



Final Project Report

Process Manufacturing Digital Twin – Open Architecture Testbed Framework	
Principal Investigator / Email Address	David Warner / dwarner@adi.com
Project Team Lead	Applied Dynamics International Inc.
Project Designation	19-04-03
MxD Contract Number	2020-04-B
Project Participants	Applied Dynamics International Inc. University of Michigan The Dow Chemical Company
MxD Funding Value	\$157,167.89
Project Team Cost Share	\$286,603.07
Award Date	15-Jun-2020
Completion Date	2-Apr-2021

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

This project was completed under the Technology Investment Agreement W15QKN-19-3-0003, between Army Contracting Command – New Jersey and MxD. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Army.



TABLE OF CONTENTS

I.	EXECUTIVE SUMMARY	3
II.	Project Deliverables	9
III.	PROJECT REVIEW	10
	Use Cases & Problem Statement.....	10
	Scope & Objectives.....	11
	Planned Benefits.....	11
IV.	Technical Approach	12
	What is an Industrial Real-Time Linux Server?	12
	Real-Time Computing and Data Handling Frameworks	13
	Open Architecture Framework Integration Test Environment (ITE)	16
V.	Results.....	19
	System Overview.....	19
	System Requirements.....	20
	System Architecture	21
	Features & Attributes	22
	Target Users & Modes of Operation.....	23
	Integration Test Environment (ITE) Design	24
	Industrial Open Automation Framework Design	33
	Digital Twin Design	41
VI.	Discussion & Analysis	54
	Industry Impact	54
	Key Performance Indicators & Metrics	54
	Accessing the Technology	56
	Workforce Development.....	56
	Lessons Learned	56
VII.	Conclusions & Future Work	59
	Next Steps & Challenges	59
	Transition Plan.....	59
	APPENDICES	62
	Appendix A: Definitions	62
	Appendix B: Demos	64
	Appendix C: Validation & Testing.....	65
	Appendix D: User Resources.....	66



I. EXECUTIVE SUMMARY

Overview

The Process Manufacturing Digital Twin - Open Architecture Testbed Framework consists of a digital twin and open process automation sandbox with a self-contained Integration Test Environment (ITE) platform. This combined system and software platform demonstrates and evaluates digital twin and open process automation technologies with cross-vendor systems. It also allows for manufacturers to develop, test, and evaluate new technologies without interrupting production operations and without costly R&D investments.

This project is a collaboration between MxD, Applied Dynamics International (ADI), the University of Michigan Barton Research Group, and The Dow Chemical Company.

Project Solicitation

The project started when MxD and its members proposed addressing the challenges of “plug and play” connectivity and interoperability between different vendors. According to Rene Reinbigler, an enterprise architect at Merck, at a BioPhorum workshop, “communication between equipment from different suppliers will no longer be a nice to have, it will become mandatory in the industry.” MxD was searching for a computing architecture that allows swapping of different technologies with minimal levels of reconfiguration required and relying heavily on the use of standards, the Open Process Automation Standard (O-PAS), developed by the Open Process Automation Forum, a member of the Open Group, was of most interest to the project.

The project involved the development and implementation of a framework for gathering and analyzing all the data from a process manufacturing line necessary to improve visibility and control. This framework is a key first step in being able to implement proof-of-concepts for ‘mobile worker’, cybersecurity, predictive maintenance, and other digital twin use cases. This well-defined vision has the factory connected to: on-premise computing at the edge; cloud computing; and, the enterprise. This project looked to demonstrate key enabling technologies to achieve that digital manufacturing vision.

ADI believed the project scope was a great fit for the ADEPT software platform and believed the development of a demonstrator industrial computing rig, for this project called an Integration Test Environment (ITE) rig, would allow the concepts to be rapidly developed and clearly demonstrated.

Project Team

ADI’s partnership with the Barton Research Group in open process automation connectivity and Digital Twin technology is part of how ADI became aware of the MxD project opportunity. The scope of this project made it a natural fit for the Barton Group to partner with ADI for the purposes of rapid prototyping and demonstrating their best-in-industry Digital Twin architecture and technology platform.

The Dow Chemical Company joined the team as the project’s “Pilot Manufacturer” and voice of the customer, and provided a key input to the project with the Dow Statement-of-Work (SOW) document. The Dow SOW document provides a detailed “customer discovery” report including solution objectives, key use cases, target demographic, solution architecture, ranked must-haves, and ranked nice-to-haves.

Solution Overview

The project team designed, implemented, and delivered an industrial open process automation framework, based on ADI's ADEPT software platform, that included multiple digital twin and open process automation apps, running in real-time, on the purpose-designed and built ITE rig. Figure 1 illustrates a logical and functional view of the Digital Twin and Open Automation Framework. ADI's systems engineering team performed an agile design engagement with the Dow/MxD team, designed, and built the ITE as an industrial-packaged computing rig based on Commercial Off-the-Shelf (COTS) industrial computing equipment, including:

- Dell edge server
- NVIDIA machine learning server
- Cisco industrial network gear
- Eaton smart power distribution equipment
- Texas Instruments industrial communication prototyping equipment

Figure 2 shows a rendering of the ITE industrial computing rig, now installed on the MxD factory floor.

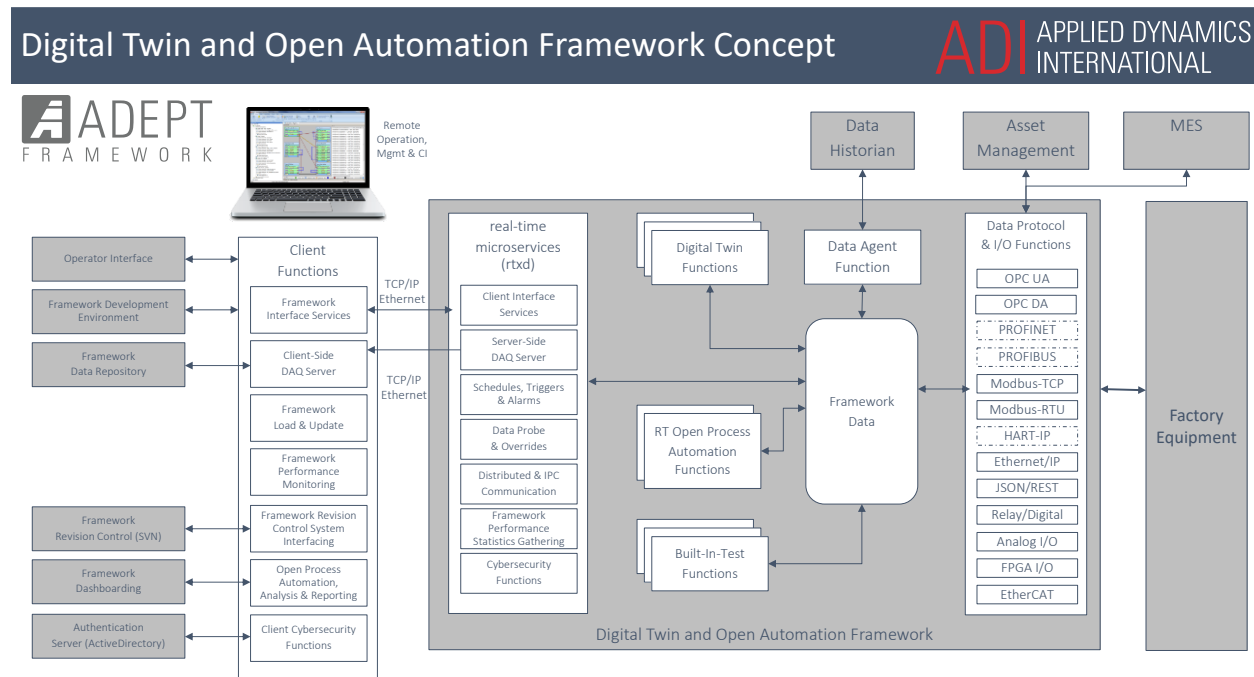


Figure 1: Industrial Open Automation Framework Concept

One of the unique features of the ADEPT software platform is its support for standards-based models. A model built using any one of dozens of commercial modeling tools, such as Matlab/Simulink and Siemens Flomaster, can be dragged into ADEPT, assigned to an app and an edge server, connected to factory data and other apps, and deployed to the computing framework with relative ease. This allowed the Barton Group team to design and prototype their Digital Twin apps using their preferred modeling tools and enabled fast iteration between the ADI modeling and integration team and the Barton Group algorithm development team.



Figure 2: Integration Test Environment (ITE) Rendering

Apps and Models

A key goal of this project was to demonstrate Digital Twin (DT) and Open Process Automation (OPA) applications, or apps, running in real-time within the ADEPT plug-and-play testbed framework. The project did not support connecting the testbed framework to a real factory, at least for this initial project scope, due to schedule and budget constraints. In order to rapidly develop and validate the DT and OPA apps, a portion of a process manufacturing line was simulated and hosted within an ADEPT plug-and-play computing framework to enable the simulated plant to be connected to, and virtually operated alongside, the DT and OPA apps. The table below summarizes the apps and simulations demonstrated within the testbed normal operation.

Function Name	App or Sim	Function Description	Developed In
Pump Health Digital Twin	App	Monitors the condition of equipment in the system by detecting degradation onset, estimating and classifying health state	Matlab/Simulink
PID Controller Health DT	App	Estimates the health state of the PID flow control loop	Matlab/Simulink
Process Health Digital Twin	App	Estimates the health percent and detects anomalies in the process	Matlab/Simulink



ITE Health Continuous Built-in-Test	App	Estimates the health of the ITE computing rig, plug-and-play computing framework, and hosted apps	Matlab/Simulink
Startup Readiness Check	App	Performs a cross-function check of readiness to start the manufacturing process by checking the readiness of connected systems at various level of ISA-95	Python
Framework Auto-Recovery	App	Performs a periodic check on the health of each app running within the OPA plug-and-play framework and performs recovery reset actions if/when needed.	Python
Process Simulation	Sim	Provides high-fidelity multiphysics simulation of a flow control unit with accurately modeled, and dynamically-controlled, process and equipment degradation.	Matlab/Simulink SimScape and C/C++

Industrial Plug-and-Play Connectivity Interfaces

The early project requirements discovery efforts included defining the industrial data connectivity needs required to support Dow's modern and legacy process manufacturing lines. The table below summarizes the industrial plug-and-play connectivity demonstrated on the testbed as well as those not demonstrated but natively supported on the testbed.

Interface Description	Demonstrated or Supported
OPC UA	Demonstrated
Modbus TCP	Demonstrated
Analog Input	Demonstrated
Relay/Digital	Demonstrated
UDP Ethernet	Demonstrated
Modbus RTU	Supported
High-Speed (FPGA) I/O	Supported
Ethernet/IP	Supported
EtherCAT	Supported
CANOpen	Supported
RS-422/485/232	Supported
Analog Output	Supported
JSON/REST	Supported
PROFINET	Future support
PROFIBUS	Future support
HART-IP	Future support



Project Achievements

This project achieved the following:

- Demonstrated how an **OPA Plug-and-Play Computing Framework** may be used to:
 - Achieve factory data interoperability, connecting factory equipment using a range of industrial communication interfaces from multiple vendors
 - Connect the factory to the enterprise at various levels of ISA-95
 - Perform rapid prototyping development of digital twin and open process automation apps in a representative cross-vender process manufacturing environment
 - Host digital twin functions built into real-time apps
 - Host open process automation functions build into real-time and non-real-time apps
 - Implement advanced, real-time on-premises and virtual cloud-deployed OPA computing capability using COTS industrial computing and networking equipment
 - Leverage existing Windows and Linux Cybersecurity policies to realize cybersecure OPA capability
- Demonstrated how **Digital Twin Functions** may be used:
 - To support equipment and process condition monitoring for process manufacturing
 - To provide health monitoring of equipment by detecting degradation onset, estimating and classifying health state
 - To provide health monitoring of process units, or entire process manufacturing lines, by estimating the health percent and detecting anomalies in the process
 - To provide Control Loop Performance Monitoring (CLPM) capability for process PID control loops
 - As an extensible DT solution that scales up from equipment, to units, to process manufacturing lines, and to support multi-facility operation
 - In cooperation with OPA computing frameworks to enable organizations to more quickly evaluate operations, test assumptions for innovation, and improve capabilities
- Demonstrated the feasibility of using the ADEPT software as a real-time connectivity and app hosting solution within a digital process manufacturing facility
- Demonstrated the feasibility of using the ADEPT software as a Digital Engineering tool for rapid prototyping, test, and evaluation of open process automation and digital twin capabilities and technologies
- Designed and implemented an industrial Ethernet networking architecture to support OPA implementations
- Designed and built an industrial computing rig, providing a sandbox for OPA technology evaluation, integration, and deployment



- Demonstrated OPC UA client-server communications within the ADEPT industrial computing software platform

Project contributions:

- This project provided an opportunity to mature the Barton Research Group DT architecture and technology platform
- This project showcased for industry, an approach for filling a key technology gap within the greater Industrial IoT market for plug-and-play industrial computing frameworks
- This project provided traceability of the integrated demonstrator capability to prevailing industry standards



II. PROJECT DELIVERABLES

The following list includes all deliverables created through this project. These deliverables will be referenced throughout this report and should be accessible on the MxD membership portal in accordance with the rights defined by the Membership Agreement.

Table 1: Project Deliverables

#. DELIVERABLE NAME	DESCRIPTION	FORMAT OF DELIVERY
1. Requirements & Feasibility Report	Report on requirements for open architecture frameworks and feasibility analysis	Technical Report
2. Integration Test Environment (ITE) Rig	Linux-based edge server with network and industrial interface boards, cabling, rack, mechanicals, and interface panels	Hardware Deliverable
3. Digital Twin Models	An extensible DT solution to support equipment and process condition monitoring for process manufacturing	Software Deliverable
4. Digital Twin and Open Automation Framework	Demonstration framework that includes digital twin and open process automation applications and user interfaces running on a high-end industrial Linux server with industrial interfaces to communicate with simulated or real factory equipment	Software Deliverable
5. Simulated Factory Framework	Factory framework that includes applications and user interfaces for emulated industrial interfaces connected to a physics-based process manufacturing simulation	Software Deliverable
6. Display wall, monitors, and Windows PCs	Display wall with informative text, three large display monitors, an industrial Windows PC for running the ADEPT client tools and driving the center monitor and two smaller Windows PCs for driving the side displays, includes a perpetual license for the ADEPT software	Hardware Deliverable
7. Standards Review Report	Document compliance of solution to prevalent standards for digital manufacturing	Technical Report
8. Project Final Report	Documentation of the ITE and open framework system, including approach and diagrams	Final Report

III. PROJECT REVIEW

This section includes use cases and a problem statement for the project, the scope and objectives, and planned benefits.

Use Cases & Problem Statement

Digital Twin Use Case

The benefits of digital twin technologies are not fully realized in practice. Digital twin being a relatively new concept in process manufacturing is evolving and expanding in breadth and reach. Demonstrating that digital twin can be backward integrated into existing manufacturing operations including physical plant asset management, engineering, maintenance, and supply chain logistics in a cohesive architecture using unifying work processes will provide the evidence needed for confident implementation in production environments.

A functional digital twin allows the organization to more quickly evaluate operations relative to market conditions and test assumptions for innovating and improving capabilities, therefore improving market position in the competitive global environment.

Open Process Automation Use Case

Seamless intuitive incorporation of varied solutions into a common ecosystem addresses a substantial roadblock to implementing continuous improvement in process manufacturing. The long-held practice of customizing process control using a single technology platform provided considerable benefits in previous decades, however recent technology advances and expectations of the present-day digital worker and digital consumer demand a change.

The most beneficial solutions for process manufactures are not made available from any single supplier of digital solutions at any single point in time. Remaining competitive requires integration of a multitude of seemingly disparate technologies in such a way as to act as a single organism. Applying open process automation concepts removes innovation roadblocks by allowing manufacturers to seamlessly integrate solutions from multiple providers. No longer locked into a single control platform, manufacturers will enjoy greater choice and be able to implement stepwise improvements continuously throughout the life of manufacturing assets and management systems.

Problem Statement

Digital Twin and Open Automation technologies show great promise to reduce risk, reduce downtime, reduce energy consumption, improve safety and improve efficiency. However, there are some significant challenges to adopting these technologies:

Table 2: Challenges for implementing Digital Twin and Open Automation technologies

Challenge #1: Sole source lock-in	Sole source lock-in arises from traditional manufacturing technologies that are vendor-specific and do not allow easy integration of components from other suppliers, thus inhibiting innovation and reducing the benefits that can be derived from newer technologies.
Challenge #2: Access to data	Accessing, processing, synchronizing and aggregating data that are generated throughout a facility poses a significant challenge, especially when the data come from different networks, different industrial interfaces and different ISA-95 layers.
Challenge #3: Access to equipment	Developing, evaluating, and verifying Open Automation and Digital Twin solutions involves extensive test and evaluation that is often not feasible on actual equipment because 1) critical industrial equipment is often in use and sometimes cannot be taken offline for testing and 2) development and



	testing, especially early on, can involve greater risk of negative outcomes that can put equipment, personnel and/or the environment in harm's way.
--	---

Scope & Objectives

The primary objective of the project is to demonstrate viable proof-of-concept Digital Twin and Open Automation solutions for use in the process industry in alignment with principles stated in Part A:

1. **COLLABORATION:** Collaborative platform for generating value for MxD members
2. **COST EFFECTIVENESS:** Lower cost of embracing digitization without disruption to production facilities
3. **RISK MANAGEMENT:** Minimize the overall risk for process industry and extend application of proven digital technologies beyond discrete manufacturing
4. **INNOVATION:** Broaden innovation scope by engaging solution/service providers and end-users at the early stages of technology development and evaluation
5. **ADVANCEMENT/SYSTEM INTEGRATION:** Accelerate screening, piloting, and adoption of digital tools/technologies (sensors, cybersecurity, data analytics, etc.)
6. **BEST PRACTICES:** Jointly address unique operational and regulatory requirements of process industries, e.g. safety requirements, process optimization, and process disruption avoidance
7. **WORK FORCE DEVELOPMENT:** Provide training for the workforce of the “future factory”

Planned Benefits

The benefit of a process focused digital twin, open process automation environment at the MxD facility for a company such as The Dow Chemical Company is the early testing of promising project ideas in a neutral environment with no limitations brought on by established company regulated infrastructure. In a pre-defined company infrastructure, the early testing takes time, sometimes comes with considerable risk, and requires realignment of several human resources. Pilots and related testing done outside of the established company infrastructure are more likely to demonstrate beneficial results much earlier leading to accelerated deployment in the active manufacturing environment.

Another benefit is the comparison of the needs of the discrete industry vs. process industry which can be discussed with other members of MxD in a neutral environment using tangible equipment. Many vendors are “at home” in the discrete industry and view the process industry as something different. Process Manufacturing Part B will demonstrate those differences, as well as the similarities.

IV. TECHNICAL APPROACH

To achieve the goals of the Digital Twins for Process Manufacturing project, the project team leveraged an Industrial Open Automation Framework to create a distributed, digital twin platform using Industrial Real-Time Linux Servers. ADI has developed and matured the computing and data handling technology included in the Industrial Real-Time Linux Server and Frameworks concepts within the Cyber-Physical Product Development and Model Based Systems Engineering (MBSE) market spaces and, in particular, within the aerospace and defense industry.

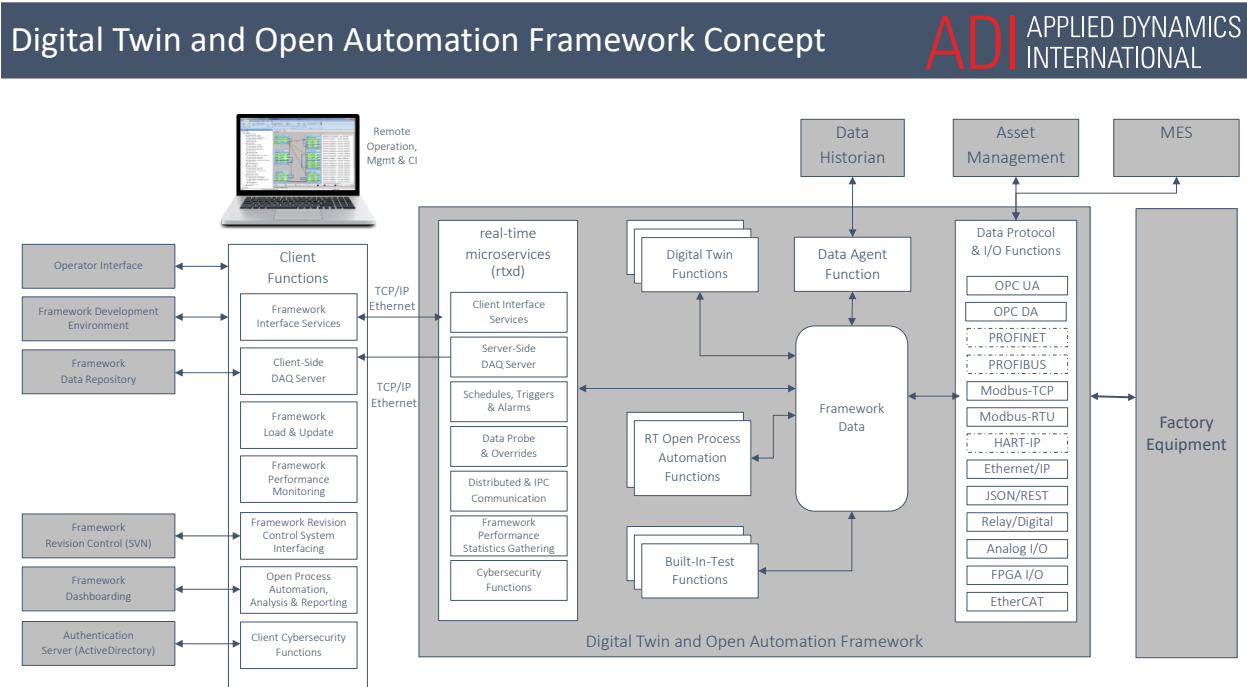


Figure 3: Industrial Open Automation Framework Concept

What is an Industrial Real-Time Linux Server?

An Industrial real-time Linux server is a computer server, low-cost or very high-end, with a real-time Linux operating system installed and providing a set of services for time-deterministic, frequency-based, computing and data handling. This real-time Linux server is used to connect your industrial capability, e.g. test facility, manufacturing line, electrical power system, into a Linux computing environment for the purpose of adding capability, e.g. predictive quality, anomaly detection, operational optimization, live data analysis, supervisory control, predictive maintenance.

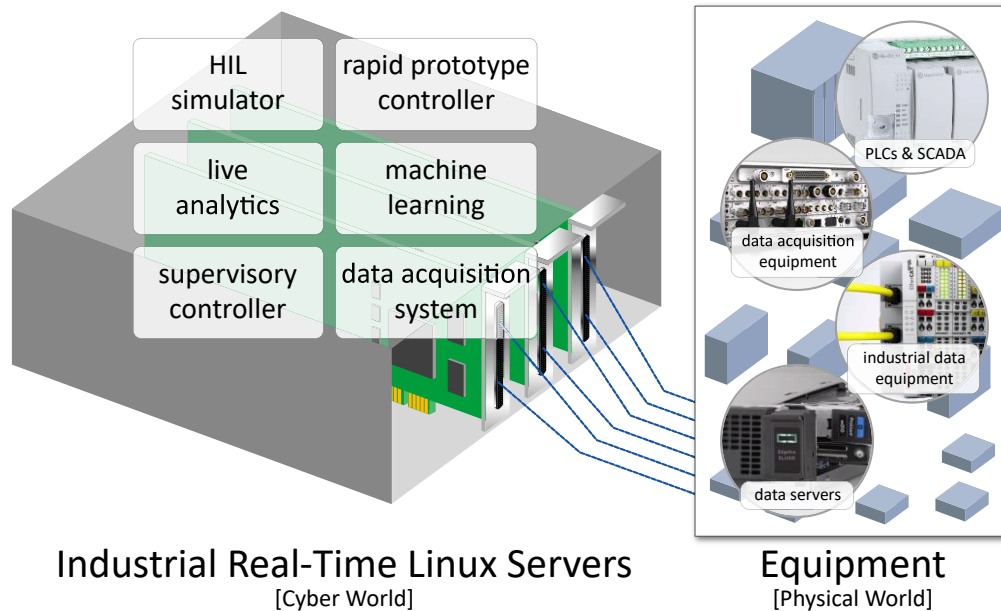


Figure 4: Industrial Real-Time Linux Servers

Common Misconception: Industrial Modernization Doesn't Need Real-Time

"If my industrial modernization project doesn't involve real-time control or real-time data acquisition, why would I need to consider real-time Linux computing?"

Some industrial applications are obvious and require real-time, closed-loop control and/or real-time supervisory control. For other applications, it is not always so obvious. Consider real-time computing as a tool for your industrial computing and data handling. You may have many functions, analysis, and data handling tasks that will not operate in a real-time, once-per-step type of computation. Many tasks and functions will be asynchronous and execute on command or in response to a condition or alert. A real-time Linux computing framework sets a high-resolution time reference for your computing environment. Accurate, microsecond resolution time measurement of events such as time of task initiation, time to complete task execution, and statistical variance of task behavior can be very useful for identifying and solving anomalous, difficult to find bugs in your operational systems. Real-time computation and analysis is very helpful and important for monitoring data throughput of mixed Industrial Ethernet and legacy industrial serial/network interfaces, e.g. Modbus, CANOpen, etc. Real-time computing and data handling may not be required for your application or it might turn out to be a very powerful tool, either way, open source real-time Linux costs the same as open source standard Linux. They are both free software.

Real-Time Computing and Data Handling Frameworks

A real-time computing and data-handling framework (real-time framework) is made up of one or more real-time Linux servers operating in coordination and with time synchronization. The real-time framework provides a flexible computational structure allowing these Linux servers to be interfaced with the real world, e.g. equipment, PLCs, sensors, actuators, data acquisition devices, etc. and operated as a time-synchronized cyberphysical system.

When a real-time Linux server boots, the rtxd process is spawned as part of the boot process. A real-time framework is uploaded to one or more Linux servers by transferring the set of assemblies to the participating servers along with configuration, data connections, and data

dictionary XML files. After uploading the framework files to the server, a framework load is commanded. This load command spawns the rtxmgr process, which proceeds to execute and control each real-time assembly executable. Client tools interface with the framework through its TCP/IP API to get and put values, setup live streaming data, and interact through a wide set of functions.

The real power of industrial real-time Linux servers comes when you make use of multi-server distributed frameworks. The rtx daemon makes distributed computation and data handling a greatly simplified task by including both inter-process communication and inter-server communication. A primary rtxd server is assigned to act as synchronization and scheduling master for all Linux servers participating in the framework. Connections between data dictionary items are defined and if connected assemblies are allocated to different servers then the rtxd services automatically transfers data at each frame or selected frame multiple. The rtxd TCP/IP APIs allow multiple clients to connect and interact with the framework for management and operation.

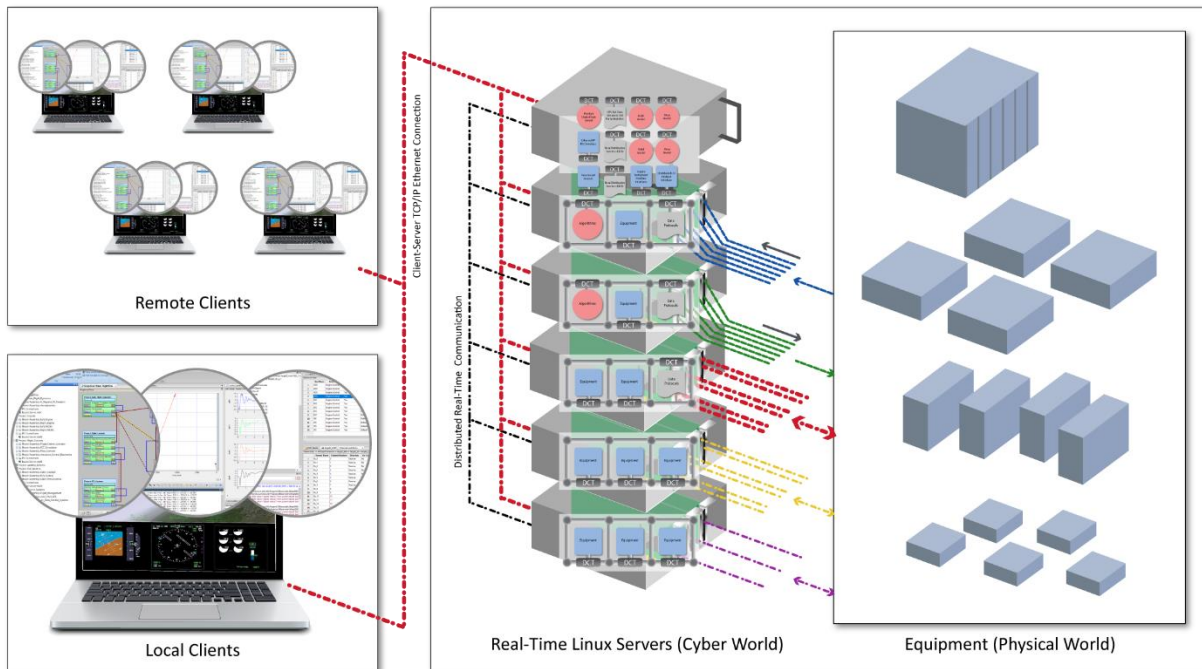


Figure 5: Distributed Real-Time Framework Executing on Multiple Real-Time Linux Servers

The rtx daemon – Linux Real-Time Executive

The “rtx daemon” is a set of Linux services, which runs on ARM- and Intel-based computers and transforms them into full-featured real-time computers that are internally instrumented with performance, latency, jitter, and other precision time-based measurements for all aspects of the computer architecture. The rtx daemon (rtxd) software was developed, refined, and optimized by ADI over three decades. The rtx daemon dramatically reduces the effort in deploying industrial computing and data handling frameworks.

By installing rtxd on your real-time Linux servers you get, right out of the box, a set of optimized real-time services with a multi-processor, multi-server distributed real-time computing and data handling architecture and a scalable client-server paradigm. The rtxd dramatically reduces the engineering and IT cost of modernizing your computing and data handling assets.

Performance-Optimized Industrial Computing

Real-time computing was once considered to be a bit of a black art. There are many reasons and opinions as to why it is viewed as unusual computing. A popular view suggests real-time computing tends to be foreign due to the way we typically learn to write software. We start with our “hello world” and we interact with the keyboard and the screen. Next we learn to allocate some memory for our program using a malloc(). Then we learn to read from and write to our hard disk drive for long-term data storage. We often are introduced to programming through the concept of a main() entry point and a procedural view of computing and task design.

In real-time computing, we switch to a frequency-based set of entry points, usually associated with a step() function in your code module. And there are a set of rules for implementing quality real-time code, including avoiding use of malloc(), avoiding disk reads and writes, and other best practices. In general, you try to avoid iterative numerical methods, e.g. solvers, integration algorithms, but sometimes they are unavoidable.

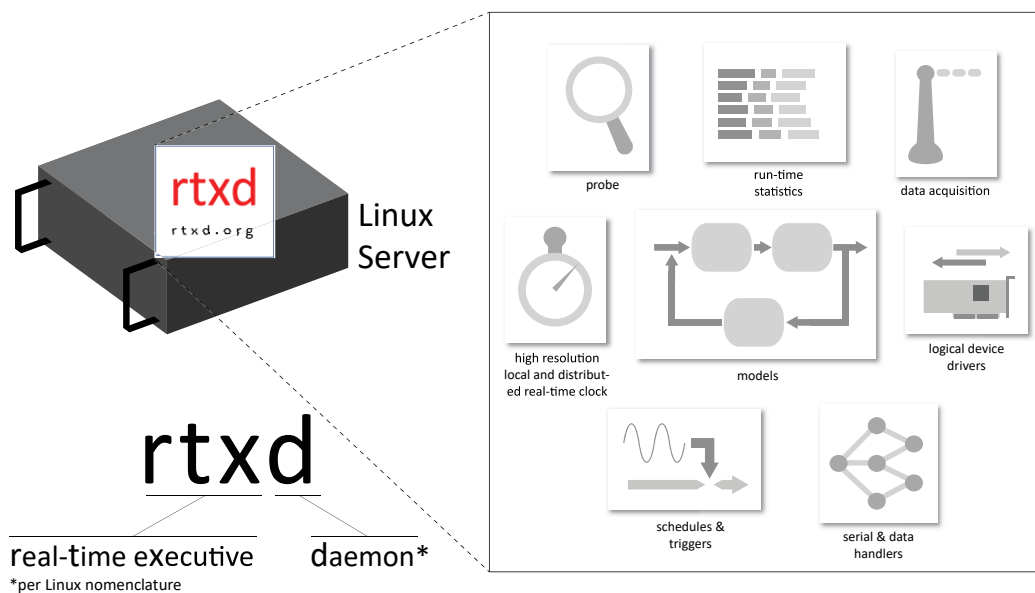


Figure 6: rtxd Linux Real-Time Executive

The rtxd software was designed and written with great effort invested in optimizing the services and architecture for real-time performance. A fundamental aspect of the rtxd high-performance real-time capability is associated with its data-centric “Data Dictionary” concept that makes use of shared memory structures and APIs for accessing data from all assemblies and connecting data from one assembly process to another with ultra-low-latency and computational overhead. This shared memory structure lends itself naturally to distributed, multi-server real-time framework deployments.

Real-Time Linux versus Commercial RTOS

The commercial Real-Time Operating System (RTOS) market is big business with serious industry players, e.g. Intel. Ever since its release in 1991, Linux has been hailed by many thought leaders in the computer science community, as an existential threat to the RTOS market. In the early 2000’s, Linux saw increased application for consumer embedded electronics devices. The commercial RTOS industry demonstrated the poor real-time performance exhibited by Linux and was able to draw a line in the market for where Linux would begin stealing market share and where commercial RTOS would continue to dominate and generate significant annual software licensing and support revenue.



In 2004, the Linux community released the “RealTime Preempt” patch, or RT Preempt. The Linux operating system, with its generic kernel, exhibits poor real-time performance as measured by real-time jitter. A key function of any real-time operating system is to allow frequency-based computing where a set of algorithms and/or data handling tasks are performed at every step, and executed at some frequency. Real-time jitter represents the time-based error associated with the operating system’s ability to begin computation exactly at the beginning of every step. Today’s commercial RTOS offerings typically perform with 3 μ s to 5 μ s of real-time jitter at three standard deviations. Traditional Linux, depending on the kernel configuration, could see real-time jitter as high as 500,000 μ s. RT Preempt dramatically improves the performance of open source Linux. Today, well-configured open source Linux with RT Preempt patch installed performs within the 3 μ s to 5 μ s jitter at three stddev range of commercial RTOS offerings. This means your engineers and IT folks no longer have to spend \$10K to \$20K per year for RTOS development tools, they can leverage the vast libraries of free open source software available for Linux, and you can benefit from far greater access to well-trained Linux development talent.

Open Architecture Framework Integration Test Environment (ITE)

This project will design, integrate, verify, validate, and demonstrate the open architecture framework on a set of computing equipment packaged as an Integration Test Environment (ITE). The purpose of the ITE is to provide:

A demonstration platform to communicate the open architecture framework with a digital twin example framework instance

Dedicated hardware to be used for verification, validation, and demonstration of the Integrated Digital Twin Framework Implementation

An experimental platform to benchmark and characterize the performance of the framework, e.g. data latencies, real-time jitter, economic computing cost analysis on different industrial computing and data handling tasks, Intel Xeon versus ARM processor performance for industrial IoT

Major equipment in the ITE includes:

- Intel Xeon based Linux server – providing the main computing and data handling server
- Array (quantity ~6) of Arm single board computers – providing industrial equipment emulation
- NVIDIA Jetson Xavier (quantity 2) Arm/GPU Industrial IoT prototyping servers
- Time-synchronized Ethernet network, e.g. IEEE-1588 PTP network switch
- Connector interface panel
- Standard 19-inch computer rack
- Rack-installed power distribution equipment

A deliverable framework demonstration shall be designed, integrated, verified, validated, and demonstrated leveraging ADI’s ADEPT software platform. ADEPT represents a promising candidate for pilot implementation of the open architecture framework. ADEPT is an industrial computing and data handling, client-server software platform developed by Michigan-based Applied Dynamics (www.adi.com). ADEPT is an open architecture, data-centric, and highly extensible real-time Linux computing paradigm with a strong level of technology maturity through more than a decade of application on demanding aerospace and defense industrial computing deployments.

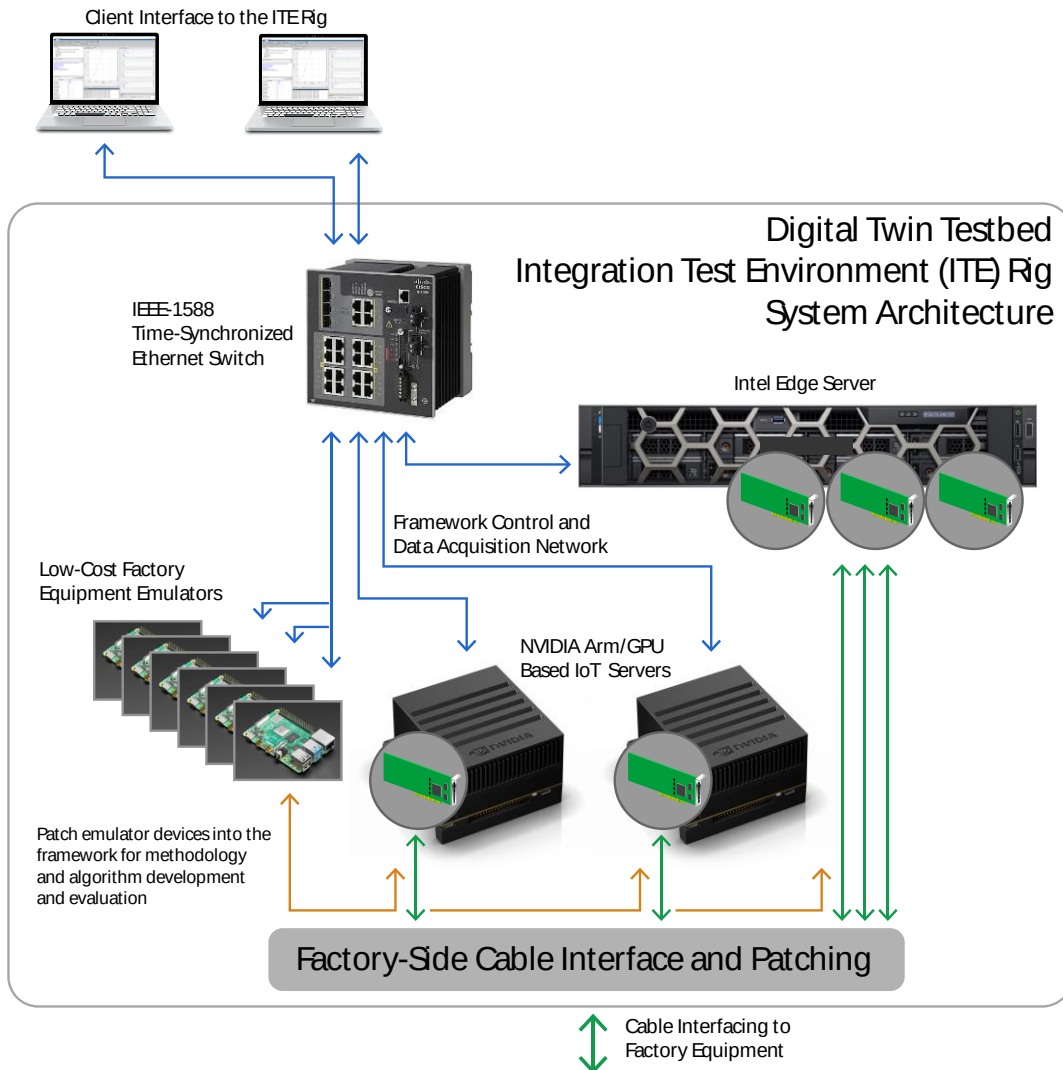


Figure 7: ITE Rig Architecture

Key strengths of the ADEPT software platform include:

Flexibility – ADEPT’s industrial computing and data handling “framework” paradigm enables models to be easily added, assigned to processing resources, connected to data and saved as a framework instance to support an unlimited set of operating modes and equipment/models included within the scope of operation.

Data Centric, Distributed Architecture – ADEPT provides a “God’s Eye View” of the data within models, coming from equipment, and moving through the deployment

Cybersecurity – ADEPT’s Linux real-time computing capability enables the testbed to leverage existing cybersecurity hardening policies

Computing Efficiency – Through decades of code refinements and optimizations the ADEPT server code reduces the cost of computing hardware by getting more out of the same CPU(s)

Faster Deployment – ADEPT reduces the engineering and IT effort to assemble an industrial computing and data handling capability using plug-and-play interaction



The ADEPT software brings together a range of technologies highly relevant to the Digital Twin test bed:

- Supports popular equipment interface communication standards: EtherNET/IP, Modbus TCP, Modbus RTU, CANopen, EtherCAT, etc.
- Supports open source and commercial Linux distributions
- Supports 100+ off-the-shelf computer boards from popular equipment vendors: analog and digital I/O, network interface boards, GPU boards, single board computers
- Supports data file format standards: Matlab (.mat), Comma Separated Values (.csv), Hierarchical Data Format (HDF5)
- Supports time synchronization standards: Precision Time Protocol (PTP), IRIG-B, various time synch network protocols
- Supports leading enterprise communications standards: Distributed Data Services (DDS), MQTT, OPC-UA
- Supports leading model standards: Matlab/Simulink, FMI/FMU, TensorFlow, C/C++
- Support for popular Machine Learning (ML) frameworks

Perhaps most important for this Digital Twin testbed, ADEPT offers virtually unlimited extensibility, whether adding driver support for new interfaces, incorporating new model types, adding new communications protocols, integrating powerful, freely available open source software, or interfacing to other systems. In addition to a comprehensive set of features/services available out-of-the-box, ADEPT allows the user to make unconstrained use of the available Linux computing environment to incorporate new technology as it emerges.

Because ADI's proposal for the open framework leverages existing, and mature, open source based software technology, the main tasks for this project include:

- Requirements and Feasibility Report
- Digital Twin and Open Automation Framework Example
- Integration Test Environment (ITE) Rig
- Process manufacturing simulation model
- Display wall, monitors and Windows PCs
- Demonstration of the Digital Twin and Open Automation example running on the ITE
- Documentation and Final Reports



V. RESULTS

This section includes a list of technology deliverables, a system overview, the system requirements, the system architecture, the features and attributes of the technology deliverables, the target users and modes of operation, and software development and design documentation.

The table below will summarize these top-line deliverables, but we will cover each in more depth, including impact & output of each deliverable.

Table 3: Technology Deliverables

# / DELIVERABLE NAME	DESCRIPTION	FORMAT OF DELIVERY
2. Integration Test Environment (ITE) Rig	Linux-based edge server with network and industrial interface boards, cabling, rack, mechanicals, and interface panels	Hardware Deliverable
3. Digital Twin Models	An extensible DT solution to support equipment and process condition monitoring for process manufacturing	Software Deliverable
4. Digital Twin and Open Automation Framework	Demonstration framework that includes digital twin and open process automation applications and user interfaces running on a high-end industrial Linux server with industrial interfaces to communicate with simulated or real factory equipment	Software Deliverable
5. Simulated Factory Framework	Factory framework that includes applications and user interfaces for emulated industrial interfaces connected to a physics-based process manufacturing simulation	Software Deliverable

System Overview

The objective of Process Manufacturing Part B is to provide The Dow Chemical Company and other MxD members and guests with an environment for use as a proving ground for Open Automation and Digital Twin approaches deploying a varied mix of existing and new technologies.

The purpose for exploring and proving Digital Twin and Open Automation technologies is to identify more intuitive and reliable solutions to the costly problem of technological lock-in; a condition that results when traditional single-sourced solutions are propagated. Solving the lock-in problem will speed innovation, hence competitiveness in the global marketplace.

Process Manufacturing Part B is intended to deliver a Digital Twin using Open Automation concepts that demonstrates functionality that allows an organization to more quickly evaluate operations relative to market conditions and test assumptions for innovating and improving capabilities, therefore improving market position in the competitive global environment.



Utilizing MxD Process Manufacturing Part A concepts, the Process Manufacturing Part B project seeks to leverage emerging aspects of Digital Twin and Open Automation in a responsible and sustainable manner.

Sustainability requires solutions that are secure, add value, are expandable, are supportable using existing skills and crafts or otherwise easily transferable middle-skills, avoids complex licensing, and are generally immune to sudden obsolescence.

Responsible deployment requires solutions that endure, occupy less space, produce less waste, and consume less power – including demands on support systems (such as HVAC).

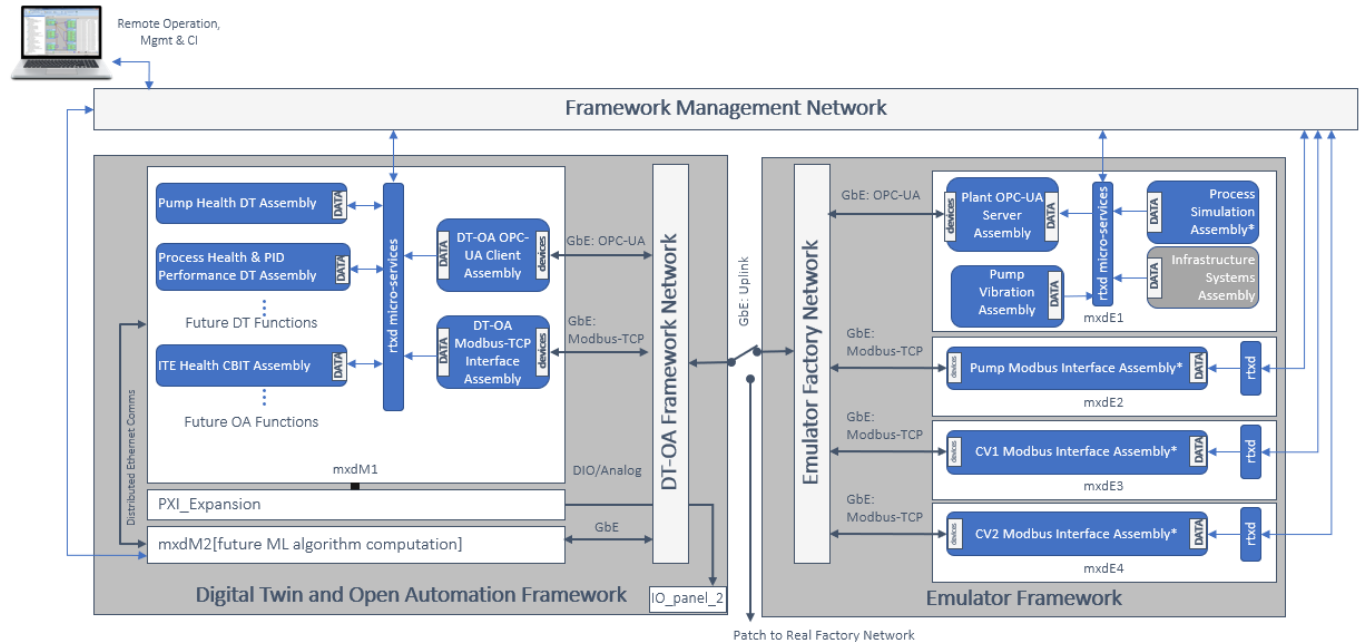
A forward-looking approach is required. This includes alignment and the possibility of integration with Part A or similar facsimile, as well as other manufacturer hosted remote environment. The pilot will be considered successful when it is determined that a viable foundation has been provided for open integration of out-of-scope architecture components including, but not limited to API Handlers, Decision Makers/Logic Solvers, Data Services, and Factory Floor components. This forward-thinking approach paves the way for an eventual Part C Digital Twin and Open Automation project that would thoroughly test a variety of Process Manufacturing solutions from cloud/enterprise to plant floor.

Many Process Manufacturing operations require long uninterrupted runtimes. Runtime goals can be as long as 10 years. Manufacturing operations desires automation solutions capable of indefinite runtime, hence having zero impact on plant turn-around (extended shutdowns for planned maintenance activities) scheduling. The solution piloted is expected to perform in such a way that it is clear that these demands can be met.

System Requirements

The requirements for this project were jointly developed, analyzed, reviewed and agreed by all project participants using the ADI requirements compliance matrix template and process. This requirement development process involves documenting the requirement text, identifying how the requirement will be met and assessing the level of risk that the requirement may not be fully achieved within the project timeline or budget.

See Deliverable #1 Requirements and Feasibility Report for details.



*Data communication from Process Simulation Assembly to Pump, CV1 and CV2 Modbus Assemblies handled using ADEPT distributed Ethernet across F/W Mgmt Network

Figure 8: DT-OA and Emulator Frameworks

System Architecture

The digital twin, open process automation, and emulator capabilities of the Process Manufacturing ITE are split across two main frameworks: one framework that handles the digital twin and open process automation capabilities and a second framework that handles the process emulation. The emulator framework contains several assemblies used to generate simulated process data, package that data into industry standard communication protocols, and send that data across the ITE network. That data is then taken in via several dedicated data protocol interface assemblies within the DT-OA framework and used by the digital twin and open process automation assemblies to monitor process and equipment health. This setup is diagrammed in Figure 8.

Each framework runs on a separate server or collection of servers. The DT-OA framework runs entirely on the mxmM1 Dell PowerEdge R540 server, while the Emulator framework is distributed across four different servers: one Nvidia Jetson AGX Xavier and three Texas Instruments AM3359 Industrial Communications Engines (TI-ICE). The mxmE1 server, Nvidia Jetson AGX Xavier, hosts the physics-based process simulation model and another model that packages certain data elements into OPC-UA packets to be sent to the DT-OA framework, while the mxmE2 – mxmE4 servers, TI-ICE's, package the process simulation data into ModbusTCP packets to be sent to the DT-OA framework.

These components are connected through a powerful industrial-grade ethernet switch, the Cisco IE 4010, and a smaller managed network switch, the StarTech 8-port Industrial Ethernet Switch. These two switches, along with an ethernet patch panel, allow the user to connect the DT-OA framework to either the emulated process data or to real factory signals.

Figure 9 illustrates the relationship between each component and the network.

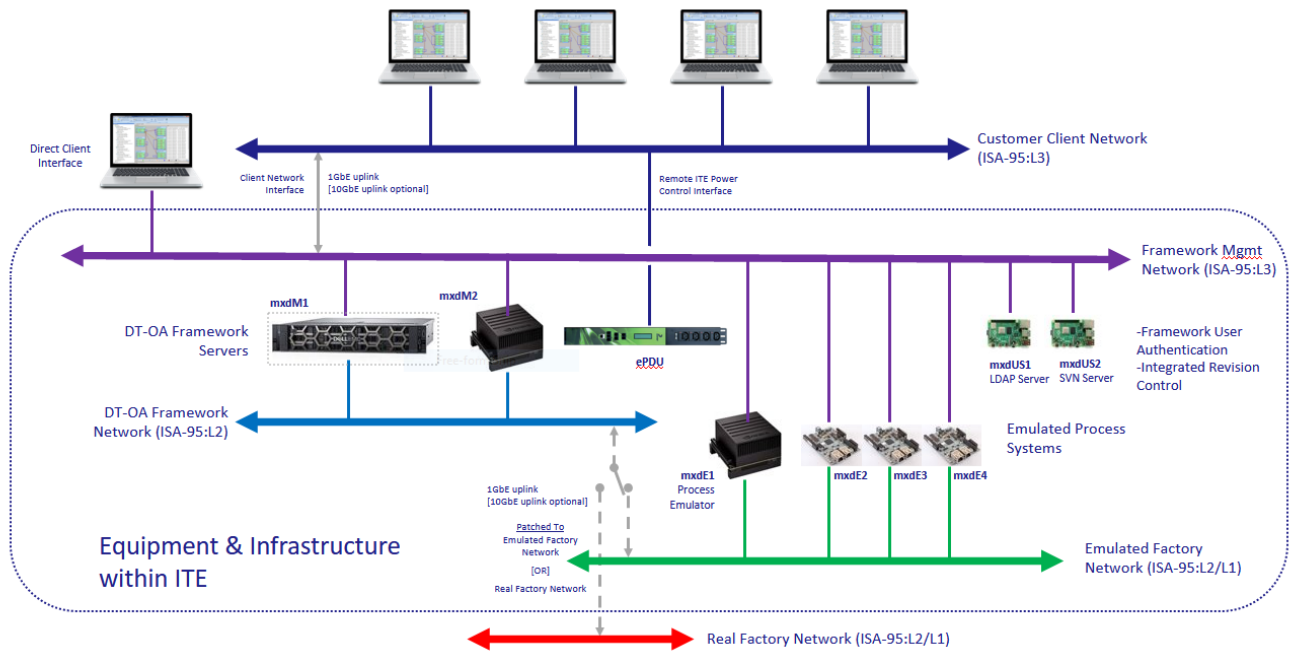


Figure 9: System Architecture

Features & Attributes

Digital Twin

A Digital Twin is a purpose-driven dynamic digital replica of a physical asset, process, system, or product; supporting improvement of a manufacturing environment (e.g., reduce unscheduled downtime, generate a production plan, and improve quality). A DT provides a value-add capability in terms of detection, prediction, and/or prescription capabilities.

The DT-OA Framework contains two digital twin assemblies designed by UM and implemented in Matlab/Simulink. The Pump DT monitors pump vibration spectra to determine the pump health. The Process DT monitors process and control command data to determine the process and controller health. These DT applications are executed on the MxdM1 server.

Open Automation

Open Automation (OA) technologies allow products from multiple vendors to interoperate as a single, cohesive system, thus enabling increased options and competition for delivering future innovation.

The DT-OA Framework contains digital twin applications, a Continuous Built-In Test assembly, Data service checks, BMS checks, User-friendly interfaces, and request-driven readiness checking, in addition to computational units, devices and communication buses available for integrating additional applications.

Plant/Process Simulation

The plant simulation is a physics-based model developed in MATLAB/Simulink executed on mxmE1, one of the Nvidia rapid prototyping computers. The model represents a liquid mixing process. The flow is controlled via a PI controller with setpoint weighting. Sensor outputs include representative noise characteristics.



The process model supports fault injection capabilities with sensor and actuator biasing and noise injection. A separate pump assembly supports equipment degradation emulation.

Framework Assemblies

The DT-OA Framework and Emulator Framework contain multiple assemblies shown in Table 4. The data definition for both frameworks is available in a separate document noted in the References Table.

Table 4: Framework Assemblies

	Digital Twin and Open Automation Framework	Emulator Framework
Assemblies	Pump Health Digital Twin	Process Simulation
	Process Health Digital Twin & PID Performance	Pump Vibration
	ITE Health Continuous Built-in-Test	Plant OPC-UA Server
	OPC-UA Client	Pump ModbusTCP Interface
	ModbusTCP Interface	Control Valve 1 ModbusTCP Interface
	DT & OA Panel	Control Valve 2 ModbusTCP Interface
		Emulator Panel

rtxd Microservices

The rtxd microservices are a set of services/features provided on an rtx server. These features include client interface services, a server-side data acquisition server, data schedules and triggers, distributed and IPC communications, and framework performance statistics gathering. These features are discussed in detail in the ADEPT Help documentation.

Target Users & Modes of Operation

A wide range of providers and users will be served by Process Manufacturing Part B. Targets include suppliers, users, and standards generating organizations of both process and discrete automation solutions. Delivery and collaboration channels are to be sought out directly with MxD members and automation consortiums.

Specific roles include, but are not limited to:

1. Process manufacturers who are endeavoring to digitally enable business and operational logistics at all levels and require proof of solution viability and benefits
2. Open Automation and Automation Modernization consortiums seeking congruent industry-wide solutions for now and the future
3. Digital technology leaders providing direction within vendor and manufacturing organizations
4. Process engineers and system integrators in need of a platform to test new digital technology in a risk-free, neutral space

Additional stakeholder roles and accompanying stakeholder personas will be identified, developed and refined during the course of the project for the purpose of testing assumptions and evaluating stakeholder acceptance.

Persona Examples

Manufacturing Cybersecurity Specialist: "I want to test and validate innovative cybersecurity technologies in manufacturing networks/systems. For example, anomaly detection – active vs passive, asset inventory tools, remote-access controls, software-defined networks, whitelisting,



IIoT security models, and others. I want to prove the effectiveness, value, and usability of these technologies to reduce cyber risk to manufacturing assets.”

Manufacturing Process Plant Engineering Staff Member: “I want to be able to test a Digital Twin of my process unit including process performance, machine health, logistics capabilities, unit profitability and production schedule so I can make the best operating decisions to optimize profit from my unit with an intuitive user experience.”

Plant Operators and Technicians: “I want to have quick and easy access to the most relevant information that enables me to choose the best immediate actions required to complete my shift and set the oncoming shift up for success using technologies I am becoming more and more familiar with in my day-to-day life at and away from work. Technologies such as Augmented Reality, Mixed Reality, and AI/Knowledge Management with instant access to the technical and workflow data I need to make the best decisions.”

Process Automation, Reliability, and Improvement Engineers: “I want to provide elegant, yet simple and sustainable digital solutions that create significant value for my organization while making my plants safer, more reliable, and easier for technical staff to operate and maintain. I need the freedom to test, validate, and then implement novel solutions without undue, seemingly arbitrary vendor restrictions and without unnecessary add-ons.”

Technology Managers/Leaders: “I want to know which emerging digital technologies are most likely to deliver on their promises and provide long-term value in the real-world of process manufacturing. I believe that in recent years that without prudence, the cost to implement and maintain digital solutions quickly begin to outpace the benefits provided. I am searching for technologies that provide value and benefits while reducing complexity and long-term cost of ownership. As a technology manager I need to provide guidance and make recommendations driven by verifiable data produced using realistic environments – not fluff.”

Supply Chain and Logistics Managers: “I need to know at all times and with accuracy where things are, where things are going, and where things will be. I need to know not only the state of things as they are, but I need to be able to almost instantly evaluate all available possibilities for what could be. I am curious about things like artificial intelligence and IIoT. I want to know how these emerging digital technologies can help me pull together a complete and coherent picture of the entire supply chain from raw material inputs to customer acceptance of the delivered finished product.”

Integration Test Environment (ITE) Design

ITE Overview

The Integration Test Environment is 24U High X 1000mm Deep X 600mm Wide and weighs an estimated 304kg (670lbs). It uses an IP54 rated cabinet that protects against dust that could interfere with the normal operation, but it is not fully dust tight. The product is completely protected against solid objects. It is also protected against water splashing from any angle. The cabinet and interfaces are EMI shielded to 60 dB at 1 GHz, 40 dB at 3 GHz (tested to IEC 61587-3). The ITE is mobile via 4 casters (2x locking) and also is fork-liftable via top mounted lifting eyes. The ITE is built to IPC 610 Class 2 & 620 Class 2 standards.



Figure 10: Rendering of the ITE Rig

General ITE Design Considerations

Reliability

The ITE is comprised of industrial grade COTS components, where possible. All of the components selected adhere to UL or CE regulatory compliance. The ITE rig is fully tested towards EN/UL 61010 safety standards as well as factory acceptance testing (FAT) on the ADEPT Framework software prior to shipping.

Maintainability

The ITE is designed with ease of maintenance and expansion in mind. The front door is lockable and allows the greatest level of accessibility into the rig. The rear rack mounted panels and side panels can be removed with a screwdriver for complete unobstructed access to the ITE components and wiring. The door mounted fan includes a grill that can be removed for simplicity of filter maintenance.

Performance

The ITE is designed to be an EMI tight and IP sealed cabinet¹. EMI sealing is designed per IEC 61587-3 and the cabinet is rated to IP54. It has a single 217 CFM fan located on the door which limits temperature rise inside the cabinet to roughly 2° C above ambient. All external connections to the rig are IP rated and EMI tight.

¹ For this particular cabinet, EMI seal and IP rating is less due to demonstration features such as the side window.



Ease of Use

The ITE rig is mobile via four casters, of which, two are locking when the rig is to be stationary. The front door is locking and allows access to the front of the rig. The power switch for the rig is located at the rear of the cabinet and has a green LED light to signal that the rig is switched on.

Quality

The ITE is built to IPC class 2 standards and undergoes testing per EN/UL 61010 safety standards to ensure quality and safety. The components chosen are confirmed for proper functionality during the final testing at ADI.

Architectural Requirements

The ITE is designed to be a mobile platform with modular Inputs and Outputs (I/O) that will be operated on a factory floor. Note, the delivered product has display additions that affect it's EMI and IP rating. The ITE can be returned to the full EMI and IP rating by replacing the display side panel with a standard solid panel from the enclosure manufacturer. The ITE as delivered, is capable of providing to a device under test (DUT) all interfaces necessary to perform closed-loop testing. The ITE interfaces can be simulated, emulated or connected to real devices.

ITE Subsystems

The ITE subsystems are listed in the table below. Each subsystem is then discussed separately in more detail.

Table 5: ITE Subsystems

Type/Function	Item Name	Network Name	Description
Server	Dell PowerEdge R540	mxdM1	16 GB RAM, 3.80 GHz, 4 core, 500 GB SSD
IO Expansion	Pickering 40-922-001		8-Slot PXI Chassis
	NI PXI-6528		48-Channel, 24 Inputs, 24 Outputs
	NI PXI-6224		32 AI (16-Bit, 250 kS/s), 48 DIO
Server	NVIDIA Jetson AGX Xavier	mxdM2	Includes Intel I350T4V2 4-Port PCIe Ethernet Card
Server	NVIDIA Jetson AGX Xavier	mxdE1	Includes Intel I350T4V2 4-Port PCIe Ethernet Card
Server	Texas Instruments TMDSC3359	mxdE2	Industrial Communications Engine
Server	Texas Instruments TMDSC3359	mxdE3	Industrial Communications Engine
Server	Texas Instruments TMDSC3359	mxdE4	Industrial Communications Engine
Server	Raspberry Pi 4 with PoE Hat	mxdU1	Raspberry Pi 4 with PoE Hat
Server	Raspberry Pi 4 with PoE Hat	mxdU2	Raspberry Pi 4 with PoE Hat
Switch	Cisco IE-4010-4S24P	Main_Switch	Industrial network switch, 24-port



Switch	StarTech IES101G2SFPW	EF_Switch	Low cost, industrial network switch, 8-port
PDU	Eaton EMAU05-10	ePDU	16-port, managed PDU
Temp Sensor	Inveo PoE Temperature Sensor	mxdT1	Inveo PoE Temperature Sensor
Tray	E144A0444 - Emulator Tray		Emulator Tray

mxDM1 - Dell PowerEdge R540 Server

Detailed Description	Interfaces	Installation Location	Available I/O
The Dell R540 server features a single Silver Xeon 10c processor, 10GB Ram and a single 1TB SATA drive. It has dual 750W power supplies.	Interfaces include two 1GbE ethernet ports, two 10GbE ethernet ports, one full height PCIe slot and two half height PCIe slots.	The 2U server is rack mounted at U3 and U4. The connections at the rear of the server are accessible by removing the 5U rackmount panel at the rear of the ITE. With all connections removed the server can be slid out from the cabinet on the integrated slide rails.	Available interfaces include two 1GbE ethernet ports and two half height PCIe slots.

PXI Chassis – Pickering 40-922-001

Detailed Description	Interfaces	Installation Location	Available I/O
The Pickering PXI chassis is an expansion of the mxDM1 Dell server. The function is to increase the I/O of the server to a more rugged PXI/cPCI format. This PXI chassis features a 350W power supply and onboard thermal monitoring.	The chassis includes seven PXI/cPCI slots and one PXI Controller slot.	The 4U chassis takes up 5U of rack space with the rackmount bracket. It is mounted at the top of the cabinet at U20 through U24. Access to the PXI slots is through the front door of the ITE.	There are five available PXI slots on the chassis.

mxDM2 - NVIDIA Jetson with Intel I350-T4V2 Ethernet Card

Detailed Description	Interfaces	Installation Location	Available I/O
The NVIDIA Jetson server features a 512-Core NVIDIA Volta GPU, 8-Core ARM 64-	The server has one PCIe slot, one Gigabit ethernet port, and	The Jetson server is mounted to a DIN rail on a fixed shelf inside of the ITE at U13. The PCIe card is secured to the	Currently, there are three available



Bit CPU and 32GB Flash Storage. It is mounted to a custom DIN rail bracket with integrated mounting for the PCIe card.	one HDMI port. With the Intel ethernet card installed the I/O is expanded to include four additional Gigabit ethernet ports.	server using a custom bracket.	ethernet ports. All of these available ports are located on the PCIe card.
--	--	--------------------------------	--

mxdE1 - NVIDIA Jetson with Intel I350-T4V2 Ethernet Card

Detailed Description	Interfaces	Installation Location	Available I/O
The NVIDIA Jetson server features a 512-Core NVIDIA Volta GPU, 8-Core ARM 64-Bit CPU and 32GB Flash Storage. It is mounted to a custom DIN rail bracket with integrated mounting for the PCIe card.	The server has one PCIe slot, one Gigabit ethernet port, and one HDMI port. With the Intel ethernet card installed the I/O is expanded to include four Gigabit ethernet ports.	The Jetson server is mounted to a DIN rail on a fixed shelf inside of the ITE at U13. The PCIe card is secured to the server using a custom bracket.	Currently, there are three available ethernet ports. All of these available ports are located on the PCIe card.

mxdE2 - Texas Instruments AM3359 Industrial Communications Engine (TI-ICE)

Detailed Description	Interfaces	Installation Location	Available I/O
The TI-ICE features an AM3359 Arm Cortex-A8 processor, an installed 64 GB SD card for memory and is powered directly from a 24VDC power supply. It is mounted to a DIN rail using a DIN mount bracket. The power to the TI-ICE is fused to prevent the current from exceeding 0.75A.	The TI-ICE has two 10/100 ethernet ports, one DB9 connector and one micro-USB connector.	The TI-ICE is located on the Emulator Shelf at U6 inside the ITE. It is attached to the DIN mount on the slide out shelf for easy swapping and maintenance.	The DB9 and micro-USB connectors are available for use on the TI-ICE.

mxdE3 - Texas Instruments AM3359 Industrial Communications Engine (TI-ICE)

Detailed Description	Interfaces	Installation Location	Available I/O
----------------------	------------	-----------------------	---------------



The TI-ICE features an AM3359 Arm Cortex-A8 processor, an installed 64 GB SD card for memory and is powered directly from a 24VDC power supply. It is mounted to a DIN rail using a DIN mount bracket. The power to the TI-ICE is fused to prevent the current from exceeding 0.75A.	The TI-ICE has two 10/100 ethernet ports, one DB9 connector and one micro-USB connector.	The TI-ICE is located on the Emulator Shelf at U6 inside the ITE. It is attached to the DIN mount on the slide out shelf for easy swapping and maintenance.	The DB9 and micro-USB connectors are available for use on the TI-ICE.
--	--	---	---

mxdE4 - Texas Instruments AM3359 Industrial Communications Engine (TI-ICE)

Detailed Description	Interfaces	Installation Location	Available I/O
The TI-ICE features an AM3359 Arm Cortex-A8 processor, an installed 64 GB SD card for memory and is powered directly from a 24VDC power supply. It is mounted to a DIN rail using a DIN mount bracket. The power to the TI-ICE is fused to prevent the current from exceeding 0.75A.	The TI-ICE has two 10/100 ethernet ports, one DB9 connector and one micro-USB connector.	The TI-ICE is located on the Emulator Shelf at U6 inside the ITE. It is attached to the DIN mount on the slide out shelf for easy swapping and maintenance.	The DB9 and micro-USB connectors are available for use on the TI-ICE.

mxdU1 - Raspberry Pi 4 with PoE Hat

Detailed Description	Interfaces	Installation Location	Available I/O
The Raspberry Pi computer with installed PoE Hat features 8GB of Ram, an installed 64GB SD card and 1.5GHz processor. The Pi is mounted to a DIN mount bracket. Power is provided from the ethernet switch via PoE.	The Raspberry Pi offers one Gigabit ethernet port, two USB3 ports, two USB2 ports and two micro-HDMI ports.	The Raspberry Pi is located on the Emulator Shelf at U6 inside the ITE. It is attached to the DIN mount on the slide out shelf for easy swapping and maintenance.	The two USB3, two USB2 and micro-HDMI ports are available on the Raspberry Pi.



mxdu2 - Raspberry Pi 4 with PoE Hat

Detailed Description	Interfaces	Installation Location	Available I/O
The Raspberry Pi computer with installed PoE Hat features 8GB of Ram, an installed 64GB SD card and 1.5GHz processor. The Pi is mounted to a DIN mount bracket. Power is provided from the ethernet switch via PoE.	The Raspberry Pi offers one Gigabit ethernet port, two USB3 ports, two USB2 ports and two micro-HDMI ports.	The Raspberry Pi is located on the Emulator Shelf at U6 inside the ITE. It is attached to the DIN mount on the slide out shelf for easy swapping and maintenance.	The two USB3, two USB2 and micro-HDMI ports are available on the Raspberry Pi.

Main_Switch - Cisco IE-4010-4S24P Ethernet Switch

Detailed Description	Interfaces	Installation Location	Available I/O
The Cisco IE-4010-4S24P switch is a rack mounted, industrial grade switch with redundant power supplies. Power is hardwired into the switch via two cables coming from the ePDU.	The Cisco switch features twenty-four Gigabit PoE ports as well as four Gigabit SFP ports.	The switch is rack mounted at U12 facing towards the rear of the ITE. It can be accessed by removing one of the rearmost blanking panels.	The Cisco switch has eleven Gigabit PoE ports and three Gigabit SFP ports available.

EF_switch – Startech IES101G2SFPW

Detailed Description	Interfaces	Installation Location	Available I/O
The Startech IES101G2SFPW is a small, industrial grade Gigabit ethernet switch.	The switch features eight Gigabit ethernet ports and two SFP ports.	The switch is mounted on the tray that also houses the NVIDIA Jetsons which is located at U13.	The switch has four available ethernet ports and one available SFP port.

ePDU - Eaton EMAU05-10

Detailed Description	Interfaces	Installation Location	Available I/O
The Eaton ePDU is a 2U, rack mounted, programmable and	The ePDU features 16 NEMA 5-15R/20R	The ePDU is mounted at U1-U2 and faces the rear of the cabinet. The bottom, rear	There are five available NEMA 5-



remote controllable PDU. It is powered by 120VAC and is capable of up to 2.88 kW output. Each outlet can be monitored for power consumption and can be independently powered on or off.	receptacles for AC power and can be controlled through an ethernet 10/100 port or RS-232 port.	blanking panel must be removed in order to access.	15R/20R receptacles.
---	--	--	----------------------

mxdT1 –Inveo PoE Temperature Sensor

Detailed Description	Interfaces	Installation Location	Available I/O
The Inveo Nano Temperature Sensor PoE is a small IP thermometer with a single probe for temperature monitoring within the ITE.	The temperature sensor has a single PoE ethernet input and a single temperature probe input.	The ePDU is mounted to a DIN mount plate and installed on the fixed shelf at U13.	There are no available ports on the temperature sensor.

E144A0444 - Emulator Tray

Detailed Description	Interfaces	Installation Location	Available I/O
The slide out emulator tray is a modular approach to housing smaller computers for emulation. The tray slides out so maintenance can be performed without having to remove any of the components. Cables are routed in via 2 flex chains separating power and signal.	The emulator tray currently houses two Raspberry Pis, three TI-ICE computers and a 24 VDC Power supply.	The tray is located at U6 inside of the cabinet.	All components are DIN mounted and currently take up approximately 75% of the available DIN rail length.

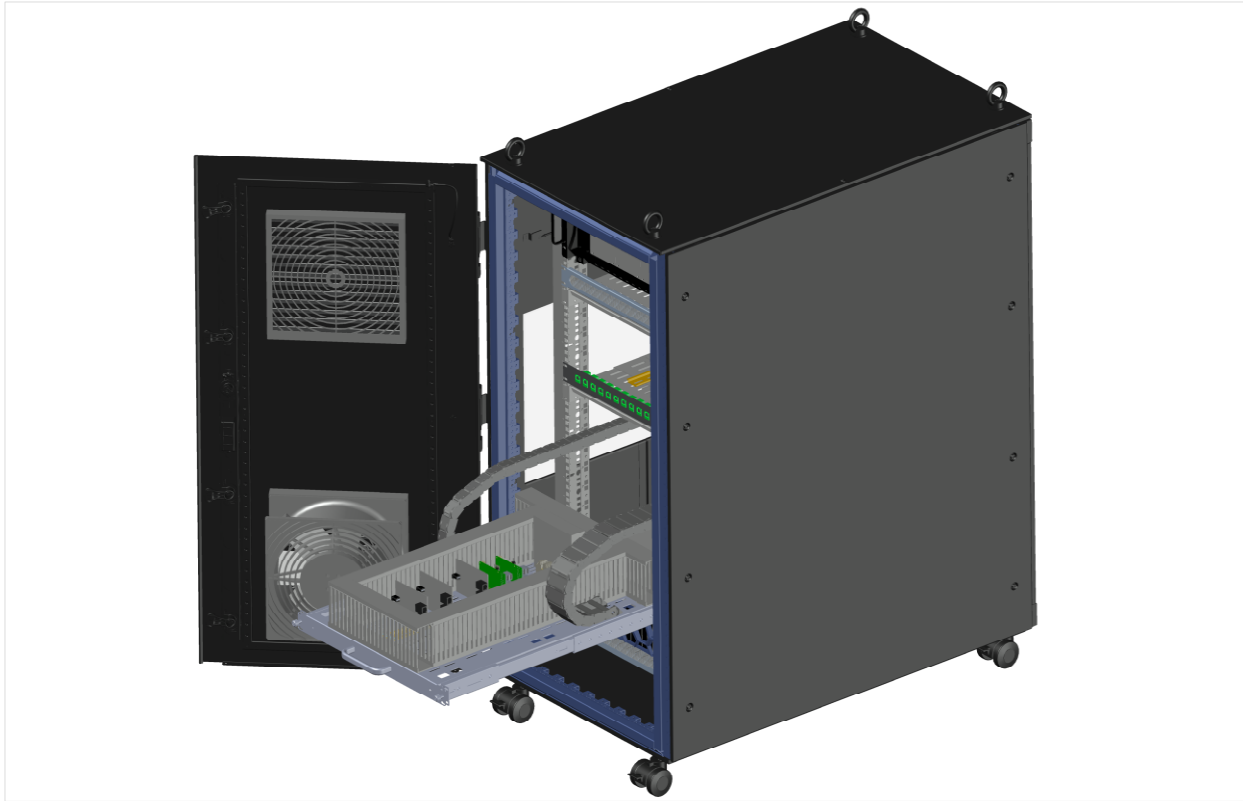


Figure 11: Rendering of the ITE Rig with emulator tray extended

ITE Expansion

The ITE was built with expansion in mind. Here is a table identifying the available expansion capacity:

Table 6: ITE planned expansion capabilities

Name	Item	# of Unused Ports	# of Ports in Use	Port Type	% Expansion
	PXI Chassis	5	3	PXI Slot	167%
Main_Switch	Cisco IE 4010 Switch	14	10	RJ45	140%
EF_Switch	Startech Switch	2	8	RJ45	25%
mxdM1	Dell R540 Server	4	1	PCIe Slot	400%
		1	3	RJ45	33%
mxdM2	Nvidia Jetson	3	2	RJ45	150%
mxdUS1	TI-ICE	1	1	RJ45	100%
mxdE1	Nvidia Jetson	3	2	RJ45	150%
IO_Panel_1	MIL-spec RJ45 Panel	2	5	RJ45	40%
IO_Panel_2	Harting Modular Panel	14	10	Harting Modules	140%

Industrial Open Automation Framework Design

Digital Twin and Open Automation Framework

This section describes all the functional units that are defined for the DT-OA Framework. A functional unit can be a process, set of enhancements to be made to an existing process, an API, or an interface between modules such as a data structure or file format.

The Digital Twin and Open Automation Framework combine multiple applications executing on the DT-OA servers and client system. Data from the emulated factory are delivered to the multiple digital twins and ITE CBIT executing on the MxdM1 server by the OPC-UA and ModbusTCP clients. That data is then used to drive a pump health digital twin.

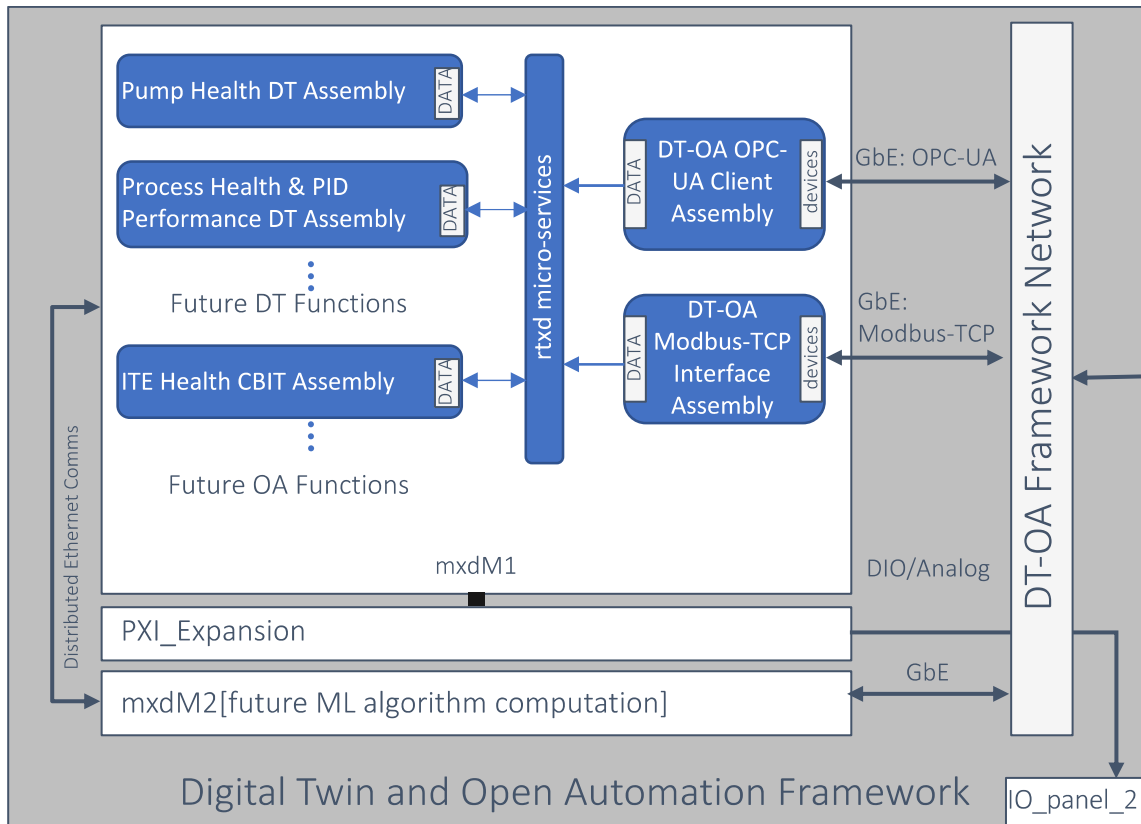


Figure 12: Digital Twin and Open Automation Framework

Pump Health Digital Twin Assembly

The pump health digital twin is composed of a Matlab/Simulink model monitoring pump vibration determining health based on collected healthy vibration data. The DT determines the device's performance index, DT confidence interval, and identifies anomalous behavior. The DT executes every second. The data source for the assembly is the emulated framework transferred via ModbusTCP from the pump ModbusTCP server. The DT health metrics are summarized on the DT-OA panel.

Process Health Digital Twin & PID Performance Assembly

The flow-control process digital twin is composed of a Matlab/Simulink model monitoring process and control data. The data sources for the assembly are part of the emulated framework with data transferred via ModbusTCP from the control valve ModbusTCP servers and via the plant OPC-UA server. The DT executes every 0.1 seconds. The DT determines the

process health and identifies probable causes (process-related or controller-related) based on learning from normal operations data and logic comparing metrics for requested flow, controller command, and response.

ITE Health CBIT Assembly

The ITE Health CBIT assembly executes a health monitor of the ITE rig, cycling at a deterministic frequency of 100 Hz. The assembly monitors the following metrics and compares them to healthy ranges:

- Temperature collected at two points in the rig cabinet
- Jitter of each DT-OA assembly (i.e. determinism of each assembly's execution time)
- Any execution overruns of each assembly

The ePDU, a programmable power distribution unit, measures temperature and electric load of each unit connected to each of its outlets. Currently the electric loads are monitored and displayed in the DT-OA panel. The assembly could be programmed to remove power from any outlet where the load is judged to be excessive via the ePDU. A second and separate thermal device provides temperature measurement at another section of the cabinet.

The execution jitter and overruns are measured via the rtxd microservices.

The overall health of the ITE system is determined with parameterized weights for each metric. The ITE health is displayed in the DT-OA panel. ITE health is indicated with text, color and a percentage of health as shown in Figure 13. The weights are configurable via an ADEPT setup file. The assembly is programmed in Matlab/Simulink and can be easily customized for additional metric checking and actions deemed useful to protect system health.

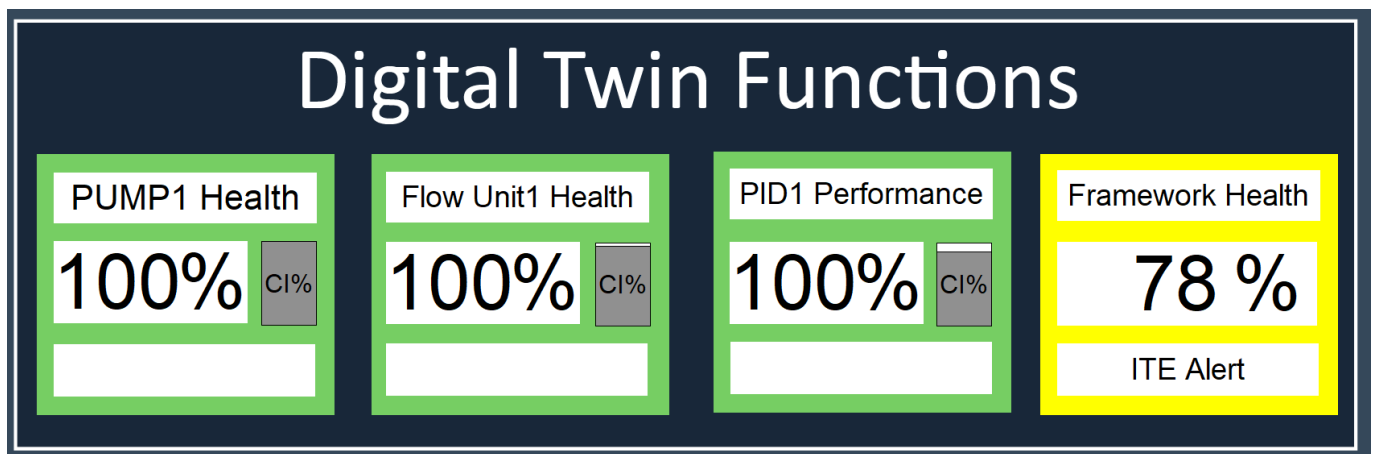


Figure 13: ITE Health Monitor

DT-OA OPC-UA Client Assembly

This assembly receives the incoming OPC-UA emulator data and makes it available for consumption by the digital twin and open process automation assemblies. The OPC-UA client receives new data at a rate of 100 Hz, and this data is immediately available to other assemblies via model-to-model inter-process communications.

DT-OA ModbusTCP Interface Assembly

This assembly requests and receives the ModbusTCP emulator data and makes it available for consumption by the digital twin and open process automation assemblies. The ModbusTCP

interface requests new data at a rate of 50 Hz, and this data is immediately available to other assemblies via model-to-model inter-process communications.

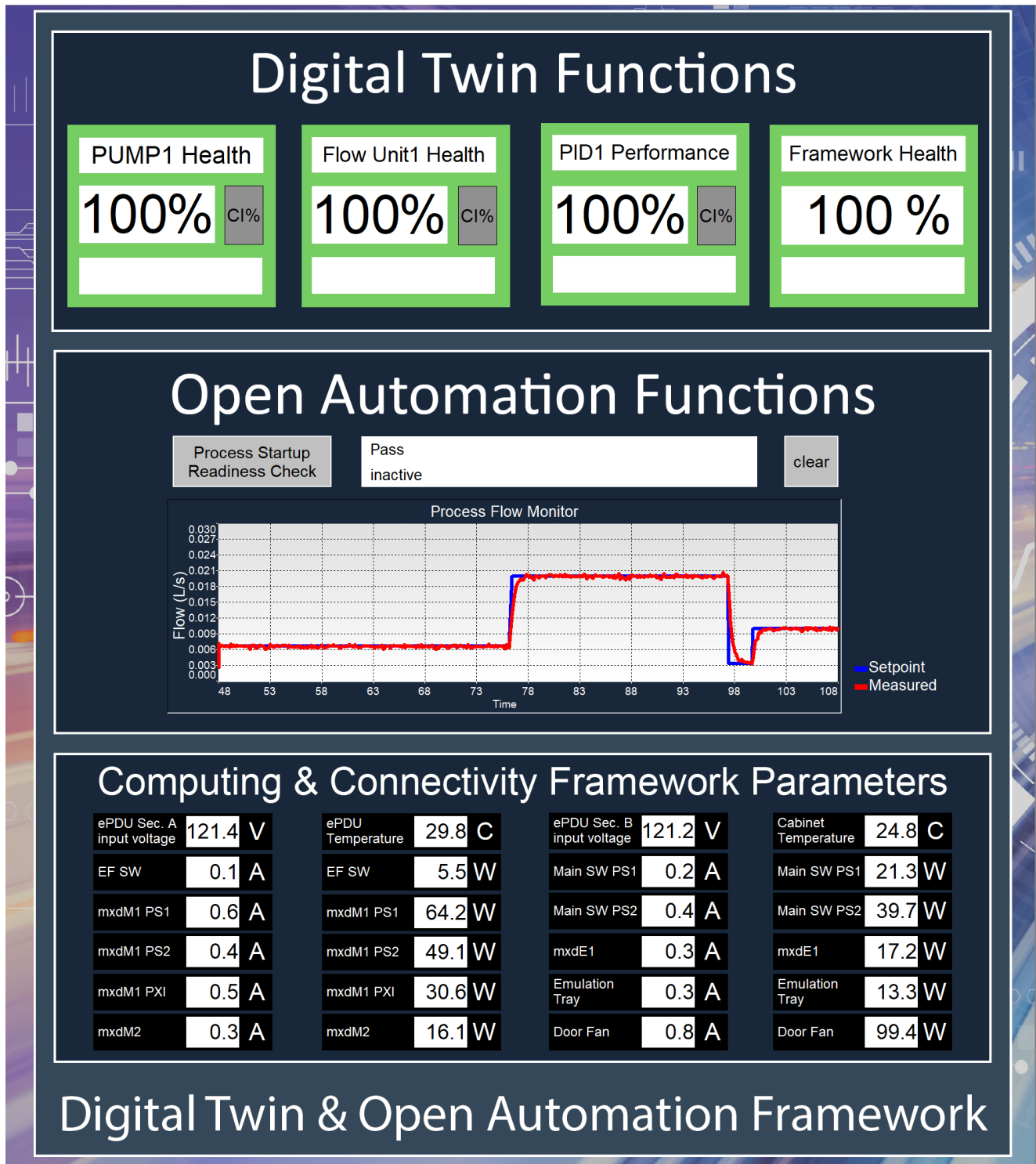


Figure 14: DT-OA Interface Panel



Digital Twin and Open Automation Panel

The Digital Twin and Open Automation user interface panel is shown in Figure 14. Digital Twins (DTs) provide health status information for the pump, flow-control process, and controller, while a Continuous Built-In-Test assembly provides health status of the ITE rig. The health status for these items is summarized in the four-box structure shown at the top of the panel. The color of the boxes indicates system, rig, or equipment health, and each is based on the computed performance index; green indicates healthy, yellow indicates the item has some degradation, and red shows the item is faulted. The performance index is a percentage and is shown in the middle section of the items' status box. The status figures for the pump, flow unit, and controller display the confidence interval of the DT's assessment with a vertical meter. The confidence interval ranges from 0% shown with an empty meter up to 100% when the meter is fully colored grey. The bottom of each status box displays messages to clarify the degradation type or anomaly. There is one pump message, identifying the degradation as a Bearing Wear Alert. The flow-unit messages identify a Control Anomaly and a Process Anomaly. The controller or PID performance messages identify Excess Control, Controller Railed, and Not Compensating.

Just below the "Open Automation Functions" heading is a two-line informational readout surrounded by two buttons on the left and right. This collection of widgets allows for the control and monitor of the startup readiness check open process automation function. This function will check that each of the following variables falls within a predefined tolerance:

Table 7: ITE Framework Health Open Automation Components

Variable	Description
OPC_SVR_Status	Status of the OPC-UA server
MES_Status	Status of the Manufacturing Execution System
DH_Status	Status of the Data Historian
NET_HLTH_ITE	Network health of the ITE
NET_HLTH_Factory	Network health of the factory
BMS_AC_MAIN_V	Voltage reading from the Building Management System
BMS_HVAC_Tout	Output temperature of the HVAC system from the Building Management System

When the button on the left is pressed, the bottom line of the display will read "active" throughout the duration of the readiness check, a few seconds. Once the check is complete, the bottom readout will display "inactive" and the top readout will display "Pass" or "Fail: <failure>", where <failure> defines which part of the check failed. It should be noted that only the first failure will be displayed. During this check, the status of each variable is logged, and a log file is saved to the SVN server on mxdu2. The "clear" button on the right can then be pressed to reset the readouts and perform the check again. This startup readiness check, along with the variables listed above, acts as a sort of placeholder to show how future open process automation functions could be developed.

The flow-unit DT monitors the flow demand and measured flow response; those quantities are displayed in the strip-chart in the middle of the panel.

The ITE Continuous Built-in Test assembly monitors the ITE framework thermal and electrical load metrics. The metrics are displayed in the lower third of the panel. The physical rig parameters, including the ePDU input voltage (two sections), the power and current load of each device plugged into the ePDU, and the temperature in two different locations inside the rig. This information, except for the "Cabinet Temperature", comes from the ePDU logical device, which constantly polls the ePDU for the information via an RS-232 serial connection. The

“Cabinet Temperature” data comes from a dedicated temperature monitoring device that communicates via ModbusTCP and updates at a rate of 1 Hz.

Emulator Framework

The Emulator Framework contains a distributed simulation which runs on four separate servers, connected via an ethernet connection and communicating over UDP. The distributed simulation emulates a process Distributed Control System (DCS) having a process simulation outputting information via an OPC-UA server along with pump and variable control valve assemblies outputting data to the DT-OA framework via ModbusTCP. One server, mxdE1, runs the process model, a pump vibration model, and the OPC-UA server. The OPC-UA server, along with the other three servers – mxdE2, mxdE3, and mxdE4 – transmit the process and noise data across the network to the Digital Twin and Open Automation framework. Each of mxdE2, mxdE3, and mxdE4 act as a ModbusTCP server, transmitting data across the network.

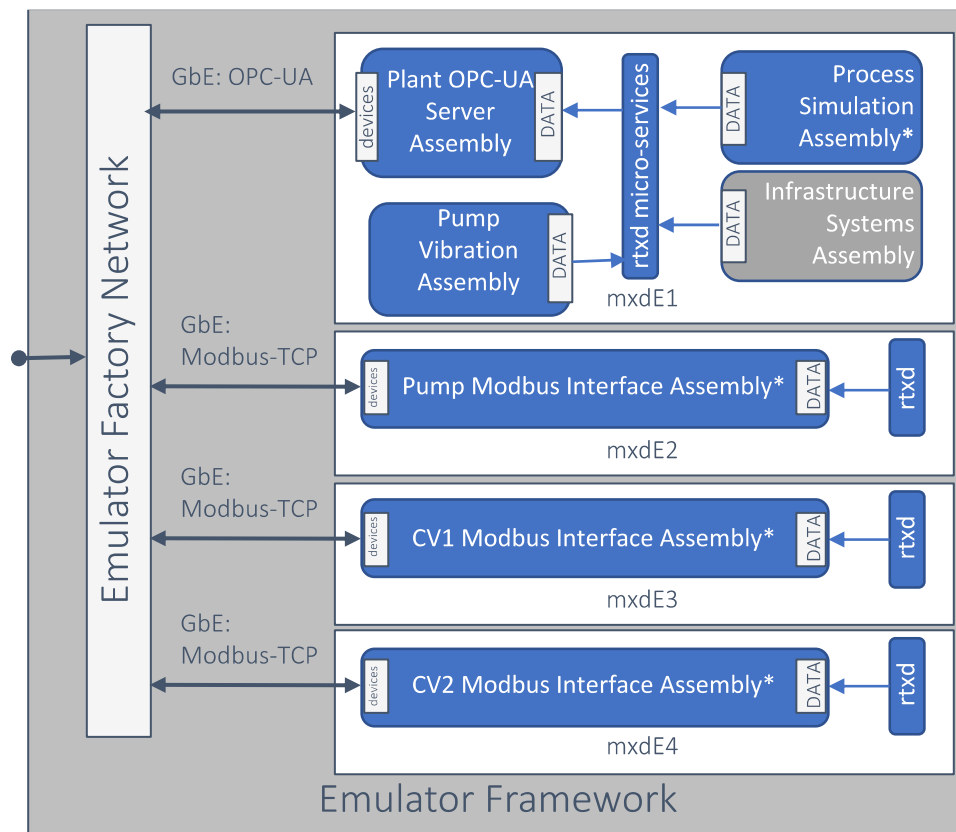


Figure 15: Emulator Framework

Process Simulation Assembly

The process simulation consists of a physics-based model developed in MATLAB/Simulink that is running on mxdE1, one of the Nvidia rapid prototyping computers. The model includes a thermal-liquid mixing process implemented with Simscape. Features include representative noise characteristics for the flow sensor and a PI controller with setpoint weighting.

Sensor noise of the forward flow measurement is represented by adding noise with a signal to noise ratio of 40 dB to the forward flow measurement. The signal power is considered a random variable uniformly distributed between zero flow and the maximum flow rate.

Figure 16 shows a P&ID drawing of the process simulation components.

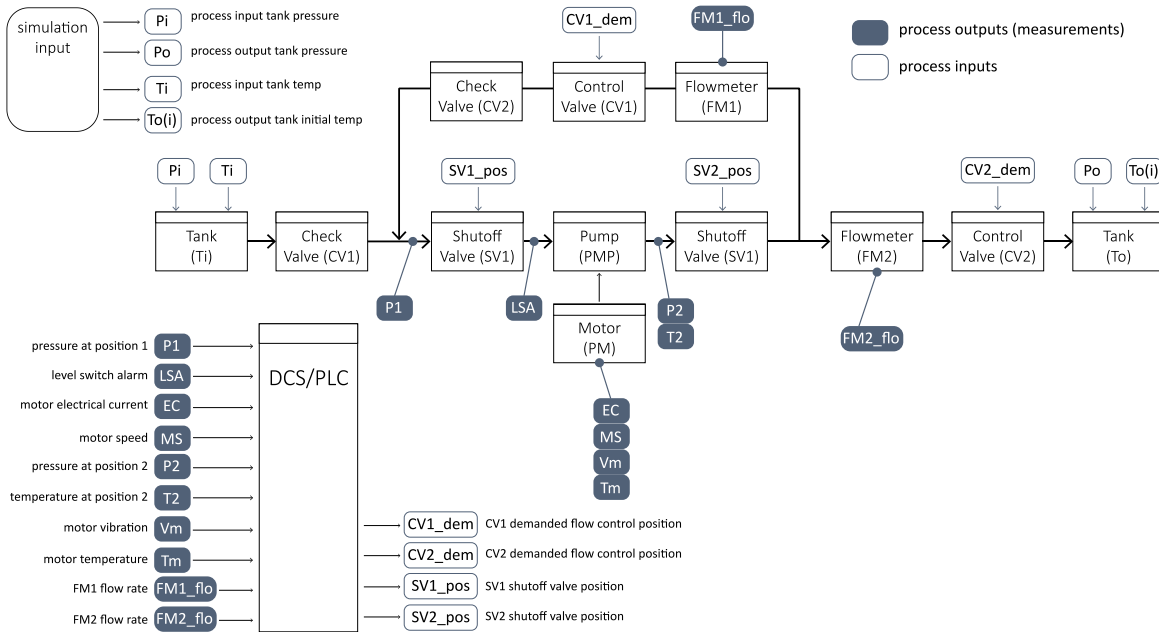


Figure 16: P&ID drawing of process simulation components

Pump Vibration Simulation Assembly

Pump vibration is simulated with a separate C-model assembly. See Appendix 1 for more information. Nominal vibration noise is generated for representation of a healthy pump. A degrading pump is represented with a profile representing vibration progression from a healthy through degradation to a fully faulted state. A four-parameter vibration profile is loaded as a binary file. The C-model can playback the profile at a requested rate and at a requested start point. Normally a pump degradation may occur over a long period of time; however, for demonstration purposes the degradation can be played back over seconds or started well into the degradation process.

Plant OPC-UA Server Assembly

The Plant OPC-UA Server assembly, which runs on mxdE1, packages plant model data into OPC-UA protocol packets to send across the ITE network to the DT-OA framework. The plant model data is generated by the Process Simulation assembly and is then transmitted via model-to-model inter-process communications – since the models run on the same physical server – to the Plant OPC-UA Server assembly.

Pump ModbusTCP Interface Assembly

The Pump Modbus Interface assembly, which runs on mxdE2, packages the pump plant model data into ModbusTCP protocol packets to send across the ITE network to the DT-OA framework. The plant model data is generated by the Process Simulation assembly and is then transmitted via network inter-process communications – since the models run on different physical servers – to the Pump Modbus Interface assembly.

Control Valve 1 ModbusTCP Interface Assembly

The CV1 Modbus Interface assembly, which runs on mxdE3, packages the control valve 1 plant model data into ModbusTCP protocol packets to send across the ITE network to the DT-OA framework. The plant model data is generated by the Process Simulation assembly and is then

transmitted via network inter-process communications – since the models run on different physical servers – to the Pump Modbus Interface assembly.

Control Valve 2 ModbusTCP Interface Assembly

The CV2 Modbus Interface assembly, which runs on mxdE4, packages the control valve 2 plant model data into ModbusTCP protocol packets to send across the ITE network to the DT-OA framework. The plant model data is generated by the Process Simulation assembly and is then transmitted via network inter-process communications – since the models run on different physical servers – to the Pump Modbus Interface assembly.

Normal Operations

Normal operations are defined for use in DT training and verification. A typical operation cycle begins from a stopped state, where the “stopped” flow rate is normally 10% of the maximum flow rate. The cycle contains two set points where the first setpoint is normally less than the second setpoint and the duration of the first setpoint is longer than the second setpoint duration. The process cycle may represent a mixing procedure followed by a flushing procedure. Figure 17 displays the cycle’s progression from a stopped state to a ready-to-start state, a starting state and then the production state.

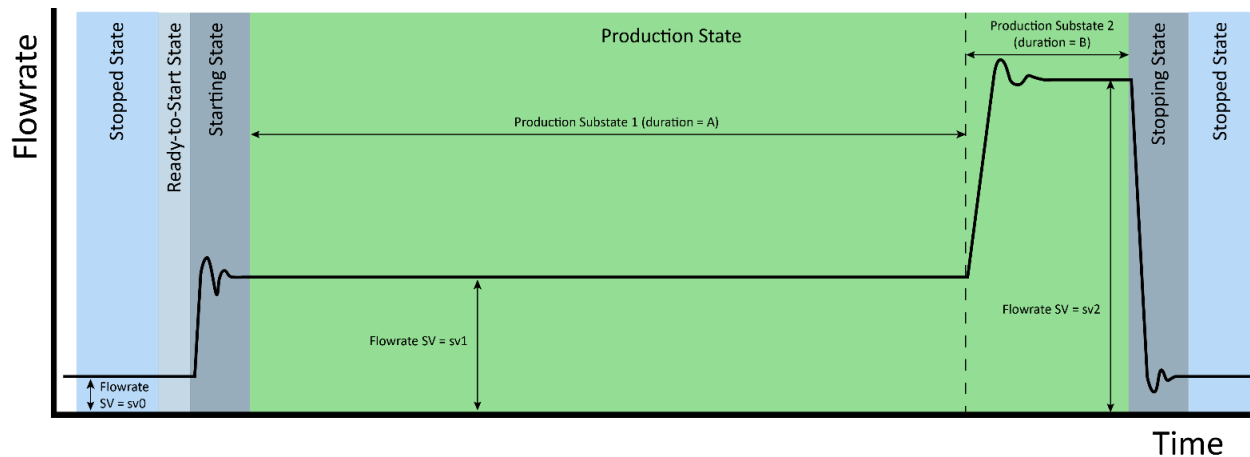


Figure 17: Standard Operation Cycle

Python scripting executed from ADEPT-VI implements an automated set of normal operations and a single-shot normal operation. The script controls the process operation via the process assembly’s data interface, setting the flowrate setpoints and durations and monitoring key performance indicators are within tolerance. The script controls and monitors the standard operation cycle sequence. Normally the durations would be on the order of hours; however, for demonstration purposes the setpoint durations are set to under a minute.

A maintenance state is supported by the process assembly. The typical maintenance sequence is setting the flow setpoint to stopped, requesting the pump stop via the pump controller, and then closing the shutoff valves. The maintenance state is exited by opening the shutoff valves, requesting the pump startup, and then remaining in the stopped state until an operation is requested.

The normal operation states and maintenance state are controlled and indicated with an operation state enumeration.

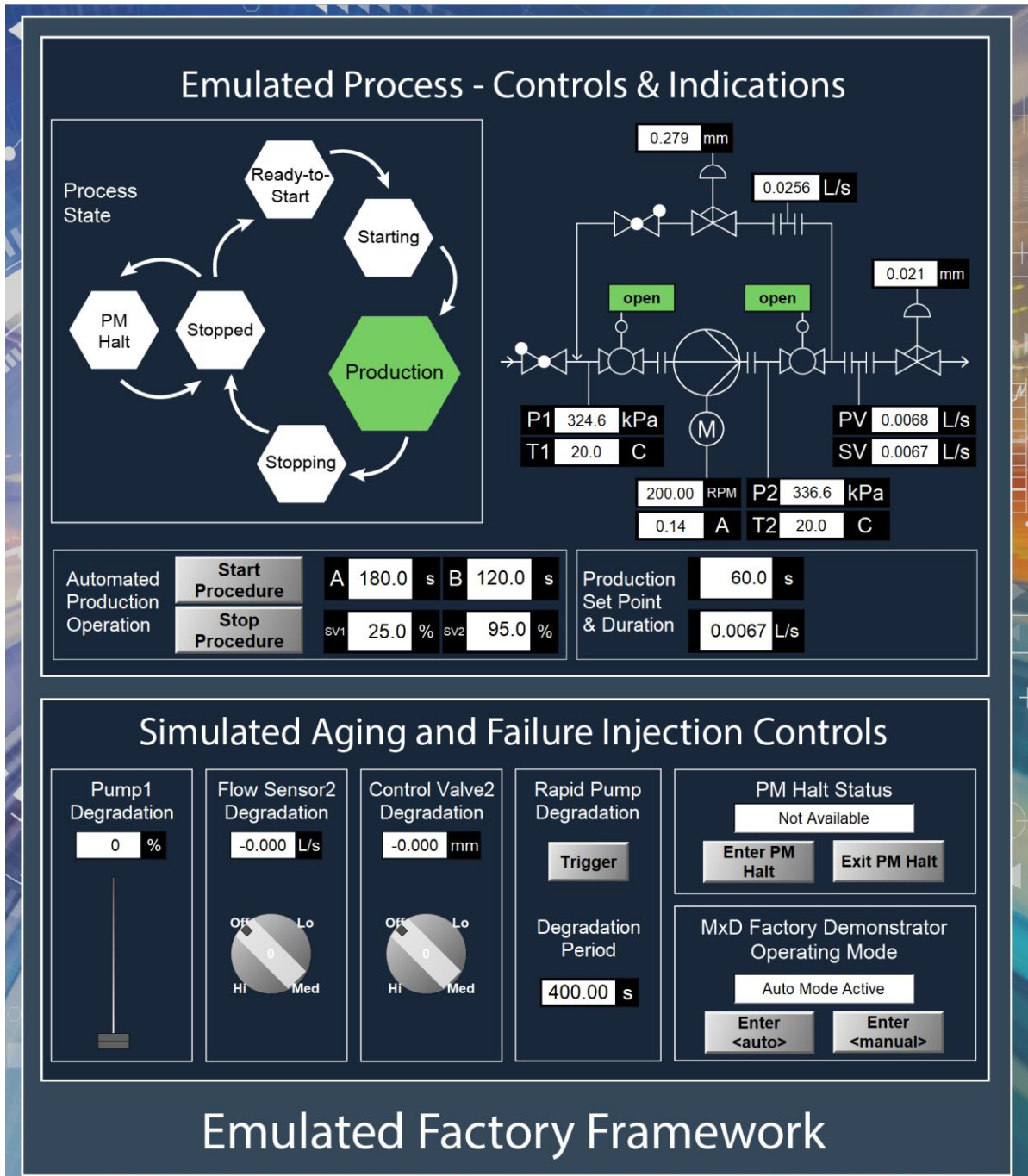


Figure 18: Emulator Framework Panel

At the top of the emulated factory panel is an indication of the production process state. To the right of the production state visual is a piping and instrumentation diagram (P&ID) that shows the basic process components, a pump, two variable control valves, two shutoff valves and several temperature and pressure sensors, along with readouts displaying the current thermal, flow, and valve conditions.

There are essentially two modes of defining and running a process cycle: an automated mode where the process cycles are defined and sequenced with a python script running from ADEPT-VI and a manual mode where a single process cycle can be defined and executed. In the



bottom right corner of the emulator panel the mode of operation is displayed and there are buttons to select the alternate mode. Above the mode selection section there are two buttons and an availability status for requesting maintenance. Manual mode must be active to allow the process to be put into a maintenance state.

In the midsection of the panel at the left identified with the text “Automated Production Operation” an operator can enter two setpoint values and the duration in seconds of each setpoint in the data-entry windows. The single production cycle can be initiated with the Start Procedure button and stopped with the Stop Procedure button. As noted above, a single production cycle can only be activated when the Operating Mode is set to manual. To the right of the setpoint and duration data-entry windows are readouts of the current setpoint and duration being processed, identified with the text “Production Setpoint & Duration”.

Near the bottom of the panel are controls for injecting simulated degradation into the pump and flow-control process. The overall pump degradation period may be set in the data-entry window identified with the text, “Degradation Period.” The Pump1 Degradation slider may be used to select how far along the degradation should be started within the period. The Trigger button begins the play-back of the degradation vibration. The toggle button displays the text “Active” when the degradation is active. To bring the vibration to a healthy state toggle the Trigger button. The Digital Twin monitoring the pump does not set the pump’s status to a healthy state until the pump vibration is healthy, and the maintenance state has been executed and cleared.

The two rotary knobs may be used to inject biases into the flow control valve or flowrate sensor to emulate process and control degradation and various anomalies.

Digital Twin Design

Executive summary

Overview Summary of the Digital Twin Solution

The objective of this part of the project is to deliver a DT solution that uses open process automation concepts to showcase functionalities that allow an organization to more quickly evaluate operations, test assumptions for innovation, and improve capabilities.

This part of the project focuses on the development of an extensible DT solution to support equipment and process condition monitoring for process manufacturing. The major deliverables include:

A DT class that monitors the condition of a pump (equipment) in the system by detecting degradation onset, estimating health state percentage, and classifying the health state into three states: Normal, Degrading, and Degraded.

A DT class that monitors the process health. This DT estimates the health percent and detects anomalies in the process by monitoring if the process set-points are tracked or not and by using information from the third DT class that monitors the PID controller performance. Two types of process anomalies can currently be detected. The first type occurs when the PID controller is applying excessive control in order to meet the set point. The second type occurs when the set point is not tracked and the PID controller hits its maximum.

A DT class that monitors the PID controller performance. This DT provides an indication of the health state of the PID controller. A controller anomaly is detected if the process set point is not tracked and the controller has not yet hit its maximum compensation value. The output of the PID controller performance DT is used by the process DT to help detect process anomalies.

The current DT classes use univariate modeling as a baseline for equipment, process, and controller health monitoring. Next generation DTs will use more complex dynamics in

multivariate models (if necessary) to aid in the assessment and provide additional output metrics if desired.

Design Constraints

The proposed DT solution will utilize the data gathered by the Integration Test Environment (ITE) to deliver a passive/monitoring real-time DT capability for understanding performance of equipment and process, i.e., health monitoring, performance degradation, predictive failure, and anomaly detection. The proposed DT solution provides the above capability in a value-add fashion in such a way that it does not interfere with live manufacturing.

The current design focuses on a single component of the system (pump) and a single control loop process (oil loop). The DT solution can be easily scaled using DT object-oriented concepts of reusability, interoperability, interchangeability, and extensibility to meet the minimal amount of additional resources the proposed DT solution will require.

Section Organization

This section completely describes the proposed DT solution, including its sub-components and their services, data management, interface mapping, global structure, and boundary conditions. This section is organized into five major subsections. Charts, tables, and graphics have been inserted to explain or clarify content.

DT Framework Architecture and Architecture Design

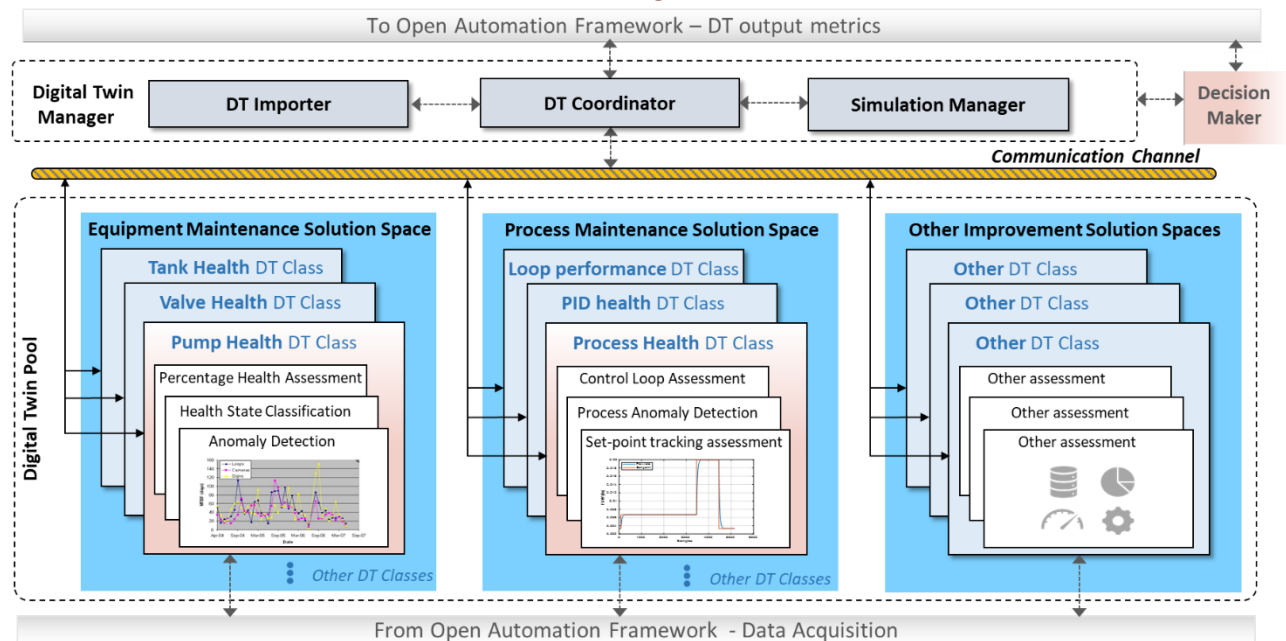


Figure 19: Digital Twin Framework

The DT framework provides the capability to construct aspects of a global view of the process manufacturing system that could be used for multiple improvement purposes. It provides the ability to represent and model the process manufacturing system at multiple levels of the ISA-95. The framework uses object-oriented concepts such as class, object, instance, aggregation, generalization/specialization, etc. to comply with the requirements of re-usability, interoperability, interchangeability, extensibility, and scalability across the entire DT lifecycle.

The foundation of the DT framework is the concept of a DT class, which is defined as a type of DT that delivers a specific capability for the DT client. The scope of a DT class is defined by the

DT output purpose metric, thus all DTs belonging to the same class must share at least one common output metric. However, DTs belonging to the same DT class might utilize different numbers and types of modeling approaches and might report additional output purpose metrics. DTs belonging to the same class might utilize and report different quality-of-output metrics. A quality-of-output metric is an indication that quantifies the quality or believability of the DT output.

The architecture of the DT framework is shown in Figure 19. The core component of the DT framework is the DT pool where different DT classes can be developed for multiple improvement purposes. For example, in the equipment maintenance space, one can define and develop DT classes for predictive maintenance for each type of equipment in the system (e.g., pumps, valves, etc.). A DT class may consist of other DT classes. For example, a pump health monitoring DT class may combine a predictive failure DT class, an anomaly detection DT class, and a remaining useful life estimation DT class. In the process maintenance space, several DT classes could be defined and developed for process monitoring such as a process anomaly detection DT class and a control loop performance monitoring DT class. Similarly, several other DT classes could be defined and developed within the DT pool depending on the intended improvement purposes.

The aggregation and generalization concepts allow for the realization of DT hierarchies that describe how the DT classes or instances within the DT solution can be reused, combined, shared, or otherwise aggregated or inherited (see, Moyne et al. 2020, Qamsane et al. 2021 for more details). An example DT hierarchy for Part A process manufacturing system will be discussed later in this document.

Generalization: a method to extrapolate from generic to specific capabilities. Generalization allows us to reuse and specialize developed DT capabilities.

Aggregation: a method to combine DTs to extend capabilities within a DT solution framework

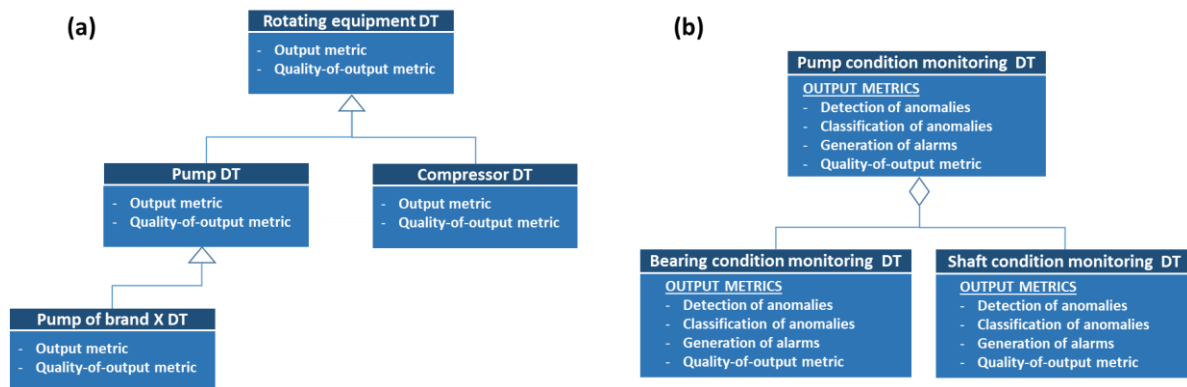


Figure 20: Illustration of the generalization and aggregation hierarchy concepts. (a) Example of a generalization hierarchy for a rotating equipment DT class supporting instantiation at multiple levels. (b) Example of an aggregation hierarchy for a pump condition

The DT framework allows the different DT class instances of the system to effectively communicate and share functionality to achieve a global objective. A communication channel allows this information sharing and a DT manager handles the initialization of the DT pool, coordinates the communication amongst the DT instances, and ensures that the DT outputs are delivered to the DT client (e.g., apps).

The DT Framework currently provides the following major components:

- Pump Health DT



- Process Health DT
- PID Controller performance monitoring DT

These major components are discussed in detail in the next subsection.

Detailed Design

This section details the steps for the design and development of the three components of the proposed DT solution, namely the pump health DT, process health DT, and PID controller performance monitoring DT. The first subsection provides a generic approach for DT design and development, then the subsequent subsections will discuss each of the three components of the DT solution.

Generic DT design and development approach

The proposed DT design and development plan is arranged around two major steps, described below.

Defining the specifications and constraints

Various specifications and constraints must be accounted for during the DT design process. For effective DT solutions, the customer specifications (e.g., throughput, product variety) and system constraints (e.g., available equipment, maximum startup time), will be encoded as formal logic statements and data structures. Similar encoding will be developed for existing standards and policies obtained from different government and industrial organizations (e.g., ISA 95, ISA 88).

Digital Twin creation and configuration

As described in Section V, a DT is a dynamic purpose-driven digital representation of an asset, system, or process. To achieve their purposes, DTs use various resources (i.e., data, models, software algorithms, applications, etc.) to update their state and output metrics. These output metrics are used to detect, predict, or prescribe some aspect of the system performance. This subsection describes how to identify, create, and configure a DT solution (a hierarchy of connected DTs) that will be used to test, analyze, or evaluate various improvement solutions through a selection of key performance indicators.

A high-level overview of the approach used to design and develop a DT solution is shown in Figure 21. More details can be found in (Qamsane et al. 2021). The DT solution design and development approach consists of the following stages:

Stage 1: Planning

The objective of the planning stage is to determine if there is a need to enhance some aspect of the manufacturing ecosystem and if that need could be addressed through the application of a DT solution. A set of high-level assumptions and constraints is identified for the manufacturing need and a DT solution is determined. The DT solution will consist of a set of concepts and methods that identifies the contribution of the DT in terms of what it will provide to address the manufacturing need. The DT contribution consists of producing quantifiable output metrics (key performance indicators) that can be used to address the manufacturing need as well as quality-of-output metrics that quantify the quality or believability of these DT output metrics.

Stage 2: Requirements and Analysis

The requirements and analysis stage analyzes the requirements for the DT design and development activities. The DT solution should be analyzed from qualitative (initially) and (then) quantitative DT lifecycle perspectives. A qualitative analysis investigates the feasibility of the DT solution based on Subject Matter Expert (SME) observations and understanding of resources,

as well as data quality requirements (e.g., availability, accessibility, volume, velocity, and veracity). The quantitative analysis studies feasible solution alternatives in terms of technical capability and performance measures using mathematical calculations and statistical analysis (see, Qamsane et al. 2021 for more details)

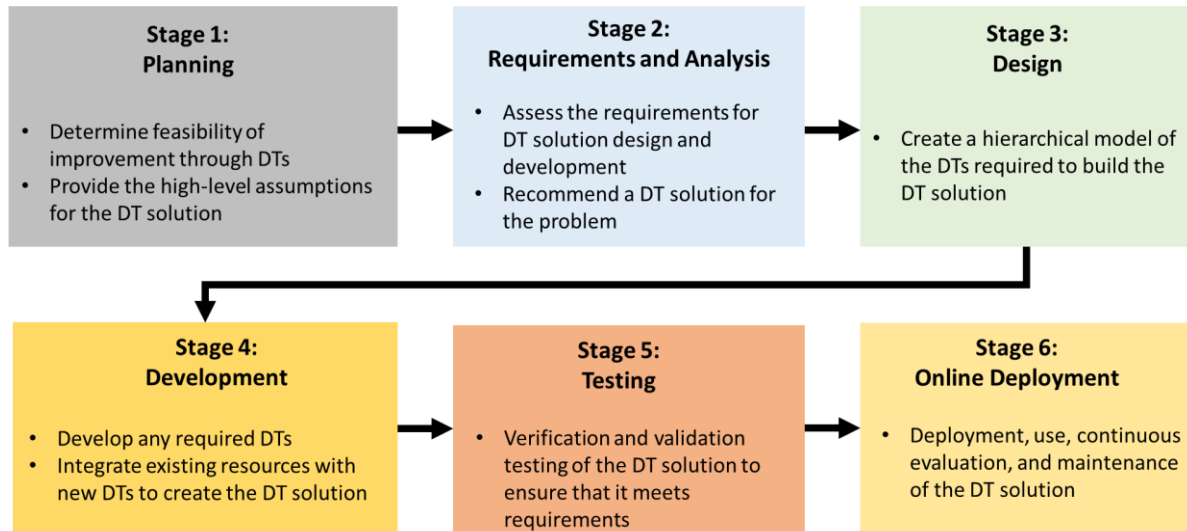


Figure 21: Overview of the DT design and development approach (Qamsane et al. 2021)

Stage 3: Design

The design stage defines a design of a DT solution as part of a DT framework based on the functional, data, and interaction requirements. The requirements are mapped into a hierarchy model that defines the DT classes, their attributes, operation, methods, and relationships. The DT hierarchy model also defines how DT resources for the overall DT solution are combined and shared, and how they can be extended and reused.

Stage 4: Development

The objective of the development stage is to transform the output design stage into a complete working DT solution capable of addressing the manufacturing need established in the planning stage. The development stage consists of activities for building DT classes and models. DT model development is the main piece of the development stage. This stage consists of formulating the DT models which could include selecting algorithms to use, writing the equations linking system response to inputs and values of system parameters, performing variable down-selection, model training, and general evaluation, etc. The DT solution is then developed by generating software programs that capture the algorithms, equations, SME input, and integrating all the DT components together.

Stage 5: Testing

In this stage, the developed DT solution is tested against the requirements to make sure that it actually meets specifications and fulfills its intended purpose of addressing the manufacturing need identified in the planning stage. The DT model is evaluated under several conditions, observing its behavior, and inspecting it for defects by comparing its results to experimental evidence. Typically, historical data is used here for sensitivity analysis to evaluate the effects of changes in inputs and in physical system changes on outputs

Stage 6: Online Deployment



In the on-line deployment stage, the qualified off-line DT solution is deployed, used, continuously evaluated, and maintained. The deployment includes the integration of the DT solution into the system where it uses run-time data from its operation environment to assess the system and make recommendations. Continuous maintenance and evaluation are required for continued effective use of the DT solution. Maintenance includes determining if and when the DT solution should be updated. Updates could be implemented on-line (e.g., model tuning) or might require the DT solution to be taken off-line (e.g., model rebuilding).

Pump health monitoring DT

Concept

The pump health DT uses vibration data generated by the process simulation assembly to detect pump degradation onset, classify its health state, and provide a percent indication on the overall health. Figure 22 shows the flowchart of the pump DT. The vibration data is delivered to the pump DT in the form of a frequency spectrum that consists of 100 frequency bins of 100 Hz spectral width, representing an underlying sample rate of 20 kHz. The vibration signal's Root Mean Square (RMS) is used by the pump DT as a condition indicator for health monitoring. The vibration RMS, equivalent to the standard deviation σ_a of the accelerometer signal $\mathbf{a}$, is recovered from the 100 bins frequency $\mathbf{S}_n$ as follows:

$$\sigma_a = \sqrt{\Delta f \sum_{n=1}^N S_n}.$$

where Δf is the frequency bin width (often called the frequency resolution)

$$\Delta f = \frac{1}{2N\Delta t}.$$

The pump health DT consists of a DT model that uses vibration data to compute and provide information to a decision maker that uses these model outputs to deliver the DT output and quality-of-output metrics.

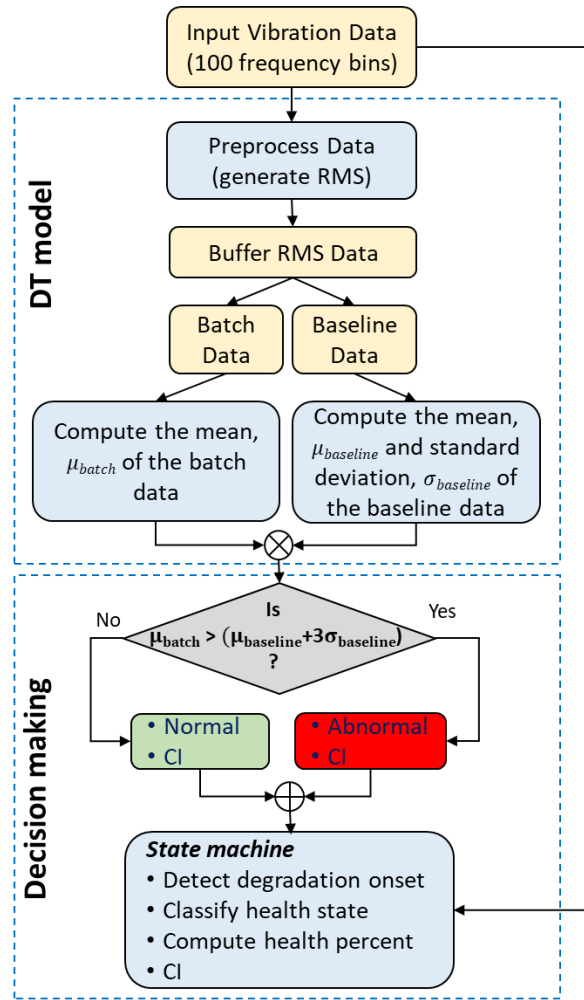


Figure 22: Flowchart of the pump DT

DT model

The DT model is based on a moving average concept that consists of the following: a batch window W of n health indicator (RMS) values is set up and the mean μ_{batch} of these n values in the window W is calculated. At the same time, the mean $\mu_{baseline}$ and the standard deviation $\sigma_{baseline}$ of all of the values of vibration RMS seen so far (baseline) are calculated as well. A circular buffer is implemented to allow for capturing the vibration RMS data over a 30 min period of pump operation, i.e., after 30 min, the new data sample will overwrite the 30 min old value. The buffered data is split into n batch data that includes the newest sample, and the rest of the data constitutes the baseline data. As the window W moves, the mean value of the batch window μ_{batch} is continuously compared to the mean of the values in the baseline $\mu_{baseline}$.

Decision making

For the degradation onset detection, the DT decision making part uses a basic statistic approach that assumes that the distribution of the RMS (health indicator) in the healthy state is approximately normal, then about 99.7% of the data points should lie within three standard deviations of the mean of the baseline. Therefore, a threshold is set at

$$\mu_{baseline} + 3 \times \sigma_{baseline}$$

A degradation onset is detected if the mean of the batch window is above the threshold.

A confidence interval estimate is associated with the degradation onset detection model. The confidence interval estimate is calculated from the sample data to determine the range likely to contain the population parameter (mean, standard deviation) of interest. The confidence interval is determined by calculating the probability that the mean of the batch data falls within the $\mu_{baseline} + 3 \times \sigma_{baseline}$ range which contains the healthy data population. A statistical model is used to find the number of standard deviations between the mean of the batch data distribution and the 3rd standard deviation of the baseline distribution as follows:

$$A = \frac{(\mu_{baseline} + 3 \times \sigma_{baseline}) - \mu_{batch}}{\sigma_{batch}}$$

Then the confidence interval is found by evaluating a normal cumulative distribution function of the values in **A**.

Another important piece of the decision making part is a state machine that classifies the health state of the pump and provides a health percent indication. As far as a degradation onset is not detected, the state machine is at its initial state and reports a healthy state, a 100% health indicator, and the associated confidence interval. Once a degradation onset is detected, the state machine transitions to the degrading state. For the example of pump bearing failures, degradation was observed from historical data to be linear. Thus, a linear model was defined to estimate the health percent while the pump is in the degrading state. Once the vibration RMS reaches a very high limit, the state machine transitions to a faulty state. The parameters (e.g., the high-level vibration limit) for the linear model can be adjusted by the user. The state machine can also transition to a maintenance state from the degrading and faulty states to clear the degradation onset event before going back to the initial state (Normal operation state).

Process health monitoring DT

Concept

The purpose of the process health monitoring DT is to assess a control loop in terms of set-point tracking evaluation, process anomaly detection, and process health percent estimation with confidence intervals.

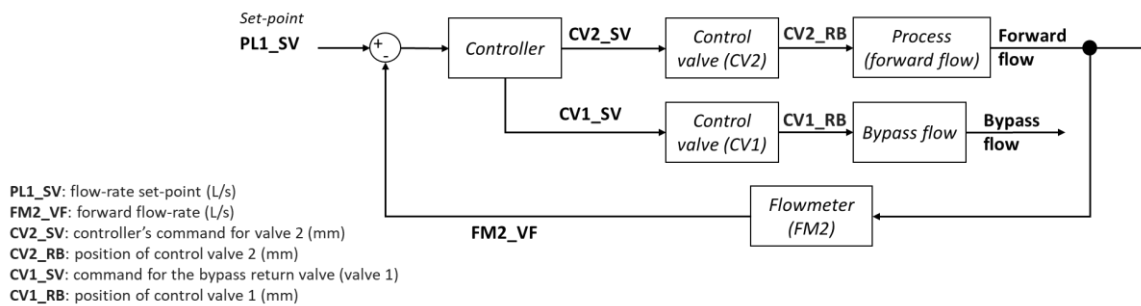


Figure 23: Monitored control loop

A schematic of the oil loop in the system is illustrated in Figure 23. The monitored process variable is the flowrate. The loop consists of forward flow and bypass flow branches that are controlled with a PID controller such that the control commands to the two branches add up to a constant. The process health monitoring DT uses data from the measured flow, set-points, and the controller command to identify any deviations from the expected normal behaviors and classify abnormal behaviors. The DT uses a model that outputs a set-point tracking index, which is an evaluation of whether the set-point is tracked or not with a confidence interval. The decision making part of the DT uses the output of the process DT model as well as the output of the PID performance monitoring DT to detect and classify process anomalies. Thus, the

decision making part of the DT will be discussed after the PID performance monitoring DT is introduced in the subsequent subsection.

DT model

The process DT model shown in Figure 24 provides an indication on whether the measured flowrate matches the input set-point or not. The model is deployed as a state machine in Simulink/Stateflow that mainly monitors the flowrate in the production steady state. The model uses the operation status (*Op_Status*) input from the process simulation to detect the transitions between the process standard operating procedure states (i.e., stopped, ready-to-start, starting, production sub-states, and stopping). During a production state, the DT model buffers flow data samples over a 1 second period and compares them to the actual set-point with limit bounds set at 3 standard deviations from the set-point value. The standard deviation is calculated from the measured flowrate variability. If the flowrate is within the limit bounds, then the DT model reports that the behavior is normal through the tracking index. Else, the DT model reports an abnormal behavior (i.e., flowrate doesn't match the set-point).

A confidence interval is associated with the set-point tracking index. The confidence interval is calculated in a similar way to the pump DT confidence interval. It is calculated by evaluating a normal cumulative distribution function for the values of the range determined by how far the mean of the batch data is from the limits of three standard deviations from the set-point.

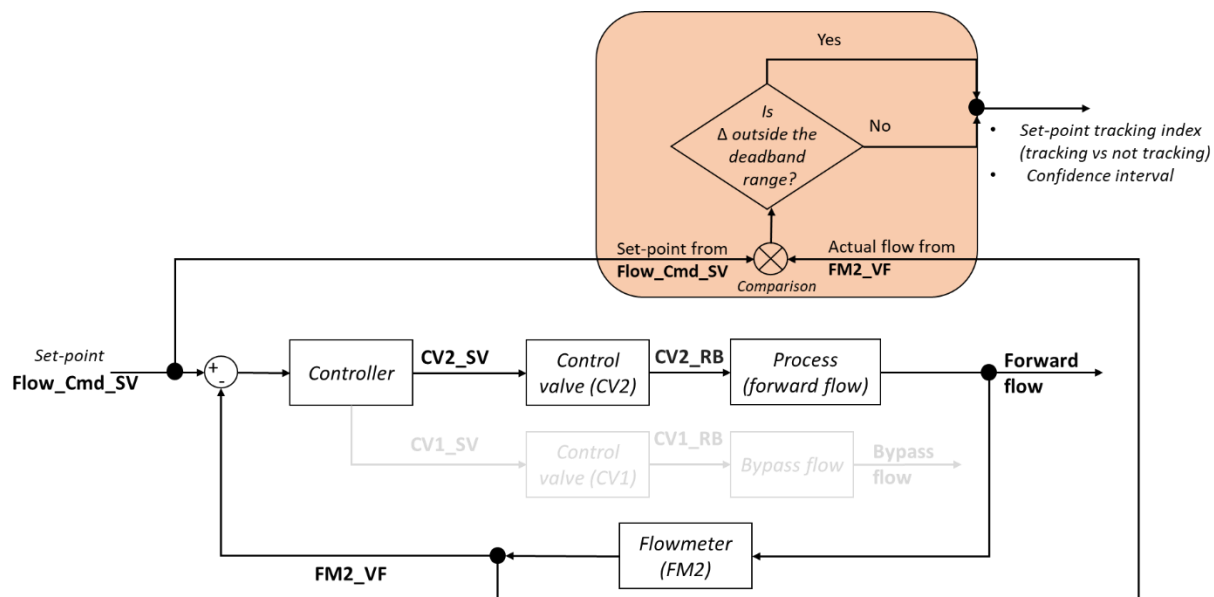


Figure 24: Concept of the set-point tracking DT model.

PID controller performance monitoring DT

Concept

The purpose of the PID controller performance monitoring DT is to assess a control loop in terms of control behavior evaluation. The PID controller output is monitored by using a control command prediction model to evaluate the controller output for a given behavior. This model uses the input set-point to predict the output of the PID controller to the control valve. More importantly the output of the prediction model is used to detect process and controller anomalies that will be discussed in the DT decision making subsection.

DT model

The DT model for the PID performance monitoring DT shown in Figure 25 provides an indication on whether the control command is as expected for a given behavior or not. Expected normal behaviors are learned from historical normal operation data. A linear regression model was trained to predict the control command for a given set-point. The output of the prediction model is compared to the actual PID controller output.

Similar to the process DT model, the controller DT model is deployed as a Simulink/Stateflow state machine that monitors the action of the controller in the production steady states. A one second batch of data is cumulated and compared to the predicted control output. If the actual control command matches the predicted one, then the controller DT model reports that the behavior is normal through the control index. Else, the controller DT model reports that the controller is excessively compensating, which is then used in the decision making part to detect and classify anomalies.

A confidence interval is associated with the control index that the controller DT model outputs. The confidence interval is calculated in a similar way to the pump DT and process DT confidence intervals. It is calculated by evaluating the normal cumulative distribution function between the mean of the batch data and the limits at 3 standard deviations from the mean of the predicted control value.

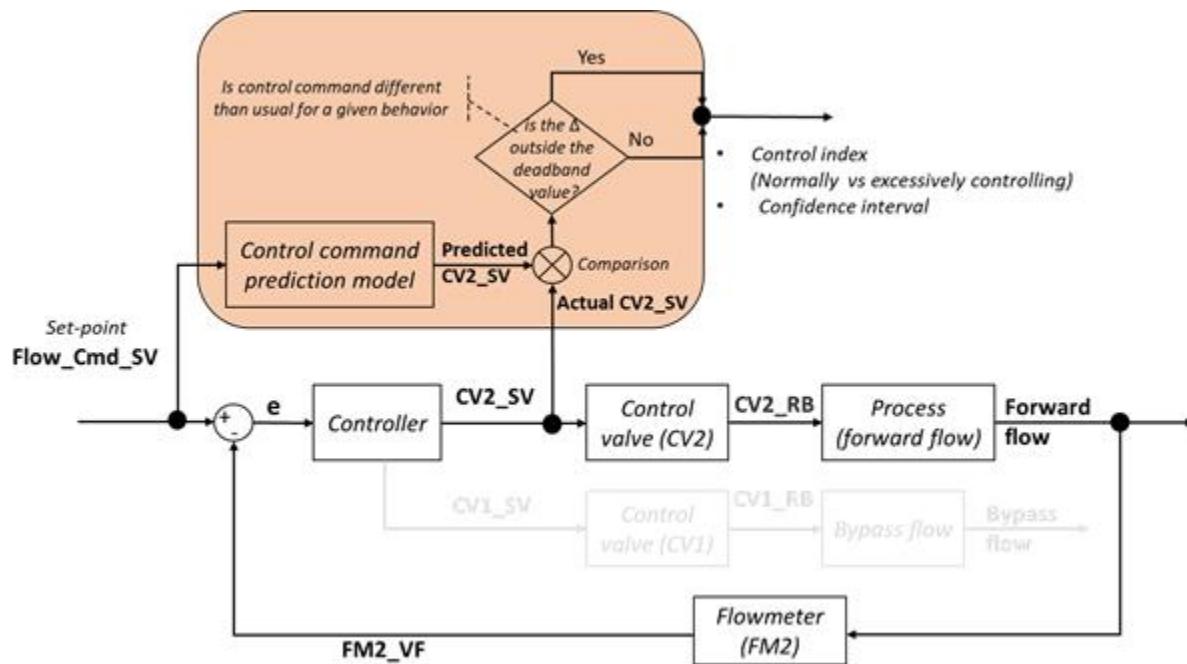


Figure 25: Concept of the PID performance monitoring DT model.

Decision making

The decision making component uses the outputs of the process DT model and the controller DT model to detect and classify behaviors. The decision making consists of the following (see Figure 26):

- If the controller output is as expected for a given behavior and the set-point is tracked, the DT reports a normal behavior + confidence intervals.
- If the controller is excessively compensating (more control than expected for a given behavior), and the set-point is tracked, then a process anomaly is detected and classified. This first type of process anomaly is described as an anomaly that the

controller can compensate for. An example of this anomaly is a small leakage in the forward flow.

- If the controller is excessively compensating (more control than expected for a given behavior), the set-point is not tracked, and the controller is railed (i.e., controller hits its maximum and still cannot compensate), then a process anomaly is detected and classified and the controller DT indicates that the controller is railed. This second type of process anomaly is described as an anomaly that the controller cannot compensate for as it hit its maximum. An example of this anomaly could be due to a control valve clogging.
- If the control command is different than predicted, the set-point is not tracked, and controller is not railed, then a controller anomaly is detected. Note that this could be a controller AND process anomaly. We now assume only one problem at a time.

The overall DT outputs include, process health index, controller performance index, process anomaly detection, controller anomaly detection, process health percent, controller health percent, and confidence intervals. The next subsection describes how these outputs are used and displayed in the human-machine interface.

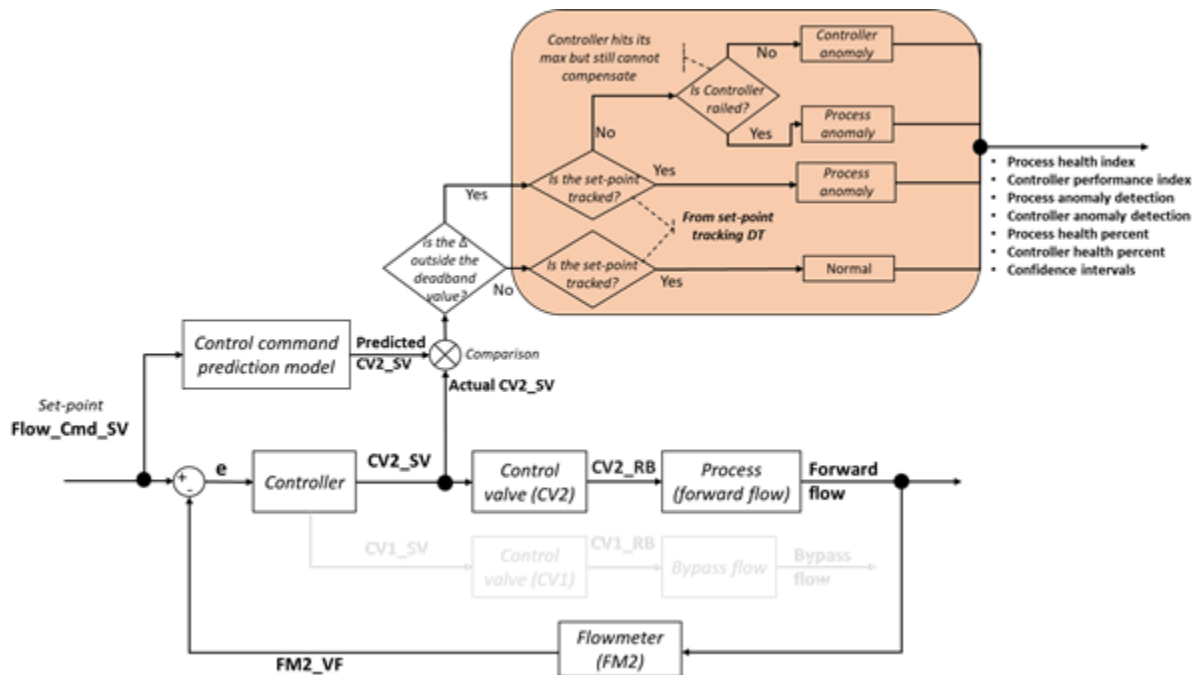


Figure 26: Process and controller anomaly detection and classification mechanism

User Interface

The three DTs described above are constantly analyzing real-time data from the virtual factory and from sensors throughout the ITE in order to determine the state of health for several aspects of the process and equipment. Interfacing between the DTs and the user interface shown Figure 27 in is described here.

Pump health monitoring DT

As described in section 2.2.2, the pump health monitoring DT provides an anomaly detection indication, a state classification, a health percent indication, and a confidence interval associated with the state classification. Figure 27 shows the user interface. The unit PUMP1 Health displays the outputs of the pump health monitoring DT. The state classification is shown

through three colors that the unit's box could take, namely, green for normal state, yellow for degrading state, and red for degraded or failure state. The health percent is shown in the middle left box of the unit, and the confidence interval is shown in the middle right gage. The bottom box displays the detected anomaly type. Note that the current development only detects bearing anomaly. Future extension could consider adding other anomaly types to the current model.

Process health monitoring DT

The unit called Flow Unit1 health shown in Figure 27 displays the outputs of the process health monitoring DT. This DT provides indications of the health state, health percent, anomaly type, and a confidence interval associated with health state classification. Similar to the pump DT, the health state classification is shown through three colors that the unit's box could take, namely, green for normal state, yellow for needs attention, and red for faulty state. The health percent is shown in the middle left box of the unit, and the confidence interval is shown in the middle right gage. The bottom box displays the detected anomaly type. As discussed above, two types of process anomaly can be detected. The first type is detected when the controller is excessively compensating and the set-point is tracked. The second is detected when the controller is excessively compensating, the set-point is not tracked, and the controller is railed.

PID performance monitoring DT

The unit called PID1 Performance shown in Figure 27 displays the outputs of the PID control performance monitoring DT. This DT provides an indication of whether the controller is normally or excessively compensating, an indication when the controller is railed, a controller health percent, an indication of controller anomaly, and a confidence interval associated with the controller state (normal vs excessively controlling). Similar to the pump and process DTs, the state classification is shown through three colors on the unit's box, namely, green for normal state, yellow for needs attention, and red for faulty state. The health percent is shown in the middle left box of the unit, and the confidence interval is shown in the middle right gage. The bottom box displays when the controller is railed or when there is a controller anomaly. The controller is railed when it hits its maximum control value but still cannot compensate. A controller anomaly is detected when the set point is not tracked and controller is not railed.

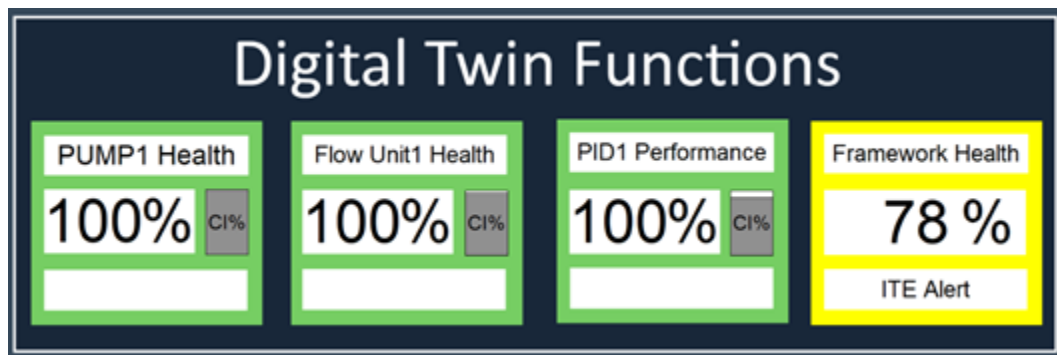


Figure 27: DT displays in the user interface

Future DT Development

Future development will consider interconnection with the physical system (Part A) in addition to emulation. Also, more DT capabilities will be considered to realize a system wide DT framework that would allow for live monitoring, testing What-If scenarios, predict future behaviors, and prescribe recommendations. The system wide DT framework will entail a hierarchy of DT classes for equipment, process, product, and even persons. Figure 28 displays an example DT hierarchy for system wide health monitoring. The hierarchy combines health monitoring DT

classes for pumps, valves, tanks, and for the process itself. Those DT classes could be instantiated (specialized) for the different components and control loops in the system. Similarly, other DT capabilities could be added to the hierarchy such as scheduling optimization, yield optimization, and quality assessment, etc.

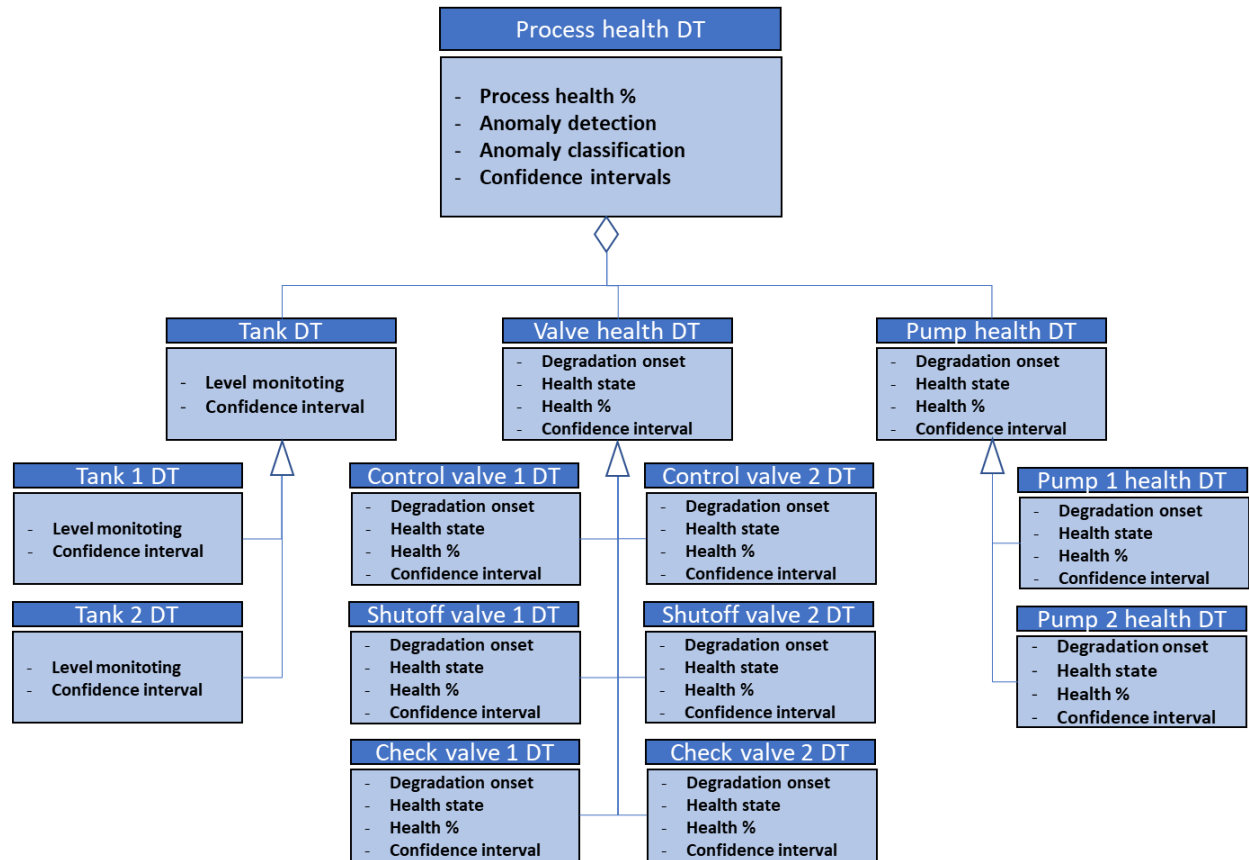


Figure 28. Example DT hierarchy

References

Qamsane, Yassine, James Moyne, Maxwell Toothman, Ilya Kovalenko, Efe C. Balta, John Faris, Dawn M. Tilbury, and Kira Barton. "A Methodology to Develop and Implement Digital Twin Solutions for Manufacturing Systems." IEEE Access 9 (2021): 44247-44265.

Moyne, James, Yassine Qamsane, Efe C. Balta, Ilya Kovalenko, John Faris, Kira Barton, and Dawn M. Tilbury. "A requirements driven digital twin framework: Specification and opportunities." IEEE Access 8 (2020): 107781-107801

VI. DISCUSSION & ANALYSIS

This section includes discussion on the industry impact, key performance indicators, accessing the technology, workforce development, and lessons learned.

Industry Impact

This project has proven open process automation and digital twin concepts in a representative industrial process manufacturing environment. While this is a significant step forward, a key next step will be to connect the technology to an actual process manufacturing environment, such as the Part A testbed.

The capabilities demonstrated in this project have the potential to benefit not only process manufacturers, but discrete manufacturers and government entities as well. The ability to implement open process automation and digital twin concepts, as well as other leading-edge technologies, in an open framework with connectivity to existing and future infrastructure, has the potential to greatly expand the reach and benefits from new technologies such as digital twin and open process automation.

Key Performance Indicators & Metrics

The following table summarizes the success criteria for this project.

Table 8: Success Criteria

SUCCESS CRITERIA	ACHIEVED?
Demonstrate and evaluate digital twin and open process automation technologies with cross-vendor systems	YES
Allow manufacturers to develop, test and evaluate new technologies without interrupting production operations and without costly R&D investments	YES

The following table summarizes the KPIs and benchmarks for this project.

Table 9: KPIs and Metrics

METRIC	BASELINE	GOAL	RESULTS
Data latencies	Multiple seconds	<1 ms	10 ms ²
Real-time jitter	Multiple seconds	<50 us	mxdM1: +/- 6 us mxdE1: +/- 300 us mxdE2: +/- 1000 us mxdE3: +/- 1000 us mxdE4: +/- 1000 us
Framework reconfiguration time	Days	<5 mins	<5 mins
Framework configuration training requirements	Weeks	<2 days	<2 days
Distributed framework computing performance	Often not available	<1 ms	10 ms ²

² During requirement discovery, these KPIs were given lower priority as <1ms performance was not needed for a successful technology demonstration



Number of standard data protocols supported	Often limited to 1 or 2	5+	3 demonstrated 6+ supported
---	-------------------------	----	--------------------------------

The following sections discuss each of the KPIs and the methods used for measurement and validation.

Data latencies

Data latency is a measure of real-time performance. Data latency is the maximum amount of time between an event and when it is realized by the system. In the industrial open process automation framework of this project, all of the systems have been synchronized to a 10 ms frame rate, thus the maximum latency in the system is 10 ms. Any violation of this frame rate would result in a frame overrun, which is one of the standard statistics that are built-into the ADEPT Framework. During factory acceptance testing, the system was validated to confirm that no frame overruns are experienced.

Our target for this project was 1 ms data latency, but during the requirements discovery phase, the team determined that 1 ms timing would not be needed to meet the goals of this project. Therefore, the system was designed with a 10 ms frame rate to better accommodate the lower powered single board computers in the system.

Real-time jitter

Jitter is another measure of real-time performance. Real-time jitter is a measure of a system's ability to meet its timing requirements, i.e. if the system is expected to process a compute cycle every 10 ms, then the jitter is how close to exactly 10 ms does the system start its compute cycle every time. Real-time jitter is another statistic that is built-into the ADEPT Framework.

Real-time jitter is highly dependent on the server architecture and the model. The Dell R540 (mxDM1) has very good real-time jitter with a maximum observed jitter of +/- 6 us. The Jetson (mxDE1) has pretty good jitter, but it will occasionally spike to give us an observed maximum jitter of +/- 300 us. Jitter on the TI-ICEs (mxDE2-4) was much higher, up to +/- 1000 us, which is expected from such lower power computers.

In terms of our target of <50 us jitter, we easily achieved this with mxDM1. MxdE1 meets the target much of the time, but mxdE2-4 just don't have the compute capacity to meet this.

See the final tab of the FAT workbook, test cases for deliverable #4, the Digital Twin and Open Automation Framework, for more details on the jitter measurements taken during FAT testing.

Framework reconfiguration time

Framework reconfiguration time is a measure of the ability to switch between applications on the system. The ADEPT Framework makes switching between applications or modifying an existing application very straightforward. To load a framework in ADEPT, one simply needs to open the framework and then download it to the system. The "9.2.1-2 Frmwk Administration" tab of the FAT workbook shows how a project can be stopped, closed, and then a new project loaded in VI.

Reconfiguring the deliverable #4, the Digital Twin and Open Automation Framework, can be readily achieved within the 5-minute target.

Framework configuration training requirements

The ADEPT Framework was developed using open interfaces, industry standards and engineering best practices. Once an engineer understands the basic ADEPT structure and concepts, then it is very straightforward to work with. ADI has a standard self-paced training course that introduces the ADEPT Framework concepts using a small mobile inverted pendulum

robot. This training course can be completed easily within two days. Alternatively, training can be provided for a specific system like this, and that training can be accomplished in much less than two days.

Distributed framework computing performance

One of the benefits of using the ADEPT framework is that it has built-in support for distributed frameworks that link multiple servers into a common data and control environment. The distributed environment is synchronized at a certain frame rate, which is often implemented at 1 ms or faster. However, for this system, we determined during the requirements discovery phase that it made more sense to run this framework at 10 ms. The data from the fieldbus protocols will not be changing faster than 10 ms and the digital twin algorithms wouldn't need data faster than 10 ms for anything.

Number of standard data protocols supported

The ADEPT Framework supports over 14 standard data protocols for aerospace and defense, automotive and industrial applications. The following table shows the standard industrial data protocols support by ADEPT and identifies which are demonstrated in this project.

Table 10: Standard Industrial Data Protocols Supported and Demonstrated

Industrial Data Protocols	Supported	Demonstrated
OPC-UA	Yes	Yes
Modbus TCP	Yes	Yes
UDP Ethernet	Yes	Yes
EtherCAT	Yes	No
Ethernet/IP	Yes	No
CANopen	Yes	No
RS-422/485	Yes	No

Accessing the Technology

The technology developed in this project leverages the following commercial software platforms:

- ADEPT Framework from Applied Dynamics International
- MATLAB/Simulink with the Simscape toolbox from the Mathworks

While MATLAB/Simulink is required to develop and compile the digital twin and process models, it is not required to run the project and work with the technology. The ADEPT Framework, however, is required to run the project, but a permanent license for ADEPT has been installed on the main PC that is part the project deliverables that reside on the factory floor at MxD. It is also important to note that the project has been developed to run specifically on the ITE system that has been developed for this project and resides on the factory floor at MxD.

The projects files, on the other hand, are all available to MxD members through the MxD Members Portal. These project files include the design and operating documentation, as well as recorded demonstrations and presentations for this project.

Workforce Development

The project includes operation manuals for both the ITE hardware itself as well as the ADEPT Framework example project that has been developed to showcase the project concepts.

Lessons Learned

Many lessons were learned throughout this project.

What went well

- **Requirements Gathering:** The team started the project by documenting and agreeing upon the requirements. The team followed a very structured approach by using an Excel worksheet to document each requirement, how it was going to be met and the level of technical risk. The team quickly realized that some key terminology needed to be defined and agreed and included that as part of the requirements activity. While this requirements process was time consuming, it proved extremely valuable to align the expectations of the team and allow the rest of the project to progress smoothly.
- **GNU Scientific Library (GSL):** The team identified some key GSL functions that were needed to implement effective and realistic noise characteristics for the process model, so the GSL functions were added to the development environment and proved to be very effective and useful. In fact, the team found the GSL functions so useful, that ADI now has plans to include the GSL as part of the standard ADEPT distribution.
- **Digital Twin Approach:** To prove the concept, we started with single input, single output digital twins. This allowed us to prove out the interfaces, methods and tools. This worked very well and will allow more complex and inter-related digital twins to be developed next.
- **Signal Rates:** We found that implementing the continuous built-in tests (BIT) was very beneficial to ensure all data were being updated on time and at the proper rates.
- **Virtual Machine Verification:** In order to verify remote connectivity, we prepared and supplied virtual machines (VM) that represented an ADEPT client and server. We supplied these VMs to MxD for hosting in their infrastructure. The team found these VMs very useful to test remote connectivity to the MxD infrastructure. The team even relied on these VMs to access the main project deliverables since the correct network interface was not setup until a few weeks after the initial delivery of the ITE and project system.

Areas for improvement

- **Simulated Data:** Preparing suitable simulated data took significantly longer than anticipated. Since physical process manufacturing equipment was not available to gather real data during the period of performance for this project, the team decided to use simulated data. The team found that extra fidelity and complexity was needed to use simulated data as a number of factors needed to be modeled and implemented in a realistic manner, including noise, vibrations and component degradations.

While some reference data was found for pump degradation, there was a surprising lack of publicly available data for the types of process manufacturing equipment involved in this project. Having access to more data, both normal and failure cases, would have been very helpful for creating realistic simulated data for this project.

- **Data Interfaces:** Data interfaces are notoriously challenging to get right. While the team had a comprehensive data interface document that included names, units and frequencies for all system data, we still had time consuming back-and-forth sessions before all of the assemblies were using compatible data interfaces. To reduce these issues in the future, the team would recommend using a similar approach with a data interface document but spending more time up front on reviewing the interface document and making sure that it is complete and understood by all stakeholders.
- **Leveraging Simscape for Real-Time:** The team was able to successfully implement the process module using the Simscape toolbox, but there were some challenges. While the Simscape models were easily converted to C-code that could be run on the



framework, there were some challenges to parameterizing the generated code so that it various aspects of the process could be controlled by the framework after the model was generated.

- **Factory Network Configuration:** While the configuration of the switches and various networks that are a part of this project went well, the team did not well understand the remote access network configuration requirements to enable a connection to the rig on the factory floor. The team finally settled on adding a dedicated router to the system to provide remote VPN access to one of the systems, but it took a significant amount of back and forth to understand the best approach for this. While the teams had a fair amount of coordination early on in the program, we would recommend more of a focus on the specific interface strategy and infrastructure requirements earlier on in the project.

Note that this would have been much more challenging without the excellent support from the MxD IT team. They knew exactly how to best set things up and were extremely helpful to get the system compliant and running properly with the rest of the infrastructure.

- **Real-Time Integration of the Digital Twins:** One of the benefits of this project was to enable the latest digital twin technology to be implemented in a real-time environment. There are significant structural differences when an algorithm is connected to a static data set versus a real-time stream of data. Since the original digital twins needed to be developed on static data (before the real-time data was available), there were some challenges to transforming the digital twin models to work properly on the live data, including getting the data frequencies and interfaces understood by all teams. To mitigate these issues in the future, the team would recommend reviewing the real-time implementation strategy with all affected team members earlier in the project.

Identified future project potential

The team identified the need for more reference data in order to build better simulations and more realistic digital twin implementations. There is a potential for a future MxD project to create a data bank that makes manufacturing equipment and process data more readily available for innovation activities.



VII. CONCLUSIONS & FUTURE WORK

This project was able to successfully demonstrate an industrial open process automation framework with multiple digital twin and open process automation applications running in real-time in a representative industrial environment that includes interfaces from multiple suppliers. The next steps are to further validate and prove out this technology by connecting the system to a real factory and expanding the scope and capabilities of the digital twin and open process automation functions.

Next Steps & Challenges

The logical next step is to further prove out this technology by connecting the ITE and industrial open process automation framework to real factory equipment that is running real process manufacturing operations. This could be accomplished either on the MxD factory floor by connecting the delivered ITE to one of the factory testbeds or by connecting this (or a similar) ITE to one of the MxD manufacturing members' labs or factories in the field. A natural choice on the MxD factory floor would be to connect the ITE to the 19-04-04 digital twins for process manufacturing – mobile worker project, which is expected to be delivered to the MxD factory floor later in 2021.

When the ITE is connected to real factory equipment, the digital twins and open process automation functions will then be exposed to live, real-time data that will enable the next critical validation of this technology. Once the live-data validation is complete, then the focus can shift to enhancing the digital twin and open process automation capabilities and performance.

The digital twin and open process automation functions implemented in this project are limited in scope and complexity; however, once the concept is proven with real factory equipment, the door is open for more complex digital twins and open process automation functions to be implemented and proven in an open platform that can be applied to both new and existing factories without the need to upgrade existing factory equipment or limit factory equipment to that of a single supplier.

In 2-5 years, it is hoped that this technology will be connected to physical factory equipment and the digital twin and open process automation use cases will be matured to provide additional performance and capabilities.

The physical project deliverables will be accessible to all MxD members and guests on the MxD factory floor, and the electronic deliverables will be available to all MxD members on the MxD member portal. On the factory floor, the system will be setup to automatically boot into a running demonstration that can be interacted with on a factory tour or with other access to the Client PC for this project. Remote access to the project Client PC may be available as well.

The major risks and challenges of leveraging this project include further development and maturing of the digital twin and open process automation applications. The industrial open process automation framework technology has been proven over many years in very demanding aerospace and defense applications.

The commercialization strategy includes identifying and maturing the most beneficial initial applications that enable manufacturers to project enough return from the technology to justify the investment. Then starting with initial pilot applications that can be replicated on similar operations throughout a manufacturer's facilities.

Transition Plan

The table below provides a catalog of all of the project deliverables and their respective transition routes. Deliverables can transition through deployment at an industry member's site,



as an educational reference or through a commercialization effort. Each of these transition routes are detailed below.

Table 11: Deliverable Deployment Summary

# / DELIVERABLE NAME	TECHNOLOGY INTEGRATION	EDUCATION	COMMERCIALIZE
1. Requirements & Feasibility Report	X	X	
2. Integration Test Environment (ITE) Rig	X	X	
3. Digital Twin Models	X	X	
4. Digital Twin and Open Automation Framework	X	X	
5. Simulated Factory Framework	X	X	
6. Display wall, monitors, and Windows PCs	X	X	
7. Standards		X	
8. Project Final Report		X	

Requirements & Feasibility Report

The requirements and feasibility report is a structured Excel workbook that will be available on the MxD member portal and will be used both to understand that technology implemented on this project as well as for an educational reference for how to document and define requirements for a project like this.

Integration Test Environment (ITE) Rig

The ITE rig is the industrial computer system that is at the center of this project display on the MxD factory floor. This ITE rig is intended for educational reference for implementing an industrial open process automation framework capability, as well as a technology integration starting point where hardware modules can be added or modified add additional capabilities to the rig.

The ITE rig documentation includes a test and validation report that is provided as an educational reference for how to test and validate an industrial open process automation framework and ITE like this. This document will be available on the MxD member portal.

The ITE rig documentation includes an operation and maintenance manual that is meant as an educational reference for the safe operation and maintenance of the ITE rig. This document will be available on the MxD member portal.

The ITE rig documentation also includes CAD models that are provided as an educational reference for the ITE hardware. We have also included an export of renderings of the ITE from multiple angles. The CAD models will be available on the MxD member portal.



Digital Twin Models

The digital twin models developed for this project have been integrated in the Digital Twin and Open Automation Framework and will be available as an educational reference in the MxD member portal.

Digital Twin and Open Automation Framework

The digital twin and open process automation framework example is a .zip file that includes all of the implementation files that support the demonstration example for this project. These files will be available as an educational reference on the MxD member portal as well as a technology integration starting point for any future additions or modifications to the demonstration for this project.

The Digital Twin and Open Automation Framework includes a system design specification as the main design documentation for the industrial open process automation framework example and the ITE system hardware. This document is provided as a technical integration starting point as well as an educational reference. It will be available on the MxD member portal.

The Digital Twin and Open Automation Framework includes a framework user manual that is a document file that is available on the MxD member portal and is intended as an educational reference to explain how the example framework operates and how to interact with it.

Simulated Factory Framework

The process manufacturing model is integrated in the Simulated Factory Framework and is provided as a technology integration starting point as well as for an educational reference. While these files are included in the framework example files, they were separated here to provide easier access if desired. These files are MATLAB/Simulink/Simscape and c code. These files will be available on the MxD member portal.

The process manufacturing simulation includes documentation that is provided as an education reference for some of the design decisions that were implemented in the model. These files will be available on the MxD member portal.

Display wall, monitors, and Windows PCs

The display, monitors and PCs include the Client PC, which is the main operator interface for the industrial open process automation framework. This setup is provided as a technology integration starting point for future modifications and updates to the framework and also as an educational reference. These physical deliverables are located on the MxD factory floor.

The demonstration script and display monitor slideshows are provided as educational references and will be available on the MxD member portal.

Standards

The standards review report is included as an educational reference and will be available on the MxD member portal.

Project Final Report

The final report is this document that is provided as an educational reference and will be available on the MxD member portal.

APPENDICES

Appendix A: Definitions

What follows are a set of definitions, terms, and acronyms used in this document. These definitions were gathered from various source including the internet, reference papers, standards organizations, and the authors of these documents.

Table 12: Terms and Definitions

Term	Definition
Open Process Automation	Traditional automation technologies have been developed and controlled by a single (or small group) of companies that require their version of hardware and software to be used. While this enables those companies to control quality, it creates single-source lock-in, which limits users' options for integrating new technologies, thus reducing competition and motivation for new entrants to address cutting edge challenges. Open Automation (OA) technologies allow products from multiple vendors to interoperate as a single, cohesive system, thus enabling increased options and competition for delivering future innovation.
Digital Twin #1	A DT is a purpose-driven dynamic digital replica of a physical asset, process, system, or product. It is driven by a need for improvement of the manufacturing environment (e.g., monitor the health of an asset, reduce unscheduled downtime, generate a production plan, improve quality, etc.). A DT provides a capability in terms of detection, prediction and/or prescription so that it can deliver on its intended purpose and provide a value-add capability to a DT information client in the manufacturing ecosystem.
Digital Twin #2	A DT is a virtual representation of a physical entity (e.g. component, equipment, manufacturing line, or entire production facility) that can be monitored and compared against the real thing. A DT can be used in real time to detect anomalies and predict failures before they occur, and it can also be used at an accelerated pace (and with many copies in parallel) in a completely virtual environment. When a DT is used in a virtual environment, it enables testing "what if" scenarios much more efficiently and safely (without the risk of harm to people or damage to the facility) and can provide insight that greatly improves quality, efficiency and safety. And since a DT is virtual, it can be tuned and improved over time to provide better feedback and more effective predictions.
Plug-and-play	Ability to communicate across a pub/sub (publisher/subscriber) network. Ability to integrate COTS computer boards with minimum custom engineering effort and minimum downtime. Flexible reconfigurability of interface connectivity and computational assemblies (applications).



Plug-and-play component interchangeability	Individual components or component grouping can be interchanged with components serving the same function without reengineering (software and/or hardware reconfiguration) of the host environment
Uninterrupted online serviceability and expansion	Functionality continues to operate/execute uninterrupted during planned changes. E.g. Reconfiguration, component additions/deletions, software upgrades and application of hotfixes and patches
Self-Healing	Native Backup and Recovery, (ranges from simple corruption recovery, restore post hardware failure, and "go back" to features making the system ransom proof.)
Currently-out-of-scope HW and SW	Any and all commonly used data processing and manufacturing control components: e.g. DCS, PLC, Manufacturing Sensors and Actuators, Analyzers, 3rd party MES, 3rd party Historians, 3rd party analytics
Reliable data authentication, tracing	Secure data handling. Ensuring data received is coming from the intended source without alteration; blockchain or blockchain like
Integration Test Environment	An Integration Test Environment (ITE) is a system made up of COTS computer equipment and other equipment used to operate a plug-and-play computing and connectivity framework. An ITE is capable of providing to a device under test (DUT) all interfaces necessary to perform closed-loop testing. The ITE interfaces can be simulated, emulated or connected to real devices. A representative ITE can greatly speed-up development and testing for a device that interacts with assets that have low availability (high costs, high utilization, etc.) or that could be negatively impacted by test failures (safety concerns, equipment damage, etc.). An ITE can be an invaluable development and evaluation tool when it's impractical or undesirable to interact with the final assets. A portable ITE can also be an effective way to test a proof-of-concept to augment or update an existing installation.
Recovery from communications loss with external nodes	Communication recovery post interruption caused by either physical separation (intentional or unintentional) or software based separation (intentional or unintentional)
Unique operational and regulatory requirements of process industries	Aligns to "Uninterrupted online serviceability and expansion" and "Reliable data authentication, tracing," as well as long term data storage handling. (Long-term storage capacity not a requirement for this specific project. e.g. 75 year retention requirements for some operations)



Appendix B: Demos

The project has been setup in such a way that it can be easily demonstrated on the factory floor. The ITE and PCs have been setup in such a way that they will automatically start the appropriate displays and software upon power-up. Once the system has booted and the displays are running, then a demonstration can be given.

The Digital Twin and Open Automation Framework deliverable contains a demonstration script that can be read to a group of visitors. The script also contains instructions for the interactive part of the demonstration and should take roughly 15-20 minutes to conduct.

The project also included a virtual demonstration of the Digital Twin and Open Automation example running on the ITE prior to delivery to the MxD factory floor. A video recording of the demonstration over a web-meeting will be available in the MxD Member Portal.



Appendix C: Validation & Testing

The Integration Test Environment (ITE) Rig deliverable documentation contains an ITE Rig Test and Validation Report for complete test plan and report.

Here is a screenshot of the cover page:

FAT Test Matrix

ADI APPLIED DYNAMICS INTERNATIONAL

Customer	MNO
ID Number	11815
System Description	Open Automation and Digital Twin ITE

Nomenclature used within workbook	
Status	
Pass	Test has passed - refer to Test Results worksheet
Fail	Test has failed - refer to Test Results worksheet
Not Tested	Feature has not been tested - see comments
Compliance	
Compliant	Feature complies with requirement
Compliant (with New Feature)	Feature complies with requirement with the addition of a new feature
New Feature	Compliance status pending new feature
Non-compliant	Feature does not comply with requirement
N/A	Not Applicable
Verification	
Test	Requirement verified via test
Inspection	Requirement verified via inspection

System Test Requirements

The following requirements will be used for FAT (factory acceptance testing)

Requirement No.	Requirement	Compliance	Verification Strategy	FAT Test Status	SAT Test Status	Test Results	Comments
9.1 Equipment Interface Requirements							
9.1.1	Shall support OPC-UA communications	Compliant	Test	Pass 2/23/21 AK	Pass 3/29/21 AK /CS	9.1.1 OPC-UA	The ITE shall support and demonstrate the following communication interfaces: 1. OPC-UA 2. Modbus-TCP
9.1.2	Shall support interfacing with as many as possible of the following industrial devices: 1. Sensors and actuators 2. DCS and PLC 3. Analyzers (GC gas chromatograph) (ability in open fashion) 4. MES and data historians (ability in open fashion) 5. Asset management systems (ability in open fashion) 6. ERP (ability in open fashion)	Compliant (with New Feature)	Test	Pass 2/23/21 AK	Pass 3/29/21 AK /CS	9.1.2 Modbus-TCP	
9.1.3	The ITE shall provide the ability to switch between emulated and actual interfaces	Compliant	Test	Pass 2/25/21 AK	-	9.1.3 Patch Panel	
9.2 Operator / User Interface Requirements							
9.2.1	Shall support remote administration of industrial plug-and-play frameworks	Compliant	Test	Pass 2/16/21 AK	Pass 3/29/21 AK /CS	9.2.1.1 Framework Administration	
9.2.2	Shall support framework administration, including: starting, stopping and loading new frameworks	Compliant	Test	Pass 2/16/21 AK	Pass 3/29/21 AK /CS	9.2.1.2 Data Acquisition	
9.2.3	Shall support flexible configuration of data acquisition from anywhere within the framework	Compliant	Test	Pass 2/16/21 AK	Pass 3/29/21 AK /CS	9.2.1.3 Data Acquisition	
9.2.4	The framework shall implement an HMI for digital twin operation and indication	Compliant	Test	Pass 2/19/21 AK	Pass 3/29/21 AK /CS	9.2.4.1 HMI	
9.2.5	The framework shall implement an HMI for open process automation operation and indication	Compliant	Test	Pass 2/19/21 AK	Pass 3/29/21 AK /CS	9.2.4.2 Supervisory Interface	
9.2.6	The framework shall support separate interfaces for engineering vs. operator environments	Compliant	Test	Pass 2/19/21 AK	-	9.2.4.3 Supervisory Interface	
9.3 Functional Requirements							
9.3.1	Shall support a modular automation, analysis, and reporting language or scheme	Compliant	Test	Pass 2/25/21 AK	Pass 3/29/21 AK /CS	9.3.1 Python Scripting	Client-side Python scripting used throughout testing.
9.3.2	Shall deliver open automation function	Compliant	Test	Pass 3/17/21 CS	Pass 3/29/21 AK /CS	9.3.2.1 Open Automation	
9.3.3	Shall deliver built-in test function, a combination of instantaneous and continuous would be most desirable	Compliant	Test	Pass 3/17/21 CS	Pass 3/29/21 AK /CS	9.3.2.2 UBT	3/29/21 - Verified FTE thermal, electrical loads were nominal via DT-DA panel
9.3.4	Shall deliver digital twin functions)	Compliant	Test	Pass 3/17/21 CS	Pass 3/29/21 AK /CS	9.3.4.1 Digital Twin	
9.3.5	Shall deliver digital twin capabilities for the oil pump and oil control loop process	Compliant	Test	Pass 3/17/21 CS	Pass 3/29/21 AK /CS	9.3.4.2 Digital Twin	
9.4 Cybersecurity / IT Requirements							
9.4.1	Shall support user authentication	Compliant	Test	Pass 2/16/21 AK	-	9.4.1 User Authentication	
9.5 Digital Twin Interfaces and Industrial Computing Framework Support							
9.5.1	Interfaces between digital twin algorithms (both) and industrial computing framework (ADPIT) shall be standards based, when possible, and open and documented.	Compliant	Inspection	Pass 2/16/21 AK	-	Source files located at [path]	
9.5.2	Industrial computing framework shall provide the capability to log runtime data	Compliant	Test	Pass	-	9.5.2.1 Log Data	
9.5.3	Industrial computing framework shall support data replay to allow historical data to drive digital twin algorithms	Compliant	By Design	Pass 3/17/21 CS	-	-	ADPIT supports playback of data with real-time schedules.
9.5.4	DTs shall interface with runtime and historical data as input to the DT algorithms	Compliant	Test	Pass 3/17/21 CS	Pass 3/29/21 AK /CS	9.5.4.1 Digital Twin	In this example vibration data is played back at selectable rates via vibration C model and vibration parameter data.
9.5.5	Industrial computing framework shall support the real-time execution of Simulink models	Compliant	Test	Pass 3/17/21 CS	Pass 3/29/21 AK /CS	See 9.3.2.3	DA functions are implemented in Simulink.
9.5.6	Industrial computing framework shall support the real-time execution of C/C++ models	Compliant	Test	Pass 2/23/21 AK	Pass 3/29/21 AK /CS	See 9.3.1.2	Tests for 9.3.1.2 are implemented as open-loop C models.
9.5.7	Industrial computing framework shall support Python for the automation of client-side activities	Compliant	Test	Pass 2/25/21 AK	Pass 3/29/21 AK /CS	See 9.3.1	Client-side Python scripting used throughout testing.
9.5.8	The DTs shall have the ability to interface with other DTs	Compliant	By Design	Pass 2/19/21 AK	Pass 3/29/21 AK /CS	-	ADPIT framework supports interprocess communication.

Standard Test		Test Status		Test Results		Comments	
Full Network Check		Pass 3/8/21 AK		-		2/24/21 Test	
Collect system information						3/2/21 2/24/21 3/2/21	
Start and stop project at least 10 times						3/2/21 3/2/21 3/2/21	
PII Hardware/Software Functionality		Pass 2/25/21 AK		-		3/2/21 3/2/21 3/2/21	
Remove Project: 10 PII 3/2/21 (3/2/21 3/2/21)		Pass 2/25/21 AK		-		3/2/21 3/2/21 3/2/21	
Digital ID: 10 PII 3/2/21 (3/2/21 3/2/21)		Pass 2/25/21 AK		-		3/2/21 3/2/21 3/2/21	



Appendix D: User Resources

This section identifies the installation manuals, user guides, and additional materials necessary for a user to use the technology deliverables.

Digital Twin and Open Automation Framework Example User Manual

The Digital Twin and Open Automation Framework deliverable contains a user manual explains the Framework usage and user interface.

Operation and Maintenance Manual

The Integration Test Environment (ITE) Rig and Display wall, monitors, and Windows PCs deliverables both contain an Operation and Maintenance Manual (OMM) covering the following aspects:

- Safety Considerations for operators and maintainers
- Installation Guidance
- Control and Operation
 - Power System Overview
- Maintenance
 - Safety Considerations
 - Removing/replacing sub-systems and components
 - Maintenance Schedules