MATLAB EXPO 2019

Industrial IoT and Digital Twins

Pallavi Kar
Sr Application Engineer
Data Science & Enterprise Integration
Digital Twin - Mode for Digital Transformation

Customer Goals

By connecting machines in operation, you can use data, algorithms, and models to make better decisions, improve processes, reduce cost, improve customer experience.
Transpower Ensures Reliability of New Zealand National Grid with Reserve Management Tool

“We record frequencies on the grid, inject them into our Simulink model, and compare the simulation results to the actual system response. With Simulink we can continually calibrate and improve our model, and ultimately improve the accuracy of our reserve estimates.”

— Heidi Heath, Transpower

Transmission lines near Transpower’s Benmore substation.

Challenge
Calculate the amount of reserve power needed to ensure that New Zealand’s national grid can continue to operate if a generator fails.

Solution
Use Simulink to run simulations of the entire grid, including generators, loads, and HVDC links, every 30 minutes.

Results
• Critical updates rapidly implemented
• Simulations verified using real data
• Updates made in-house
Transpower - Building Reserve Management Tool using Digital Twins

**Objective**: Always have enough reserve energy

**Digital Twin**:  
- Simulink model of entire grid and tune parameters  
- Simulate 100s future scenarios to predict maximum energy needed

**Outcome**: Optimize & provided operators control setpoints for sufficient energy reserves

“We record frequencies on the grid, inject them into our Simulink model, and compare the simulation results to the actual system response. With Simulink we can continually calibrate and improve our model, and ultimately improve the accuracy of our reserve estimates.”  
— Heidi Heath, Transpower
Operationalizing Digital Twin with Industrial IoT infrastructure

Smart assets: Data Ingestion to Local Communications

Edge systems: Edge Management to Long-Range Communications

OT Infrastructure: Integration

IT Systems

Value of data to decision making:

- **Milliseconds**: Hard real-time control
  - C/C++
  - MODBUS TCP/IP

- **Seconds**: Real-time decisions
  - Model-Based Design
  - Code generation

- **Minutes**: Time-sensitive decisions
  - Stream Processing
  - MATLAB Production Server

- **Hours**: Big Data processing on historical data
  - Hadoop/Spark integration
  - MDCS, Compiler

Software tools and technologies:

- Hadoop/Spark integration
- MDCS, Compiler
- Edge Processing
- Model-Based Design
- Code generation
- MATLAB & Simulink
- MODBUS TCP/IP
- Stream Processing
- MATLAB Production Server
- MODBUS TCP/IP
- Kinesis Event Hub
- Amazon Web Services
- Azure
- Docker
- Spakr
- TCP/IP
Challenges in building Digital Twins & related applications:

– Building Digital Twins from scratch: Physics based or Data based statistical Models

– Keeping Digital Twins updated – Tuning Models & AI Algorithms with new data

– Scaling number of Digital Twins to match the number of assets

– Deploy Digital Twin Models & Algorithms across the IIoT ecosystem
Digital Twin Example: Motorized Pump Demo Hardware

Hydraulics

- Solenoid Valve
- Pressure Sensor
- Reservoir
- Pump1
- Pump2
- Motorized Valve

MATLAB EXPO 2019
Digital Twin Example: Motorized Pump Demo Hardware

MATLAB EXPO 2019
Digital Twin Example

Condition Monitoring & Parameter Estimation

- Monitor
- Analyze
- Predict
- Control
- Optimize

Digital Twin Demo

Control Panel

<table>
<thead>
<tr>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>EV3</td>
<td>0</td>
</tr>
<tr>
<td>MBV</td>
<td>70</td>
</tr>
<tr>
<td>RPM</td>
<td>1296</td>
</tr>
<tr>
<td>1298</td>
<td></td>
</tr>
<tr>
<td>PumpCHOICE</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>PT2</td>
<td>24.50</td>
</tr>
</tbody>
</table>

Write Data: MOTOR, RESET
Time: 105

Fault Classification
- Normal
- Leakage
- Blockage

Parameter Tuning
- Estimate
- EV3
- MBV
- 0
- 66.5187

Physics based Model

Data based Model
Acquire Real-Time Data for Updating Digital Twin

Monitor Analyze & Update Predict Control Optimize

Pump Hardware
MODBUS TCP/IP

Digital Twin

Acquire Real-Time Data for Updating Digital Twin

MODBUS TCP/IP

Digital Twin

m = modbus('tcpip', '192.168.2.1', 308)
m =

Modbus TCP/IP with properties:

DeviceAddress: '192.168.2.1'
Port: 308
Status: 'open'
NumRetries: 1
Timeout: 10 (seconds)
ByteOrder: 'big-endian'
WordOrder: 'big-endian'
Creating Multi-Domain Physical Models using Simscape

Monitor  Analyze & Update  Predict  Control  Optimize

Pump Hardware
Simscape: Multidomain Modeling and Simulation platform

**Built-in faults**

- Seal Leak
- Off (Off)
- On (On)

**Parameters**

- Revolute Joint: Crank Bearing
- Damping Coefficient: bearing_visc_frict

**Variants**

**Custom**

MATLAB EXPO 2019
Use Simulink Design Optimizer to Parameterize Pump Model

- Setup Experiments
- Parameterize
- Save Sessions
- Generate Code
Parameter Estimation – Behind the scenes

Monitor → Analyze → Predict → Control → Optimize

MATLAB EXPO 2019

% Group the model parameters and initial states to be estimated together.
% v = [p; s];

% Estimation Function
estFcn = @(v) sdoPumpEstimation_Objective(v, Simulator, Exp);

% Optimization options
opt = sdo.OptimizeOptions;
opt.Method = 'lsqnonlin';

% Estimate the Parameters
vOpt = sdo.optimize(estFcn, v, opt)

Initialize → Set Objective → Estimate
Digital Twin Example: Estimate Model Parameters to match System

MATLAB Standalone App
1. Communicating with Hardware
2. Reading Pressure Values
3. Writing Valve Setting
4. Identify Fault conditions
5. Estimating Model Parameters to match the System

Model based Digital Twin
Workflow for developing data & AI based digital twins

Monitor  Analyze & Update  Predict  Control  Optimize

Label Faults

<table>
<thead>
<tr>
<th>Time</th>
<th>1 LeakFault</th>
<th>2 BlockingFault</th>
<th>3 BearingFault</th>
<th>4 FaultType</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 sec</td>
<td>2.8472</td>
<td>-0.1477</td>
<td>1.8000</td>
<td>All</td>
</tr>
<tr>
<td>0.001 sec</td>
<td>-0.1498</td>
<td>-0.4207</td>
<td>1.3103</td>
<td>Bearing &amp; Blocking</td>
</tr>
<tr>
<td>0.002 sec</td>
<td>0.6511</td>
<td>1.6521</td>
<td>-0.5557</td>
<td>Leaking</td>
</tr>
<tr>
<td>0.003 sec</td>
<td>0.1469</td>
<td>-0.2775</td>
<td>1.0074</td>
<td>All</td>
</tr>
<tr>
<td>0.004 sec</td>
<td>-0.6480</td>
<td>0.7065</td>
<td>-0.0878</td>
<td>Blocking</td>
</tr>
<tr>
<td>0.005 sec</td>
<td>-0.8165</td>
<td>-0.5434</td>
<td>-0.3079</td>
<td>Blocking</td>
</tr>
<tr>
<td>0.006 sec</td>
<td>-1.0061</td>
<td>1.2083</td>
<td>0.0661</td>
<td>Bearing</td>
</tr>
<tr>
<td>0.007 sec</td>
<td>1.0125</td>
<td>-1.9098</td>
<td>-0.7027</td>
<td>Leak &amp; Blocking</td>
</tr>
</tbody>
</table>

Represent Signals

Train Model

Validate Model
Failure Scenario Generation - Run Parallel Simulations to scale up
Video showing App in action
Condition Monitoring: Develop AI based models
Off-the-shelf Remaining Useful Life (RUL) estimators

Similarity Models

Degradation Models

System Data

Known failure threshold

Life time data with or without covariates

Run-to-failure history

Similarity Models

Degradation Models

Survival Models

Test Engine 8

Remaining Useful Life Prediction (cycles)

Remaining Useful Life (cycles)

K=50 Nearest Training Engine Trajectories

Engine Health indicator

PDF of Training Engine RUL

RUL: 459 hours
(95%CI: 374-558 hours)
Challenges in building Digital Twins & related applications:

✓ Building Digital Twins from scratch: Physics based or Data based statistical Models

✓ Keeping Digital Twins Updated – Tuning Models & AI Algorithms with new data

➢ Deploy Digital Twin Models & Algorithms across the IIoT ecosystem

➢ Scaling number of Digital Twins to match the number of assets
Operationalizing Analytics across IIoT infrastructure

- **Smart assets**
  - Data Ingestion
  - Local Communications

- **Edge systems**
  - Edge Processing
  - Model-Based Design, code generation

- **OT Infrastructure**
  - Long-Range Communications
  - Edge Management

- **IT Systems**
  - Integration

---

- **Value of data to decision making**
  - Hard real-time control
    - C/C++ MODBUS TCP/IP
  - Real-time decisions
    - Edge Processing Model-Based Design, code generation
  - Time-sensitive decisions
    - Stream Processing with MATLAB Production Server
  - Big Data processing on historical data
    - Hadoop/Spark integration with MDCS, Compiler

---

- **Scopes**
  - Speed
    - Milliseconds
  - Seconds
  - Minutes
  - Hours
  - Days
  - Months
Operationalizing on Edge

- Low Compute
- Near range Communication

- Higher Compute
- Both Near & Far Communication

- Model-Based Design with MATLAB & Simulink, code generation
- Edge Processing with Model-Based Design, code generation
- Stream Processing with MATLAB Production Server

Value of data to decision making

- Hard real-time control
- Real-time decisions
- Time-sensitive decisions

C/C++ MODBUS TCP/IP

Speed

MATLAB EXPO 2019
Deploying Analytics on the Edge

Use MATLAB Coder to generate C code.
Running MATLAB on Edge and streaming processed data

Real-time decisions

Edge Processing Model-Based Design, code generation

Time-sensitive decisions

Stream Processing with MATLAB Production Server

librdkafka: https://github.com/edenhill/librdkafka
jansson: http://www.digip.org/jansson/
Operationalizing Analytics across IIoT infrastructure

- Smart assets: Data Ingestion → Local Communications
- Edge systems: Edge Processing Model-Based Design, code generation
- OT Infrastructure: Long-Range Communications → Edge Management
- IT Systems: Integration

Value of data to decision making:
- Hard real-time control: C/C++, MODBUS TCP/IP
- Real-time decisions: Edge Processing Model-Based Design, code generation
- Time-sensitive decisions: Stream Processing with MATLAB Production Server
- Big Data processing on historical data: Hadoop/Spark integration with MDCS, Compiler

Scope:
- Speed: Milliseconds, Seconds, Minutes, Hours, Days, Months
- Modbus: TCP/IP, Docker
- Edge Processing: Model-Based Design, code generation
- Stream Processing: with MATLAB Production Server
- Hard real-time control: C/C++, MODBUS TCP/IP
- Real-time decisions: Edge Processing Model-Based Design, code generation
Stream based Analytics deployed using MATLAB Production Server

Asset
Generate telemetry

Edge
Process & Stream

Production System

Apache Kafka

MATLAB Production Server
Worker processes
Request Broker

State Persistence

Azure

Generate telemetry

Asset

MATLAB EXPO 2019
Scaling batch operations with MATLAB Parallel Server

Run parallel threads of Digital Twins in batches

Apache Kafka
Connector

MATLAB Production Server

Worker processes

Request Broker

Time-sensitive decisions

Big Data processing on historical data

Scope

Apache
Kafka

Kafka
Event Hub

Scope

100 days

120 days

200 days

Motor Systems

GearPump

Motor Systems

GearPump

Motor Systems

GearPump

Motor Systems

GearPump

Motor Systems

GearPump

Motor Systems

GearPump

Motor Systems

GearPump
MathWorks Reference Architectures

mathworks.github.io

https://mathworks.com/cloud  Verified

matlab-aws-s3
MATLAB interface for AWS S3.
- MATLAB  Updated 26 days ago

matlab-azure-blob
MATLAB interface for Windows Azure Blob Storage.
- MATLAB  Updated on Feb 21

matlab-azure-data-lake
MATLAB Interface for Azure Data Lake.
- MATLAB  Updated on Feb 21

matlab-aws-common
Code common to MATLAB interfaces. Code in this repository is used as a dependency for other projects such as matlab-aws-s3.
- MATLAB  Updated on Feb 21

matlab-parquet
MATLAB Interface for Apache Parquet
- MATLAB  Starred 1  Updated on Dec 20, 2018

matlab-avro
MATLAB interface for Apache Avro files.
- MATLAB  Updated on Feb 9

---

### Databases
- Cassandra
- Microsoft SQL Server
- MongoDB

### Cloud Storage
- Azure Blob
- Azure SQL
- Amazon S3

### Big Data / OT Platforms
- Cloudera
- Hortonworks
- OSIsoft PI System

---

### Streaming
- AWS Kinesis
- Azure IoT Hub

### OT Platforms
- OSIsoft PI System

---

### Dashboards
- Microsoft Power BI
- Tableau
- Qlik
- Spotfire
Summary

– With MATLAB you can **read hardware data** over various protocols & DAQ systems

– With **Physical Modeling blocks & AI libraries** in MATLAB you can now build Digital Representations of your asset

– You can **tune physical models** using Simulink design optimization & **RUL models with update methods**

– With **deployment** abilities in MATLAB you can **operationalize across edge and IT/OT infrastructure**
Call to Action

Digital Twin & Streaming Analytics

➢ Building IoT solutions
➢ Developing and Deploying on Cloud
➢ Build Digital Twins with Physical Modeling workflow
➢ Learn: How to build Predictive Maintenance Applications?
➢ Learn Data Science with MATLAB

References

Attend Trainings

MATLAB EXPO 2019
Q&A
MATLAB EXPO 2019

Email: Pallavi.Kar@mathworks.com
LinkedIn: https://www.linkedin.com/in/pallavi-kar-2a591518/
Twitter: @PallaviKar2512