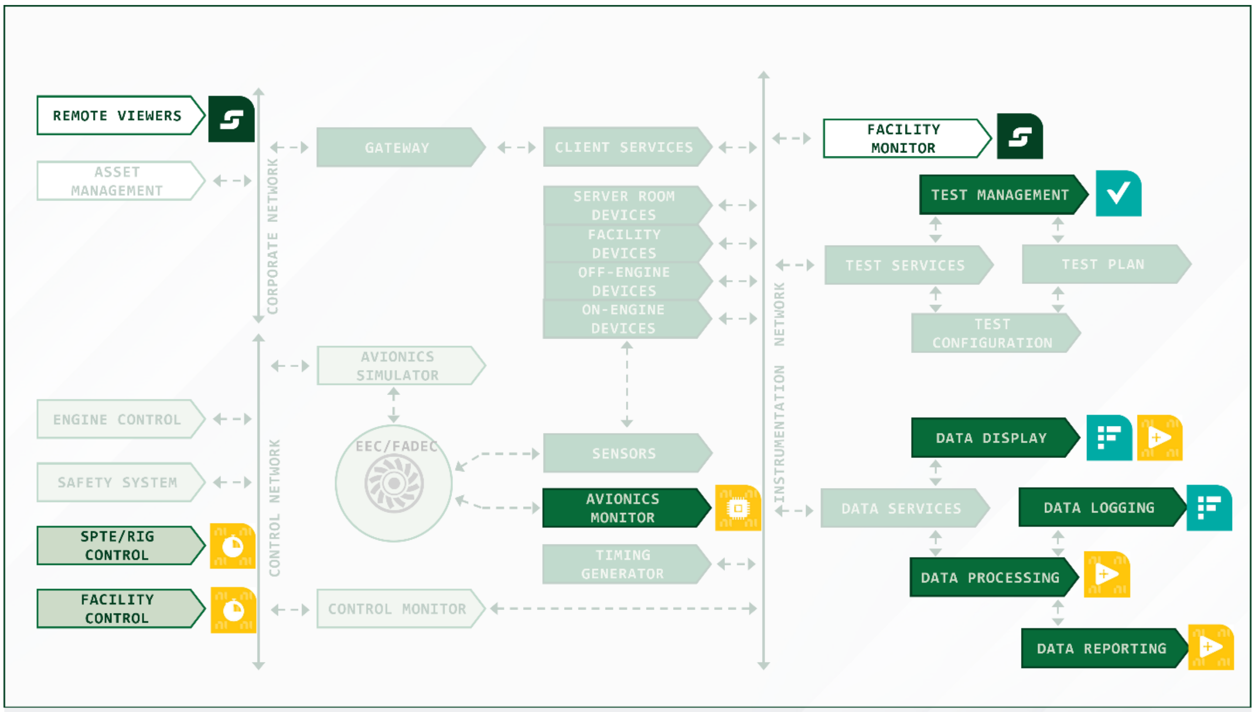


Figure 7: NI SW Tools for Turbine Engine Test Solution Architecture

There are multiple NI development tools and application tools available to support the Turbine Engine Test architecture.

Graphical Programming: LabVIEW Development Tools

LabVIEW is a graphical development platform engineers can use to create customized test solutions, providing thousands of analysis functions, configurable display elements, drivers to interface to instruments and data acquisition equipment, and connectivity to other languages and protocols.

The LabVIEW Professional edition provides tools for engine testing, including the capability to build executable files for distribution to various test stations around the facility. It also includes important development tools including graphical code diff and merge, dynamic code analysis, static code analysis, and unit testing. With this edition, developers can create custom tools for the modular components. LabVIEW's focus on data and instrumentation makes LabVIEW an ideal tool for developing the data display, data logging, and data processing tools. With LabVIEW's platform-independent code compiler, these can be targeted to Windows, Linux, or Mac OS stations.

For custom applications that require real-time reliability and performance, the LabVIEW Real-Time add-on toolkit enables LabVIEW programmers to take graphical code to embedded targets running Linux Real-Time OS. These embedded applications are appropriate for Facility Control and SPTE/Rig Control applications.

Applications that require high-speed data processing are ideally suited to FPGA targets. The LabVIEW FPGA add-on toolkit supports graphical programming on these targets, with a robust ecosystem to support IP development of Avionics Monitor tools. LabVIEW FPGA can be used to create interfaces to most avionics buses and share that information to the instrumentation network.

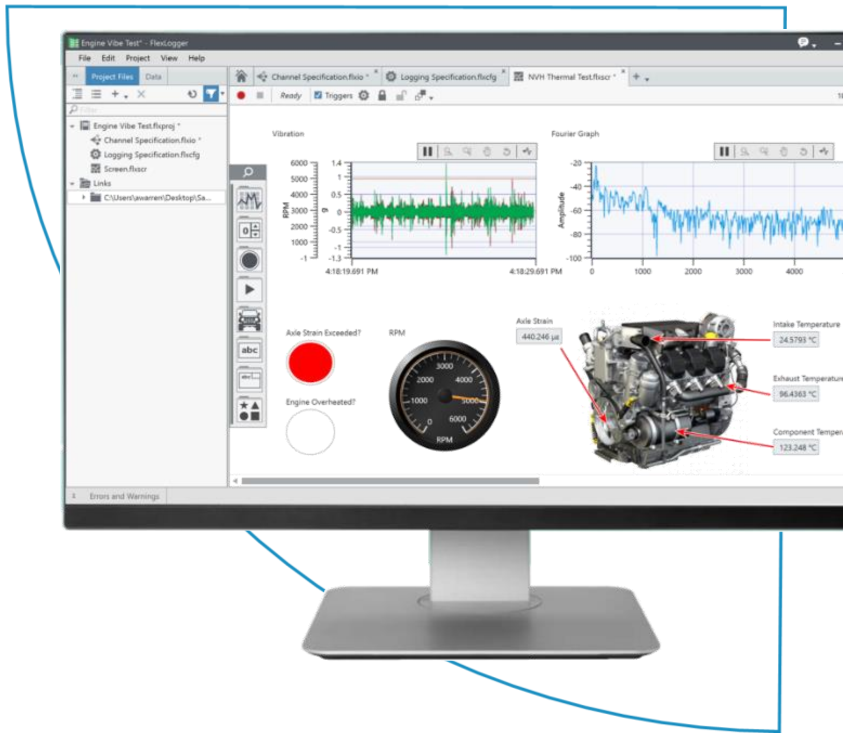
G Web Development Software helps engineers create web-based user interfaces for test and measurement applications without the need for traditional web development skills. Web applications developed with G Web Development Software can connect to existing systems written in LabVIEW, Python, or C#.

More information about LabVIEW Professional, LabVIEW Real-Time, and LabVIEW FPGA is available at ni.com/labview.

Configuration-Based Data Visualization and Logging: FlexLogger

For engineers who want to minimize programming with out-of-the-box software, FlexLogger provides a powerful data display and data logging tool. Engineers configure FlexLogger to collect data from data devices attached to sensors, then have access to analysis, processing, visualization, and logging tools.

Operators can use sensor-specific configuration workflows to quickly set up, visualize, and log a mix of synchronized measurements from analog sensors, digital signals, and communication buses. FlexLogger can generate voltage, current, or digital signals to drive actuators or control set points. FlexLogger automatically saves metadata documenting test configuration, so engineers can quickly trace test results and make comparisons across multiple tests. Engineers can interactively review test results in the integrated data viewer to visually inspect data and draw conclusions.



FlexLogger supports a variety of hardware platforms, including PXI, CompactDAQ, and FieldDAQ. FlexLogger provides valuable tools in the Turbine Engine Test Architecture for data display, data processing, and data logging.

More information about FlexLogger is available at ni.com/flexlogger.

Test Sequencing and Test Management Executive Software: TestStand

TestStand is ready-to-run test management software designed to help engineers quickly develop and execute test routines from test plans. Engineers build TestStand test sequences that integrate code modules written in a variety of programming languages, including LabVIEW, C/C++, .NET, and Python. TestStand also provides extensible plugins for reporting, database logging, and connectivity to other enterprise systems. Engineers can deploy test systems to production with easy-to-use operator interfaces.

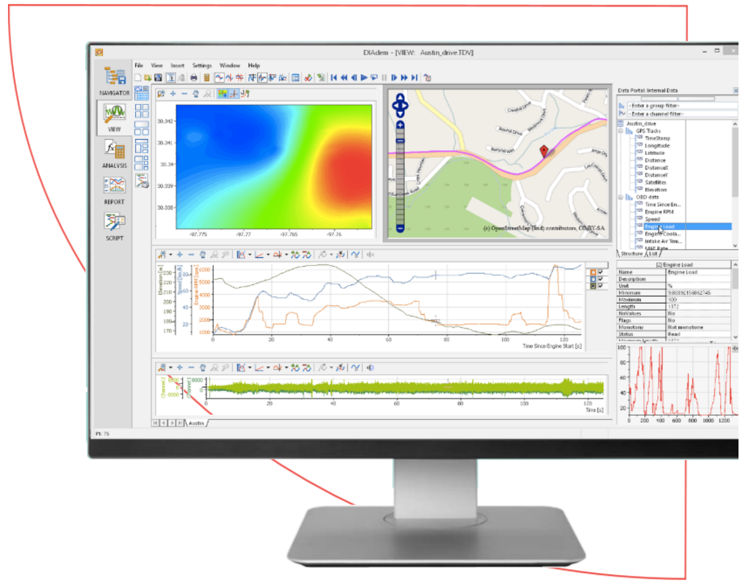
TestStand automates, accelerates, and standardizes the overall test process with native functionality for calling and executing tests written in LabVIEW, Python, C/C++, or .NET. TestStand handles complex tasks, such as parallel testing, sweeping, looping, and synchronization. Engineers can use TestStand to create custom operator interfaces and robust tools for deployment and debugging. TestStand supports unit tracking, automated reports, and storing results to local or network databases.

TestStand is ideally suited in the Turbine Engine Test Architecture as the Test Execution tool.

For more information about TestStand, visit ni.com/teststand.

Data Analysis and Reporting—DIAdem

DIAdem is application software that helps engineers accelerate post-processing of measurement data. It is optimized for the large data sets common in engine test and includes tools to quickly aggregate and search for the needed data. Engineers can view and investigate data with engineering-specific analysis functions. Reports can be built manually or automated through DIAdem's powerful scripting interface.



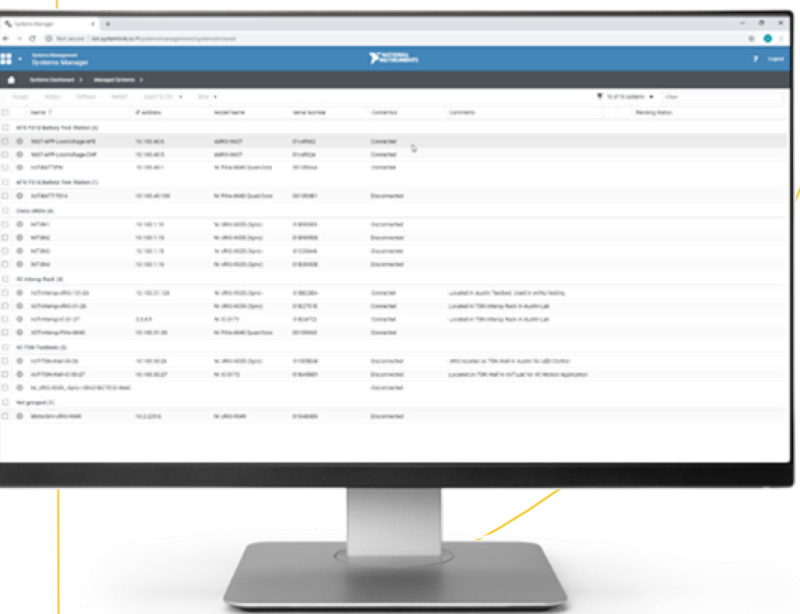
Using DIAdem, engineers can build integrated report dashboards that include data visualizations, graphs, graphics, and even video files from the test. Test engineers, design engineers, and data analysts can use these dashboards to gain valuable insights into the engine, the facility, and the test process.

DIAdem is a powerful tool to provide reporting for the Turbine Engine Test solution.

For more information on DIAdem, visit ni.com/diadem.

Data and Systems Management—SystemLink Software

SystemLink is an intelligent Systems and Data Management environment that connects the test facility to the rest of the organization. Designed for engineering use cases, SystemLink software combines focused applications and data services that accelerate time-to-knowledge and time-to-market by leveraging comprehensive real-time information. From engineering teams to enterprises, SystemLink software helps organizations achieve peak performance.



SystemLink centrally manages test equipment and assets across the engine test facility, including software, systems, and fixtures. SystemLink collects calibration status, system and application software versions, and equipment health. Using this information, test engineers can plan to minimize downtime for maintenance. Test engineers can also deploy software updates for system software, drivers, and applications.

SystemLink also centrally collects valuable data about the test facility and the testing in the facility. From this data, test engineers can monitor test trends, facility efficiency, and engine data. Using this data, engineers can predict equipment failures and schedule repairs to reduce cost and minimize downtime. Managers can use this data to adjust engine test schedules and most efficiently use these valuable resources.

SystemLink provides an important set of tools for Facility Monitoring and Asset Monitoring in the Turbine Engine Test Architecture.

For more information on SystemLink, visit ni.com/systemlink.

MDS Gas Turbine Engine Solutions and nxDAS

If you need additional engine test resources to support your team, you need a partner like MDS. Their specialists can help you design, service, and manage your test facility and the engine test instrumentation.
MDS is an expert integrator with more than 30 years of proven experience providing tailored test solutions for a full spectrum of test article types (aero engines, compressors, industrial gas turbines, component testing, etc.), so organizations will not have to worry about selecting various components or getting them to work together.
Each made-to-measure solution is based on the customer's unique needs and capabilities. From test-article-driven requirements such as measurement types, quantities, accuracies, and acquisition rates, to constraints based on existing installations or instrumentation inventories, MDS can help design, service, and manage your test facility and instrumentation.

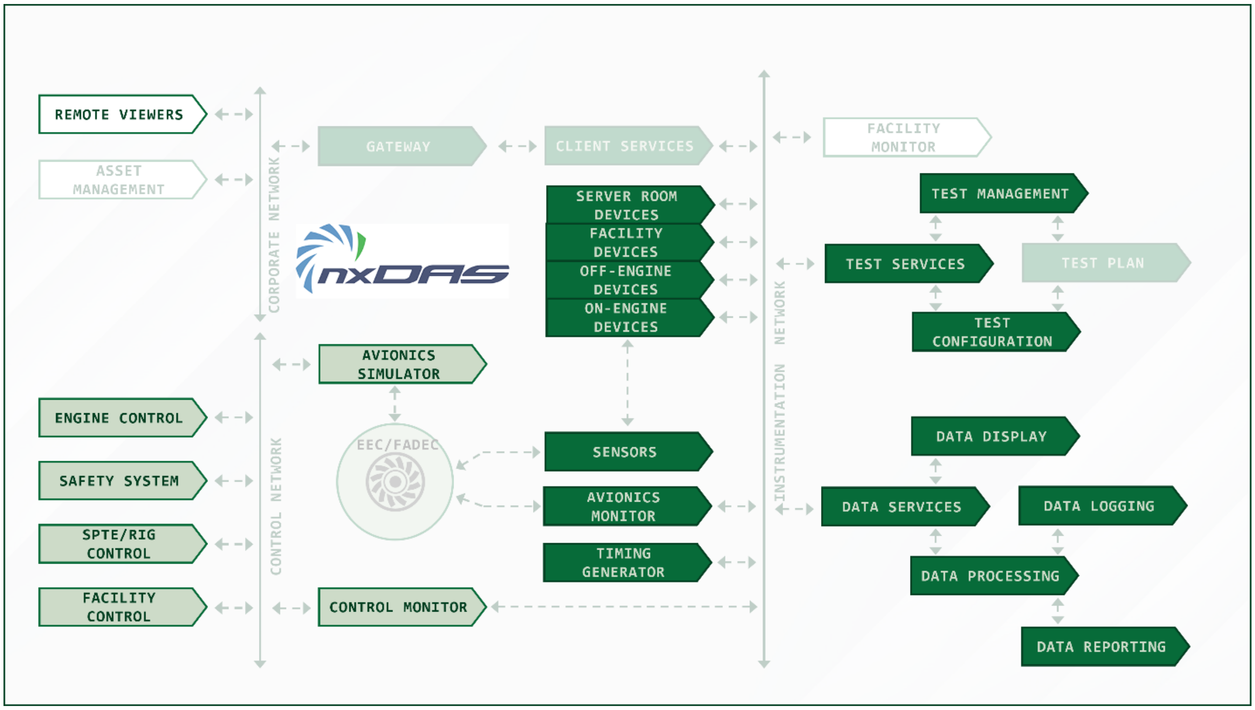


Figure 8: MDS nxDAS Tools for Turbine Engine Test Solution Architecture

Product Details

nxDAS (network extensible Data Acquisition System) is a test software suite provided by MDS and built on decades of experience. nxDAS uses iDDS, an instrumentation-centric extension of OMG's DDS (data distribution service) in a publish/subscribe paradigm. This offers many advantages such as driverless connectivity to iDDS-compliant instruments (new instruments self-identified on the iDDS Network) and data time stamping at source (IEEE-1588).
MDS plays a leadership role in the development and promotion of the iDDS measurement industry standard by hosting the iDDS Working Group collaborative site and by chairing regular Working Group meetings. Although iDDS is the way forward, nxDAS also includes support for a broad array of legacy, non-iDDS compatible instruments and test network components.

nxDAS Platform

The nxDAS system incorporates various hardware and software test components into a single, intuitive, and highly efficient tool. Elements such as steady-state (low speed) and dynamic (high speed) measurement front ends, avionics simulation and monitoring, engine, facility, SPTE/Rig, and safety control systems are seamlessly integrated into a single, powerful tool that is simple to learn and easy to use.

nxDAS is fully scalable from tens to thousands of channels (signals) and is just as powerful whether used for smaller, more constant, MRO (Maintenance, Repair, and Overhaul) type configurations, or larger and more fluid R&D (Research and Development) type configurations. The user interface has been designed to follow an intuitive "windows" style standard familiar to most users, resulting in minimal effort required to gain proficiency for unpracticed users. The extensive library of calculated functions as well as the ability for simultaneous multiple channel configuration make it simple to build test configurations from scratch or to clone and edit existing configurations. Even though the system can perform a multitude of complicated functions, no programming knowledge is required to configure it to do so. Either MDS or trained customers can easily build and manage test configurations.

Test Automation

Means of automation are provided using Python. Since nxDAS is fully integrated into the Control Network, it is possible to introduce any desired level of automation into the testing sequence. From entirely manual to fully automated (including data recordings and throttle movements) and anywhere in between, the automation capabilities can be used to refine the test runs for more consistent and repeatable results.

Health Monitoring and Alarm Processing

The nxDAS system includes an instrumentation management interface which allows for health monitoring and troubleshooting of connected iDDS instruments.

A comprehensive Limits and Actions module is used to warn system operators of potential trouble or automatically react to return the test article and/or test facility to a safe condition within 5 ms of a limit exceedance event. The duration and magnitude of any limit exceedance events is automatically recorded by the system. Limits can be configured either as fixed or variable (based on a measurement or calculation) values. Actions ranging from event messages, starting/stopping recordings, setting and clearing of channel values (e.g., shut-off engine fuel valve bit), or launching a Python script can be configured as a response to a limit state change. Configured alarm limits can be easily displayed to the system operators via the graphical data displays. A concise alarm summary window listing any channels in alarm state along with other pertinent information (state, limit, current value, max deviation, etc.) is also available.

Data Display

A straightforward drag-and-drop display editor with a full suite of graphical elements is provided to build simple or sophisticated real-time data displays. There is no limit to the number of displays which can be created. Displays can be dedicated to a specific monitor, be made available to all monitors, or be sequenced for viewing in a specific order on a specific monitor. Data displays can be tiled on the same display monitor or maximized for full-screen viewing. Parameters may be displayed using any combination of dial gauges, horizontal and vertical bar gauges, digital displays, strip-charts, y-x graphs, digital indicators, or profile plots. The appearance of the custom controls is also highly configurable.

Data Recording and Test Reports

nxDAS offers reliable recording of steady-state and transient data including data time alignment and integration of high-speed dynamics data recordings. All recorded data can be pushed to central networked storage.

The unified measurement system event logging and recording module tracks and records all system events, alarms, faults, user actions, etc., and makes these available for use in data post-processing. A rotating buffer, called the critical log, records all parameters defined as critical. The duration of data saved in the circular buffer can be configured so that when the critical log is saved, data from both before and after the save event is recorded; a feature not often required yet very

convenient in post-catastrophic event analysis. The continuous recording and test playback utility automatically records all acquired data. When accessed in playback mode, the continuously recorded data can be used to reproduce the entire test sequence; calculations can be updated and missed data recorded without the expenditure of another test run.
The reporting and viewing modules provide a means to quickly review recorded test data as well as to generate test reports and templates. Test reports and templates are Microsoft Excel-based, providing considerable flexibility in the possible level of report complexity—based on the data features available in Excel. Once generated, the existing templates can be populated with new data with a few clicks of the mouse.

Documentation and Customer Support

nxDAS includes a detailed, fully indexed digital product support manual provided as a reference for training and self-service of basic support questions.

Turbine Engine Hardware Components and I/O

Hardware provides critical interfaces to turbine engine through the sensors and the facility and SPTE/Rig interfaces. Hardware must be carefully selected to match the acquisition requirements, processing speed, signal accuracy, and environmental specifications demanded by the engine test application.

NI Hardware for Turbine Engine Test

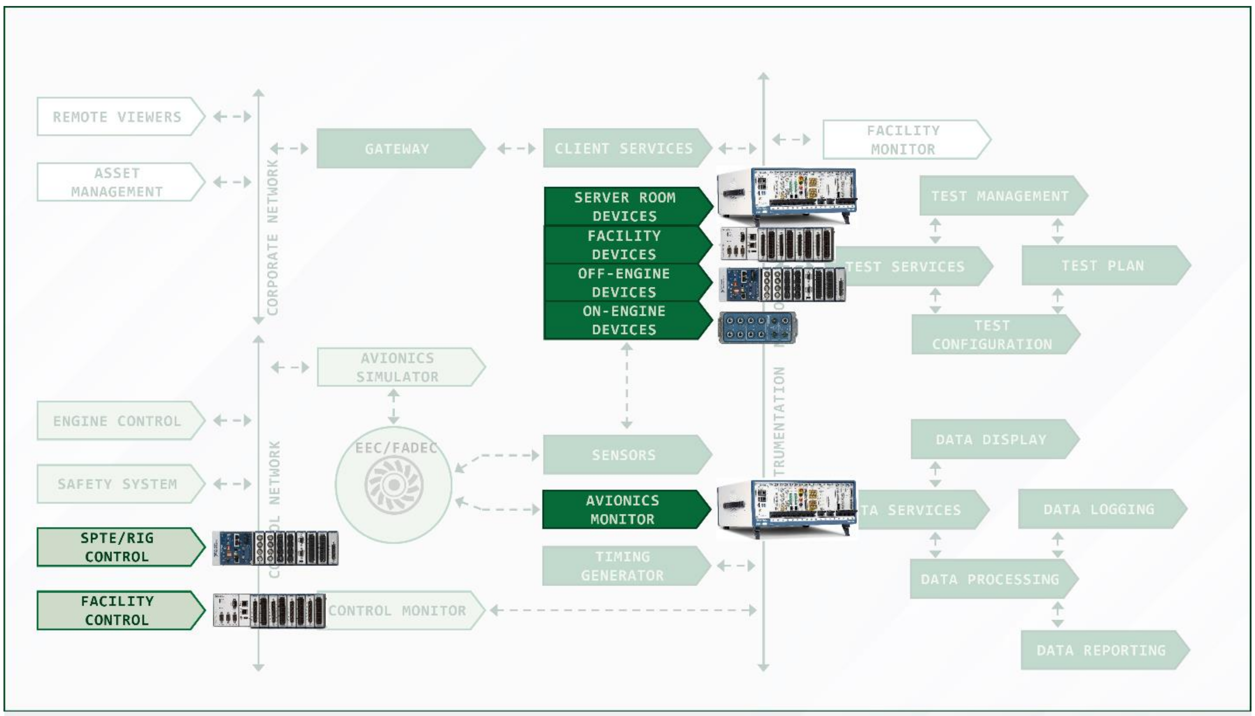


Figure 9: NI Hardware Tools for Turbine Engine Test Solution Architecture

As described in the solution overview, it is convenient to identify measurement zones. These zones can then be used to create specific environmental requirements for the hardware. Specific requirements can simplify the selection and management of hardware within each zone.

On-Engine	On-Gantry	Facility	Server Room
Short-cable measurements Temperature, pressure	**Larger number of channels** Temperature, voltage, flow, strain, vibration, acoustic	**Permanent channels** Safety interlocks, environmental parameters, facility control signals, PLCs	**Bus signals** Current measurements, avionics buses
High vibration Electrical noise	Medium vibration Electrical noise	Low vibration	No vibration
Rugged connectors Rugged Ethernet Power over Ethernet No moving parts Extended temperature	No moving parts Extended temperature	Rugged hard drive Real-time performance	Computer-grade instruments
FieldDAQ	CompactDAQ	CompactRIO	PXI

Figure 10: NI Hardware by Measurement Zone

Rugged I/O: FieldDAQ

FieldDAQ delivers NI measurement capabilities in a rugged form factor appropriate for on-engine measurements. FieldDAQ devices have an ingress protection rating up to IP67, which means they are dust-resistant and water-resistant. They maintain excellent accuracy across a wide range of extreme environments, including temperatures from -40 °C to 85 °C, 10 g vibration, and 100 g shock.

Test engineers can distribute FieldDAQ systems across Ethernet networks to get closer to sensors and signals. FieldDAQ also distributes timing signals with Time Sensitive Networking (TSN) to provide accurate synchronization within 1 µs over long distances using standard Ethernet cables. Sharing data transfer and synchronization over a single cable cuts cabling costs and complexity.

FieldDAQ devices are available to support a variety of sensors, including voltage, strain, IEPE/vibration, and thermocouples.

For more information about FieldDAQ devices, visit ni.com/fielddaq.

Distributed Measurements: CompactDAQ

For measurements distributed around the facility, especially next to the engine on the gantry and other fixtures, CompactDAQ provides a rugged, modular platform. Engineers select CompactDAQ chassis based on communication interfaces (USB, Ethernet, or wireless) and the number of module slots (4 or 8). CompactDAQ delivers rugged performance at a cost reasonable for large number of channels. CompactDAQ can be used in environments with temperatures ranging from -40 °C to 70 °C, 5 g vibration, and 50 g shock.

Test engineers select from more than 70 C-Series data I/O modules to create a customized measurement node. Modules available from NI and other providers support signals including voltage, current, temperature, vibration, digital, relays, frequencies, flows, avionics and automotive buses, and GPS.

CompactDAQ chassis support Time Sensitive Networking (TSN) to synchronize measurements across the entire engine test network within 1 µs.

For more information about CompactDAQ chassis and C-Series modules, visit ni.com/compactdaq

Distributed Processing and Control: CompactRIO

CompactRIO adds distributed control and processing to engine test applications. CompactRIO systems provide a local processor running embedded NI Linux RT OS and a user-programmable Field-Programmable Gate Array (FPGA). CompactRIO systems are rugged enough to survive engine test environments, with temperature ranging from 0 °C to 55 °C, 5 g vibration, and 50 g shock.

Test engineers select from the same C-Series modules for CompactRIO as they do for CompactDAQ, reducing maintenance costs across applications. Modules available from NI and other providers support signals including voltage, current, temperature, vibration, digital, relays, frequencies, flows, avionics and automotive buses, and GPS.

Engineers can program the CompactRIO device using LabVIEW Real-Time software, or with other industry standard Linux tools. Using these tools, engineers can add signal processing or local control and distribute this around the facility, including next to the engine.

Engineers can also program the on-board FPGA using LabVIEW FPGA software or other FPGA toolchains. Engineers can add complex algorithms to the CompactRIO node with FPGA, operating at hardware speeds and reliability.

CompactRIO is NI's platform for iDDS. NI engineers have developed embedded software to run on the CompactRIO processor, collecting data using C-Series devices and publishing that data to the iDDS network.

For more information on CompactRIO systems, visit ni.com/compactrio.

Industrial Computing: NI PXI/PXI Express

PXI is a PC-based industrialized system that combines PCI Express electrical-bus features, a modular chassis, and I/O synchronization technology with user-defined or application-specific test software. PXI is an open industry standard governed by the PXI Systems Alliance, a group of more than 70 global test companies. NI was one of the pioneer companies in the formation of PXI and is recognized as a leader in PXI test and measurement devices.

Instrumentation available in PXI/PXI Express form factor includes:

- Analog and digital I/O
- Digital multimeter
- Oscilloscope/digitizer
- Waveform generator
- Switch and timing/synchronization
- Source measure unit (SMU)
- Programmable DC power supply
- Electronic load
- Instrument control and synchronization
- FPGA processing boards

Engineers select from a variety of available chassis and controllers and add instrumentation and measurement modules to create measurement systems. These systems can be customized to match the specific needs of the test facility.

PXI controllers run standard operating systems, including Windows and Linux. For embedded control applications and real-time performance, NI Linux Real-Time OS is available on PXI controllers. Engineers can program these systems using LabVIEW Real-Time software.

Programmable FPGA boards available for PXI can be programmed using LabVIEW FPGA software and provide powerful signal processing capabilities to engine test engineers. These boards can also be used to interface to avionics buses and run models of avionics components to complete the test environment for the engine under test.

For more information on PXI systems, visit ni.com/pxi.

Services and Support

NI provides several services and support options to help ensure your short-term and longer-term success with our products. We've partnered with many aerospace and defense companies to provide extended, long lifecycle support that can span decades. Select from NI's services and support options to design a system management approach that meets your budget, risk, and resource needs.

Training
Choose from online, classroom, and on-site training options for NI products

Support
Access online technical support, phone support, or dedicated support resources

Repair
Select standard 3-year or optional 5+-year repair options

Calibration
Send products to NI or certified centers for repair, or arrange on-site calibration services

Sparing
Minimize downtime with locally spared repair pools set up to meet your needs

Long Life Support
Get product update notifications and arrange long-life support (20 years or more) for your system

NI offers a full suite of service and support offerings to meet your individual project needs. For more information about any of NI's services, please visit ni.com/services.

NI's Partner Ecosystem for Turbine Engine Test

We work closely with our uniquely qualified partners so that our customers see the greatest return on investment from their turbine engine test systems. MDS is a high-value partner who provides full-scale engine test solutions, including facilities, management, and instrumentation.

MDS Gas Turbine Engine Solutions

The world's biggest engine manufacturers trust MDS. Founded in 1985, MDS is the preeminent supplier of turnkey test facilities, test systems, and service solutions for the aviation, industrial, and marine gas turbine engine community. With a steadfast commitment to the provision of superior products and services to its clients around the world, MDS designs and builds highly complex multi-million-dollar gas turbine engine test facilities worldwide. MDS is also proud to have been selected as one of Canada's Best Managed Companies since 2012.

MDS provides unparalleled quality services to both OEMs and MROs around the globe. Certified to ISO 9001, 14001, and 17025 standards, MDS has supplied engine test facilities, test support equipment, data acquisition systems, engineering services, and facility operation and maintenance services to a variety of industries and organizations globally for nearly four decades.

Working towards a more sustainable future, the company is at the forefront of research and development, continually driving innovation. A single source for all engineering and project requirements, MDS has scaled their core offerings to help all stakeholders see opportunities to optimize energy demands and, where possible, incorporate energy-efficient or energy-saving systems.

For Additional Information

More Turbine Engine Test Resources

- NI Turbine Engine Test Solutions

Customer Success Stories

- Lockheed Martin Reduces Costs and Time Testing F-35 Joint Strike Fighter with LabVIEW Real-Time
- Upgrading a Test Bench for a Fighter Aircraft Turbojet Engine with NI CompactDAQ and LabVIEW

Products

- LabVIEW Development Environment: https://ni.com/labview
- LabVIEW Real-Time: https://www.ni.com/en-us/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-labview-real-time-module.html
- LabVIEW FPGA: https://www.ni.com/en-us/shop/electronic-test-instrumentation/add-ons-for-electronic-test-and-instrumentation/what-is-labview-fpga-module.html
- Web Development Module: https://www.ni.com/en-us/shop/electronic-test-instrumentation/programming-environments-for-electronic-test-and-instrumentation/what-is-g-web-development-software.html
- FlexLogger: https://ni.com/flexlogger
- TestStand: https://ni.com/teststand
- DIAdem: https://ni.com/diadem
- SystemLink: https://ni.com/systemlink
- FieldDAQ: https://ni.com/fielddaq
- CompactDAQ: https://ni.com/compactdaq
- CompactRIO: https://ni.com/compactrio
- C-Series Modules: https://www.ni.com/en-us/shop/compactdaq/compactdaq-modules.html
- PXI: https://www.ni.com/en-us/shop/pxi.html