

# DIGITAL ENGINEERING SUMMIT

16-17 June 2025

Rome Marriott Park Hotel, Italy



EMULATE3D

by ROCKWELL AUTOMATION



Creating the Future of

# INDUSTRIAL OPERATIONS



Emulate3D  
TestRunner

Full Test Suite

C:\Users\ADeeble\OneDrive - Rockwell Automation, Inc\Emulate3D\Projects\E3D 2025 Feature Videos\Test Runner for Emulation

Zone 070

NEW TEST SCENARIO RUN ALL STOP

Name	Description	Configuration	Result
070PinStop1Retract	PinStop1 on 070 Fails to Extend	Action	Running
070PinStop2Retract	PinStop2 on 070 Fails to Extend	Action	Passed
070PinStop1Feedback	No AtExtended Feedback from PinStop1	Action	Passed
070PinStop2Feedback	No AtExtended Feedback from PinStop2	Action	Passed
060to070CamLiftTableFailure	Incoming lift table fails to raise	Action	Passed
070to080CamLiftTableFailure	Outgoing lift table fails to raise	Action	Passed
PX1Disconnect	No signal from PX1 prox	Action	Passed
PX2Disconnect	No signal from PX2 Prox	Action	Passed
PX3Disconnect	No signal from PX3	Action	Passed

Test Runner Aspect Viewer PX1



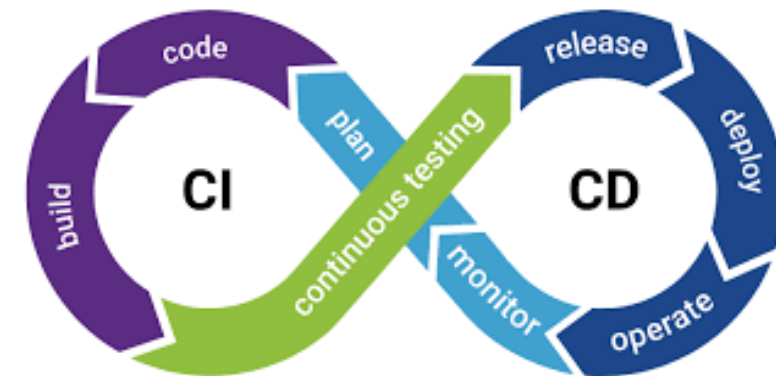
# | What is Test Runner?

Built on top of NUnit Testing Framework

User Interface which generates NUnit Tests  
...or create your own scripted tests

Application runs externally, launching  
Emulate3D from the command line  
...or use Test Runner inside Emulate3D

Enables powerful CI/CD workflows!



# Why use Test Runner

Single test framework for Simulation, Controls Testing, and Catalog Development

## Simulation

Compare models and layouts against different scenarios

Metaheuristic Algorithms for model optimization

Compatibility with existing data collection methods

**Experiments on steroids!**

## Emulation

Create and automatically run emulation regression tests

Run test scenarios with each controls code change

Perform fault simulation with fault framework

**CT Regression Test Dashboard!**

## Catalog Development

Create and automatically run script and catalog unit tests

Run test scenarios for each commit... more on that later!

Make full use of NUnit through scripting tests

**Facilitates CI/CD workflows!**

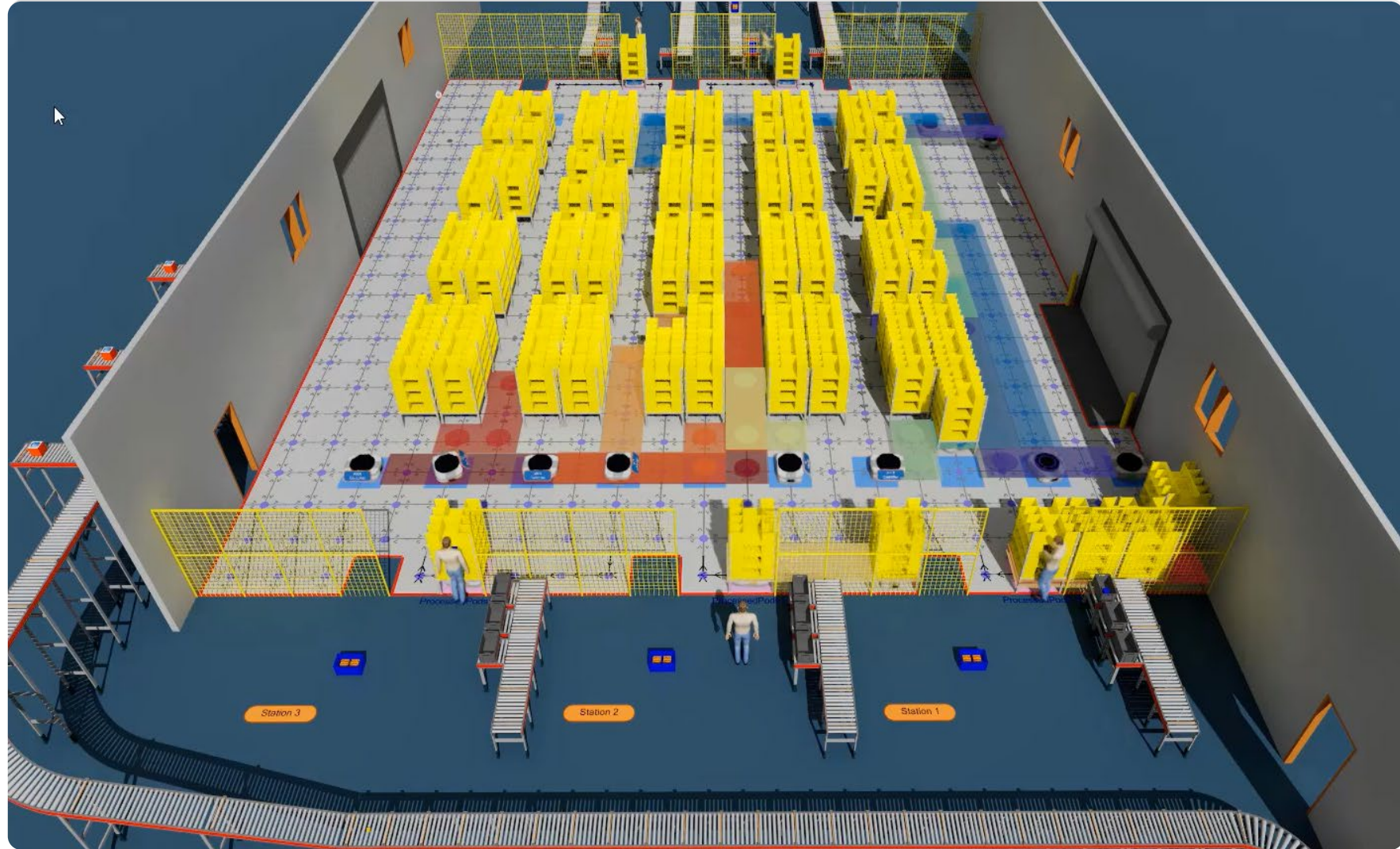


# RA | Example - Test Runner for Simulation

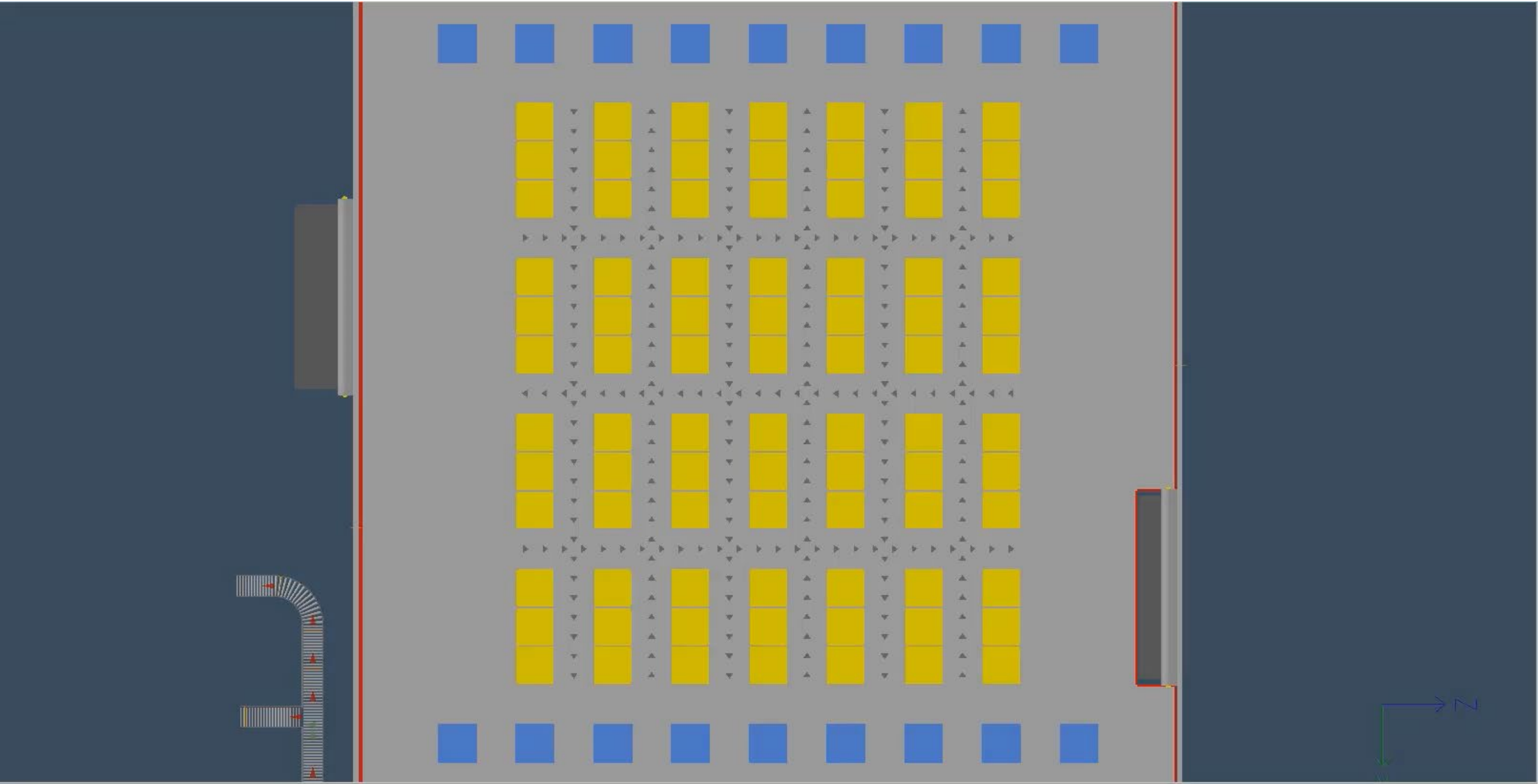
Test Runner as a successor to the Experiments feature

Use Test Runner to:

- Compare designs *such as different pod and aisle arrangements*
- Run "What If" scenarios, *such as a Picking Station going down*
- Vary model parameters to optimize a system, *such as AMR fleet size*



# Test Scenarios – Comparing Model Variants







# Creating Scenarios

Run a **Model** against **Actions**, and report on the results

Test Runner can be opened inside E3D, or as a separate application.

A Test Project can contain multiple models. These can be run against different scenarios.

Full data is collected in results folders. Summary KPIs can be seen inside Test Runner

Test Runner

Emulate3D TestRunner | AMR Testing | C:\Users\ADeeble\OneDrive - Rockwell Automation, Inc\Emulate3D\Projects\E3D 2025 Feature Videos\Test Runner for Simulation

### Layout Variations

NEW TEST SCENARIO RUN ALL STOP

Name	Description	Result
Layout 1	Pod Gap Pod Gap Pod	Not Run ▶ ✎ 🗑
Layout 2	Pod Pod Gap Pod Pod Gap	Not Run ▶ ✎ 🗑
Layout 3	Pod Pod Gap Gap Pod Pod	Not Run ▶ ✎ 🗑

Build Display World Location X 22.485 m Y 0.005 m Z 7.027 m

15 - Comparing Model Variants

EMULATE3D by ROCKWELL AUTOMATION



# | What's Planned – Replications and Reporting

Test Runner is being updated in each minor release, with a [roadmap here](#).

Easily create replications for each scenario.

Run a test multiple times with different random number generation.

Confidence intervals presented for KPIs.

Show the pinned KPI in the dashboard view.

Improved summary data outputted as csv.

Filtering and sorting of test results.

Test Runner

Emulate3D  
TestRunner

Replications C:\Users\Rizhar\Downloads

Search

NEW TEST SCENARIO

STOP

Search Test Scenarios

4

0

1

Name	Result	Action
Scenario 1	4 0 1	Throughput: 215.75 ± 1.88
Seed 0	Passed	Throughput: 215
Seed 1	Passed	Throughput: 218
Seed 2	Passed	Throughput: 217
Seed 3	Passed	Throughput: 213
Seed 4	Running	





# | Creating Actions

Create complex and flexible “What If” scenarios

Action schedules can be applied to a Scenario:

- Property setters change any property, aspect, or enable Faults.

- Property recorders capture headline KPIs, alongside data collectors

- Assertions check model behaviour, and pass/fail.

The screenshot displays the Emulate3D Test Runner software. The main window shows a 3D perspective view of a warehouse floor layout with several rows of yellow pallet racks. The floor is marked with a grid and various colored zones. At the bottom of the 3D view, there are labels for 'Build Display', 'World Location', and coordinates 'X 15.849 m', 'Y 0.01 m', 'Z 14.384 m'. On the right side, there is a 'Test Runner' panel. It includes a search bar, a menu icon, and a 'Group 1' section. Below this, there are buttons for 'NEW TEST SCENARIO', 'RUN ALL', and 'STOP'. A table lists test scenarios with columns for Name, Configuration, Model, and Result. The table contains one entry: 'Baseline Design' with configuration 'Action', model 'AMRs Single Aisle Double Pod.demo3d', and result 'Not Run'. At the bottom right, there is a 'Reset Layout' button and the 'EMULATE3D by ROCKWELL AUTOMATION' logo.

Name	Configuration	Model	Result
Baseline Design	Action	AMRs Single Aisle Double Pod.demo3d	Not Run

15 - Creating a Scenario



# Tips and Tricks – Generating Many Tests

Test Runner is being updated in each minor release, with a [roadmap here](#).

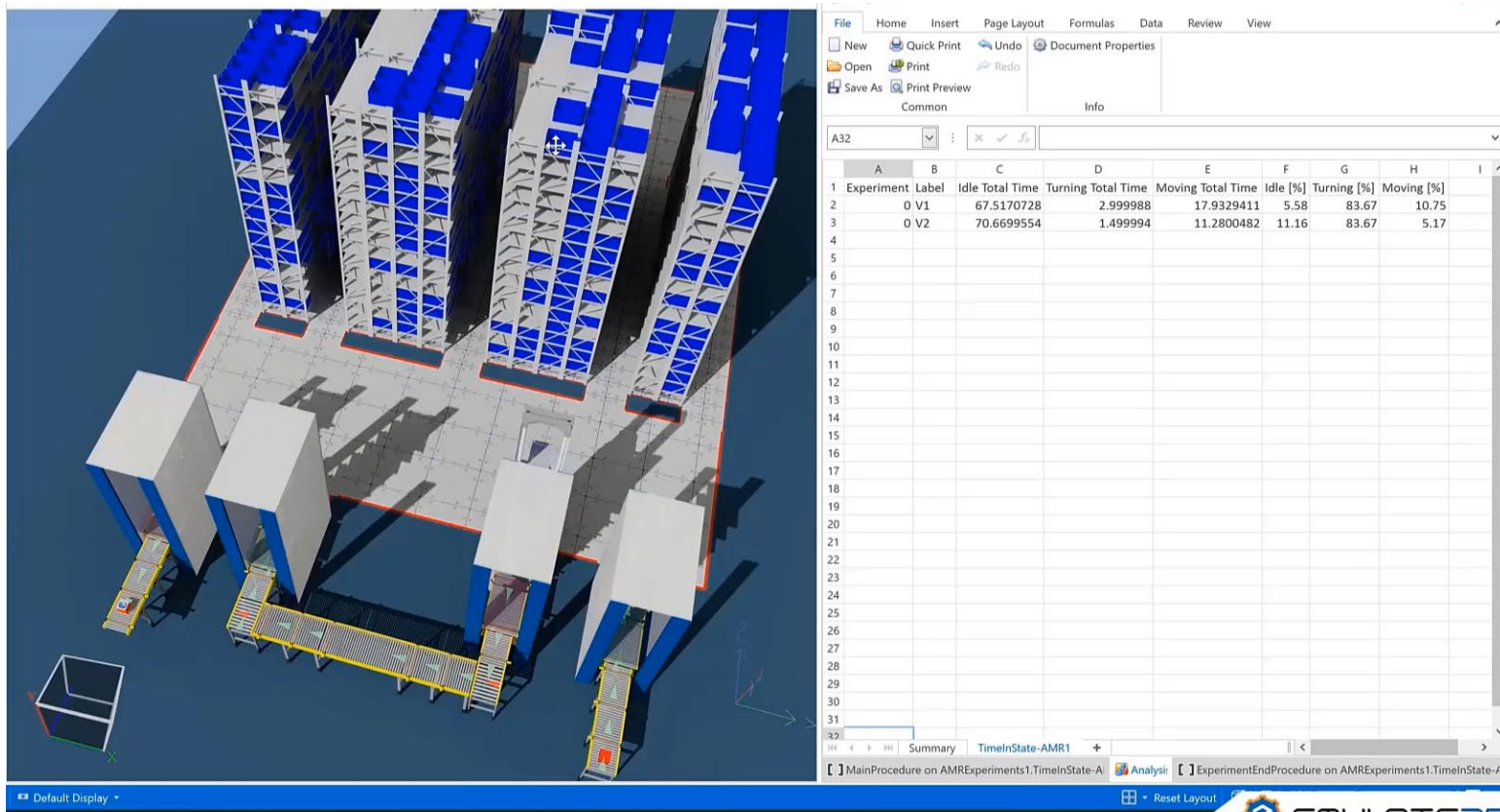
For example, sizing a fleet of AMRs. How can we run a test for 20, 21, 22, 23..... 39, 40 AMRs?

Note that the AMR framework has been updated for improved experimentation!

**Use the Optimizers.** The Brute Force optimizer is ideal for ranges of tests!

**Use “Based On”** to make one set of Actions inherit from other Actions.  
*Coming soon – actions triggered by events / delays from preceding actions*

**Customise Actions** – all editable JSON  
*Coming soon – import actions from CSV*





# | Model Optimization

Tools to automatically vary parameters to maximize a chosen goal

Find combination of parameters that maximizes some goal(s).

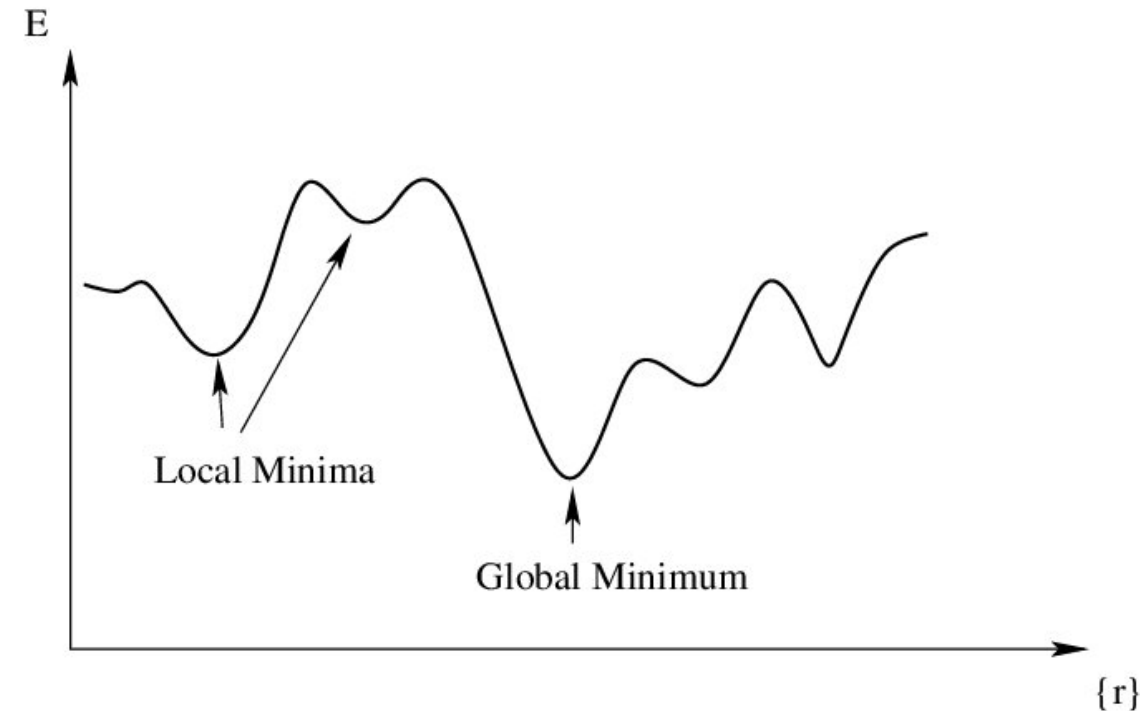
Parameters are user-specified:

- Number of vehicles, size of buffers or storage...
- Any numeric property (continuous or discrete)

Goals are user-specified

- Throughput, Utilization, Operating cost
- Any function that returns a numeric value

The gradient based methods you learned at school are not enough...





# | Metaheuristic Algorithms Inside Test Runner

Metaheuristics are algorithmic frameworks that help find solutions to complex optimization problems that are difficult to solve with traditional methods

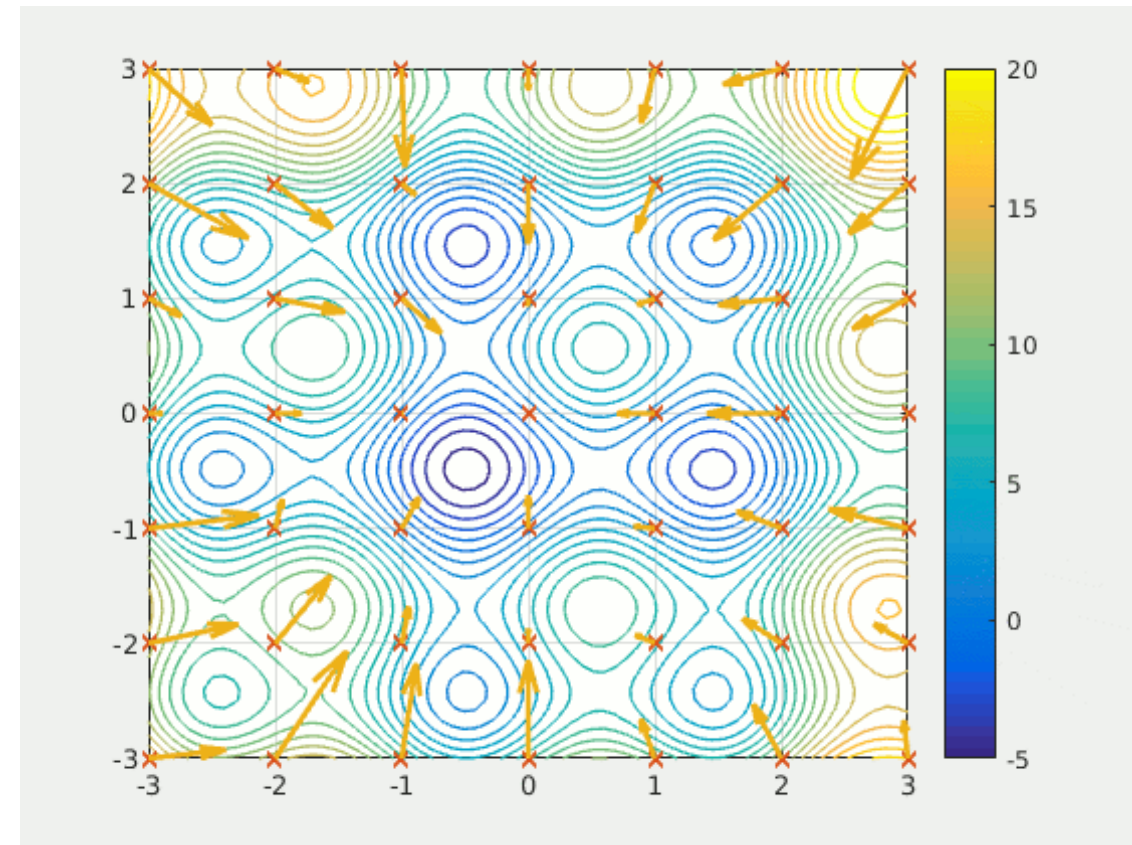
Major advantages for Emulate3D:

- Less likely to get trapped in a local minima
- Can be stopped at any time
- Can be easily parallelized

A balance between **Exploration** and **Exploitation**

**Exploration:** Venture into new areas of the search space

**Exploitation:** Refine promising solutions





# Metaheuristic Algorithm Options in Test Runner

Many options offered out of the box, suitable for different applications

Algorithm	Description	Advantage
Particle Swarm Optimization	Particles gravitate towards their historic best position and the global best position	Strong exploitation to refine best solution
Genetic Algorithm	Evolutionary approach, the best parents produce children with random mutations	Strong exploration to find novel solutions
<i>What's Planned – Tabu Search</i>	<i>Intelligently avoids searching previously explored regions of the search space</i>	<i>Reaches a conclusion with fewer tests</i>



# Optimizers

Automatically generate tests, learn from the results, and repeat to find an optimum solution

Specify parameters to vary and their range:

AMRs, 10 – 50,

Merge Delay, 0 – 0.4s

Specify objective function, the target goal  
e.g Throughput

Generates and runs tests  
Reads results, generates  
new batch of tests  
Reports results the user  
via dashboard

*Coming Soon – live data  
reporting as tests run*



# Stages of Virtual Commissioning

Can we connect?

**Simply connecting to the model finds issues:**

Map IO to virtual equipment

Connect the HMIs

Initialize and reset alarms

Check safety feedback

Can we run in manual?

**Begin testing system by forcing values:**

Verify sensor feedback

Dry run motors

Test manual functions

Step through the sequence

Can we run in auto?

**Create product and see how our system responds:**

Run idealized cycle on auto

Reposition field sensors

Verify predicted throughput

Test safety, stops, restarts

Ready to go on site?

Wait there is more!

# ...Maximising Value from Virtual Commissioning

Can we connect?	Advanced testing	Operator Training	System Upgrades
Can we run in manual?	<b>Testing in the virtual world is easier than in the real:</b>	<b>Familiarize and train on an accurate model:</b>	<b>Reuse the model to test potential changes:</b>
Can we run in auto?	Inject device faults	Familiarize with HMIs	Regression test code changes
	Check alarms & diagnostics	Run training scenarios	Optimize performance
	Run varied load schedules	Train on device failures	Test hypothetical scenarios
	Stress test the system	Grade operator responses	Reproduce issues virtually





# Using Test Runner for Emulations

Automated testing of your controls code, enabling CI/CD workflows

Create a suite of test scenarios:

- Inject simulated device faults
- Run different product variations
- Simulate operator interactions
- Stress test the system

Define expected behaviour with Assertions:

- Add to the model or to the scenario
- Define severity level, prerequisite conditions

**Run test suite from Test Runner every time  
you make a PLC code or design change**

**See a dashboard of test results**

The screenshot shows the Test Runner interface for a test scenario named "EStop on Turntable". The interface includes a sidebar with navigation icons, a search bar, and a table of test actions. The table has columns for Enabled, Time, Description, Source, and Severity. The actions are as follows:

Enabled	Time	Description	Source	Severity
	17	Set ESTOP1.EStopPressed to true		
	20	Assert that STA4.TRN.State equals false		High
	20	Assert that STA4.PRB.State equals false		High
	30	Set Restart1.ButtonPressed to true		
	33	Assert that STA4.TRN.State equals true		High
	33	Assert that STA4.PRB.State equals false		High
	36	Assert that STA4.PRB.State equals		High



# | Test Runner for Emulation

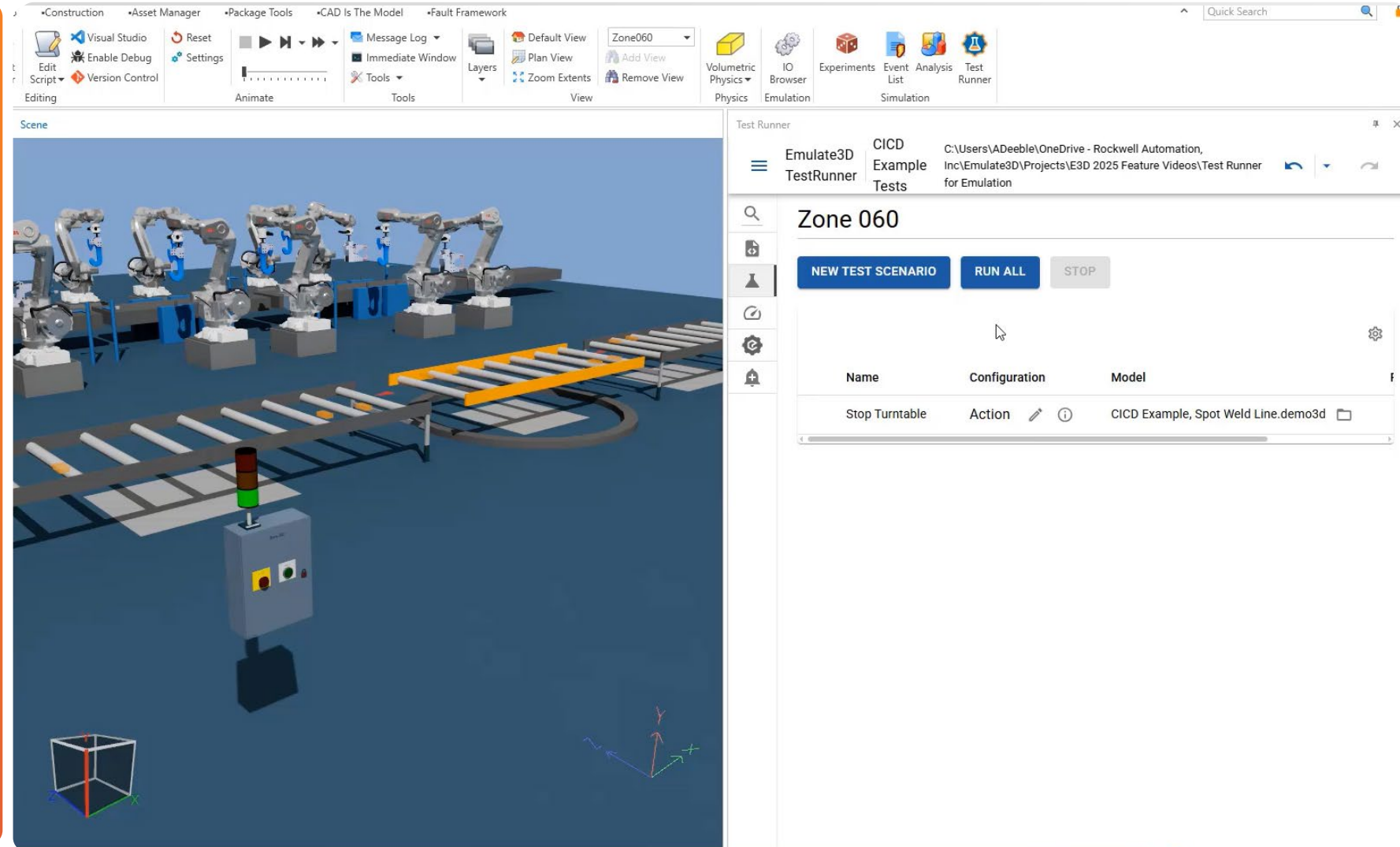
Create regression tests for your controls code

Create a set of repeatable actions which can be run against your PLC code.

Define Assertions in the test, or in the model.

Run the test suite, report on Pass/Fail/Warnings.

*Tip – The Emulation State component can help each test automatically connect*



FileHomeArrangeVisualizationHelpConstructionAsset ManagerPackage ToolsCAD Is The ModelTanks & PipesFault Framework

SelectNavigateFindFind NextEdit Custom PropertiesAspect ViewerEdit ScriptVersion Control

Visual StudioEnable DebugSettings

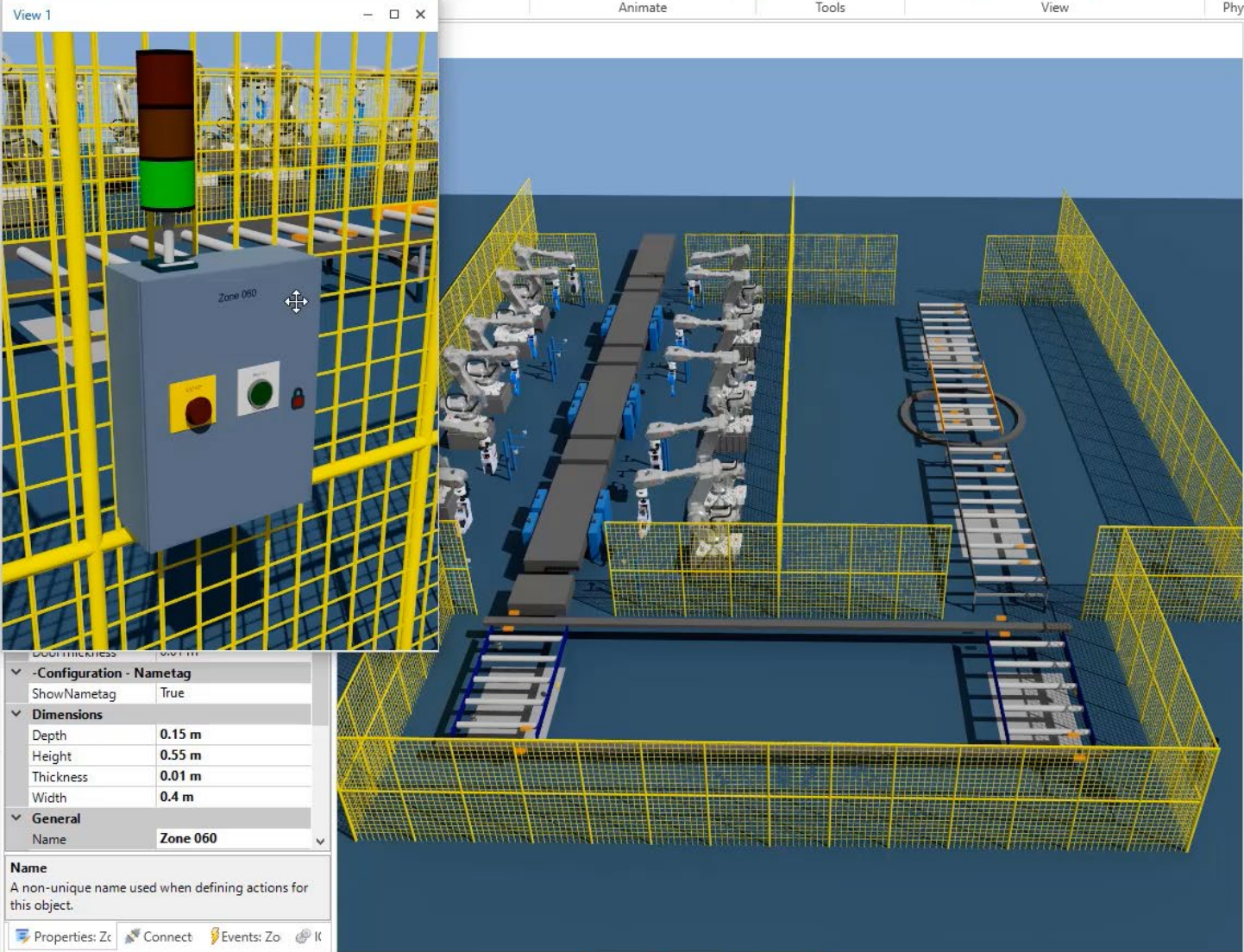
Message LogImmediate WindowTools

LayersPlan ViewZoom ExtentsAdd ViewRemove View

Zone060Volumetric PhysicsIO BrowserExperimentsEvent ListAnalysisTest Runner

AnimateViewPhysicsEmulationSimulation

View 1



Configuration - Nametag

ShowNametagTrue

Dimensions

Depth0.15 m

Height0.55 m

Thickness0.01 m

Width0.4 m

General

NameZone 060

Name

A non-unique name used when defining actions for this object.

Properties: ZcConnectEvents: ZoIK

Test Runner

Emulate3D TestRunnerCICD Example TestsC:\Users\ADeeble\OneDrive - Rockwell Automation, Inc\Emulate3D\Projects\E3D 2025 Feature Videos\Test Runner for Emulation


Zone 060

NEW TEST SCENARIORUN ALLSTOP

Name	Configuration	Result
Stop Turntable	Action	Not Run
Stop After Turntable	Action	Not Run
Stop on transfer to 070	Action	Not Run
Stop on cam lift table	Action	Not Run

FPS: 7.61Speed: 1.00 / 1.00Size: 398x565Movie DisplayWorld LocationX: -7.338 mY: 1.569 mZ: -15.093 m

Reset Layout

 **EMULATE3D**  
by ROCKWELL AUTOMATION

Test Runner for Emulation - Running Test Groups





# Test Runner with Fault Framework

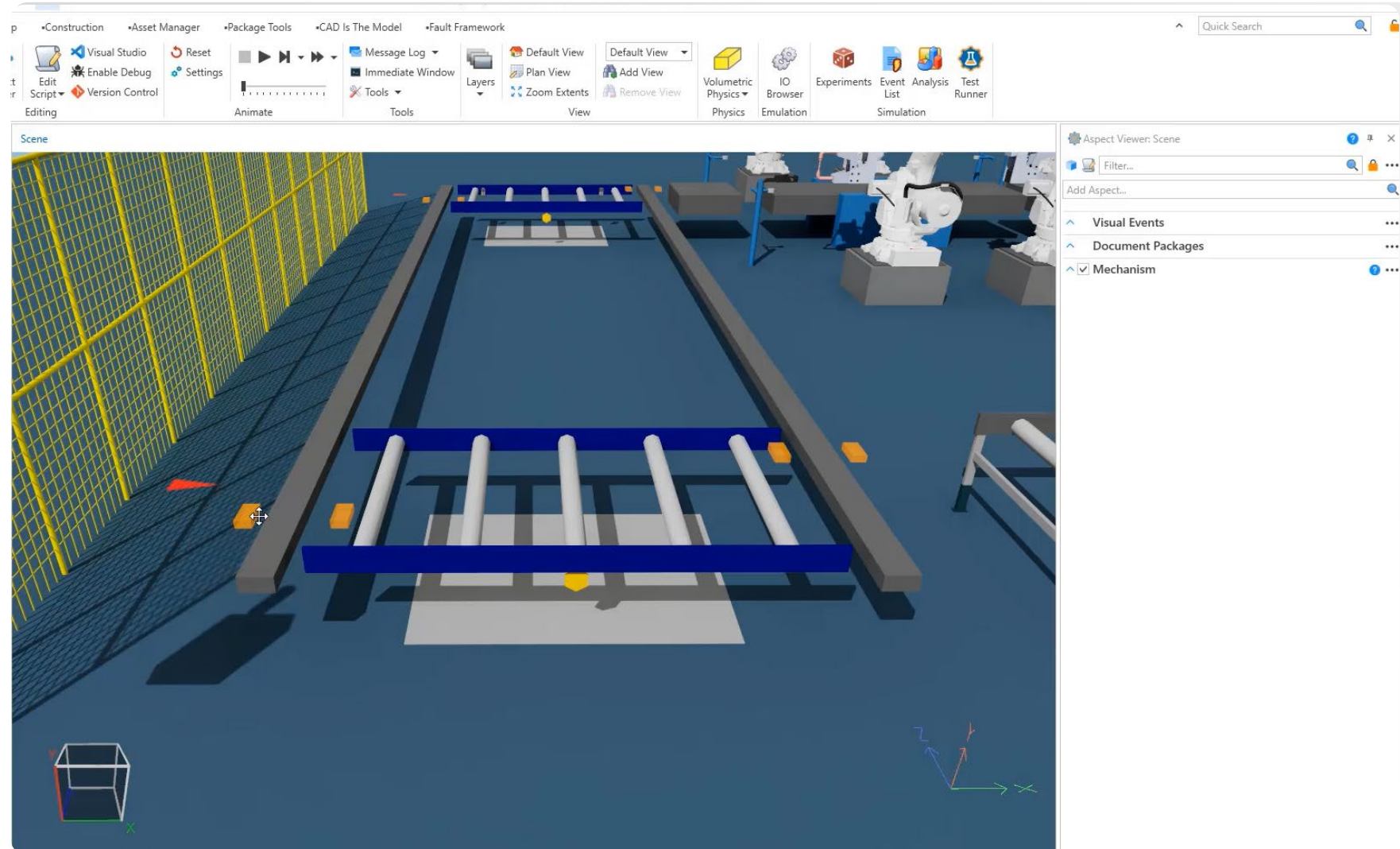
Simulate device faults using the Fault Framework aspects, from within Test Runner

Easily trigger faults from the Test Runner.

Tests adhere to Fault Framework assertions, or to test actions.

Also applicable to Operator Training

*Coming Soon – Dedicated Fault action with autofill*







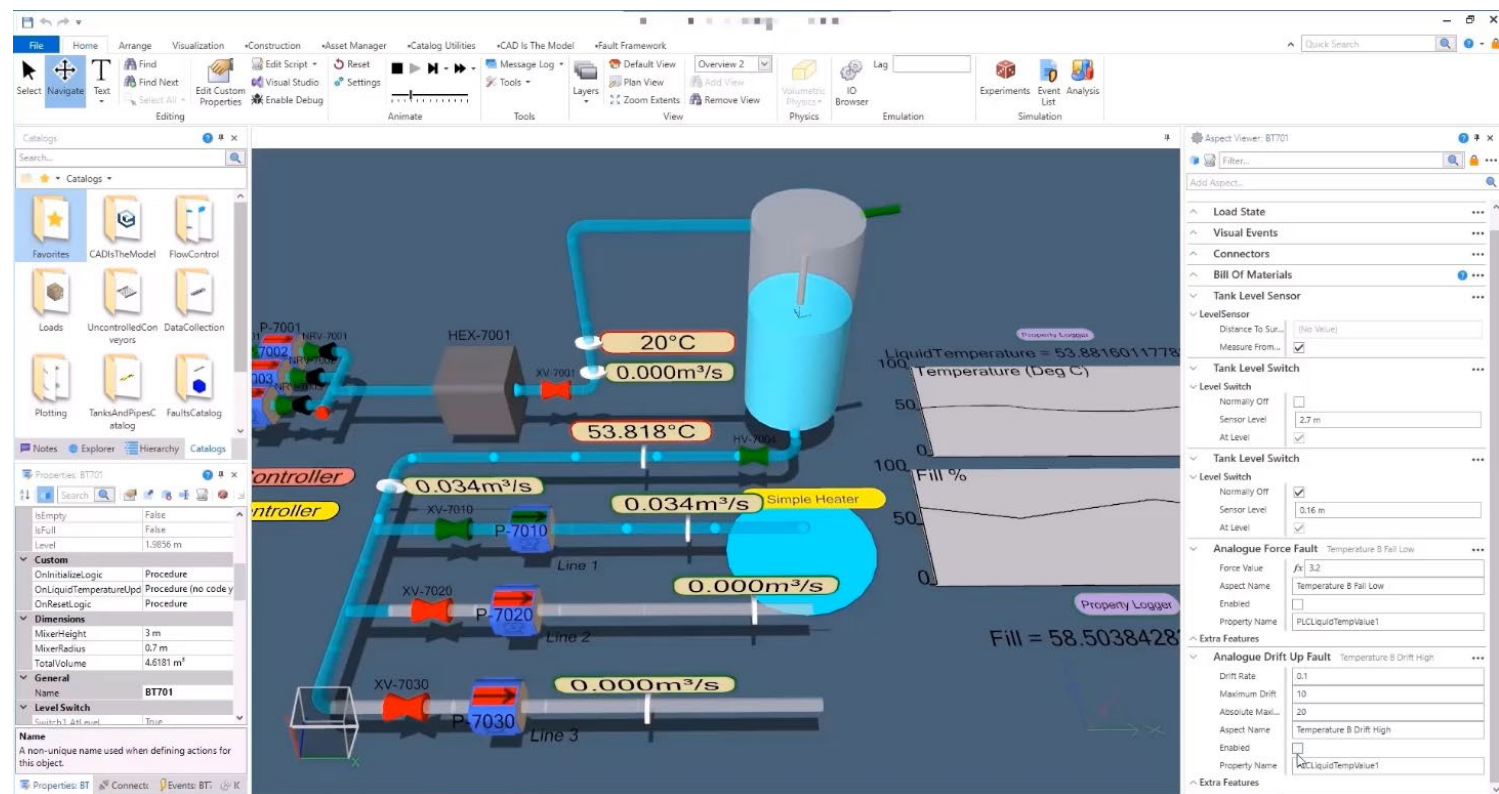
# Operator Training

Fault Framework and Test Runner enables Operator Training workflows

Create operator tests with actions and device faults for operators to diagnose

Use assertions to see if the operator correctly resolved the challenge

Automatic log of all interactions, ability for operator to log notes



Faults  
 Assertions  
 Advanced Controllers  
Aspects

New Schedule Name

Current Schedule ▾

Run Mode - None ▾

Create New Schedule

Delete Current Schedule

Add All Faults To Schedule

Import Schedule

Open Current Schedule

Save Faults Report To Schedule

Min Time

Max Time

Start

Step

Schedule State: not started.

Disable All Assertions

Disable All Faults

Enable All Assertions

Enabled Faults Count

Log Message

Add Message To Table

Operator Training Mode

Execution

# Test Runner is also for Catalog Development...

... but you'll hear about that in another presentation!

Use actions and assertions for simple tests

NUnit support in custom scripting enables advanced testing

CI/CD for custom catalog development:

- Version control with integrated GIT UI
- Run automated tests with Test Runner
- Deploy to custom package feed

Let's first see GIT version control in detail!

```
[TestFixture]
public class MyTests : Emulate3DTestFixture {
    [Emulate3DTest]
    public async Task Pass() {
        await Task.Delay(2000);
        Assert.Pass();
    }

    [Emulate3DTest]
    public async Task Fail() {
        await Task.Delay(2000);
        Assert.Fail();
    }

    [Emulate3DTest]
    public async Task ConveyorTest() {
        var conv = App.Document.FindVisual("SB1");
        Assert.IsNotNull(conv);
        var lc = conv.FindChild("LoadCreator1");
        Assert.IsNotNull(lc);
        if (lc.Props is not LoadCreatorProperties props) {
            Assert.Fail("Unexpected type");
            return;
        }
        props.DelayRate = 20;
        props.CongestionZone = false;
        App.Invoke(App.Reset);
        RunActionAtTime(5, () => {
            var loadCount = App.Document.PhysicsEngine.CountLoads();
            Assert.That(loadCount, Is.GreaterThanOrEqualTo(5),
                "Not enough loads in the scene, is the LoadCreator1 release rate too low?");
            Assert.That(loadCount, Is.LessThanOrEqualTo(10),
                "Too many loads in the scene, is the LoadCreator1 release rate too high?");
        });
        await RunModelForSecondsAsync(5.1, 2);
    }
}
```

# Test Runner is used by the Emulate3D Developers...

...for catalog, regression, communications, and performance testing!

Emulate3D TestRunner

Emulate3D TestRunnerctqa

Model Actions

ADD ACTIONS

Name	Timeout (s)
SimplePropertyChange Actions	10
ConnectBuiltinMemory Actions	10
TestBuiltinMemory Actions	10
MemoryStruct Actions	50
TelegramsTCP Actions	15
Allen-Bradley StartUp Actions	0.1
AB PrimitiveDataTypes CIP3 Actions	35
AB RT CIP3 Actions	100
AB RT CIP1 Actions	100
AB RT MicroLogix PCCC Actions	100
AB MicroLogix Sawtooth Merge 1 Actions	95
AB MicroLogix Sawtooth Merge 2 Actions	95

Emulate3D TestRunner

Emulate3D TestRunnerctqa

Models

IMPORT

Name
AA_AA_SinglePropertyChange.demo3dx
AA_AB_ConnectBuiltinMemory.demo3dx
AA_AC_TestBuiltinMemory.demo3dx
AA_AD_Memory_Struct.demo3dx
AA_AE_RoundTrip_TcpTg.demo3dx
AB_AA_StartUp.demo3dx
AA_AB_ConnectBuiltinMemory.demo3dx
AA_AC_TestBuiltinMemory.demo3dx
AB_AC_PrimitiveDataTypes_CIP3.demo3dx
AB_AD_RT_CIP3.demo3dx
AB_AE_RT_CIP1.demo3dx
AB_AG_RT_MicrologixPCCC.demo3dx

Emulate3D TestRunner

Emulate3D TestRunnerctqa

CTQA Test Suite

NEW GROUP

CTQA Test Suite

NEW SCENARIO

RUN ALL

Name	Model	Actions	Result
SinglePropertyChange	AA_AA_SinglePropertyChange.demo3dx	SimplePropertyChange Actions	NotRun
ConnectBuiltinMemory	AA_AB_ConnectBuiltinMemory.demo3dx	ConnectBuiltinMemory Actions	NotRun
TestBuiltinMemory	AA_AC_TestBuiltinMemory.demo3dx	TestBuiltinMemory Actions	NotRun
MemoryStruct	AA_AD_Memory_Struct.demo3dx	MemoryStruct Actions	NotRun
TelegramsTCP	AA_AE_RoundTrip_TcpTg.demo3dx	TelegramsTCP Actions	NotRun
Allen-Bradley StartUp	AB_AA_StartUp.demo3dx	Allen-Bradley StartUp Actions	NotRun
AB PrimitiveDataTypes CIP3	AB_AC_PrimitiveDataTypes_CIP3.demo3dx	AB PrimitiveDataTypes CIP3 Actions	NotRun
AB RT CIP3	AB_AD_RT_CIP3.demo3dx	AB RT CIP3 Actions	NotRun
AB RT CIP1	AB_AE_RT_CIP1.demo3dx	AB RT CIP1 Actions	NotRun
AB RT MicroLogix PCCC	AB_AG_RT_MicrologixPCCC.demo3dx	AB RT MicroLogix PCCC Actions	NotRun



# Thank you

[www.rockwellautomation.com](http://www.rockwellautomation.com)





