A Digital Twin Approach to Online Monitoring in Industrial Internet of Things Applications

Akshay Rajhans and Dan Lluch
MathWorks
1 Lakeside Campus Drive, Natick, MA 01760.
{arajhans, dlluch}@mathworks.com

I. Introduction

Cyber-physical systems (CPS) deploy interconnected computational, or cyber, elements to sense and control a physical environment. Given the complexity of software functionality that gets implemented using these computational elements, model-based design (MBD) approaches are often used to develop and deploy such systems. The use of computational models in this traditional MBD workflow at design time (*before deployment*) has been extensively studied in the literature.

With the introduction of internet-of-things (IoT) applications, there is usually an internet-enabled physical *thing*, or a *node*, which collects some data from the physical environment in the form of a data stream. For internet-enabled embedded sensors applications, depending on the available compute power it is possible do perform some simple computation, often referred to as *edge computing*. An example includes counting the number of cars seen on a highway using a USB webcam and a Raspberry Pi [3]. For additional compute-intensive tasks, such as analyzing historical traffic flow patterns, the data stream is often sent to the cloud for further processing in a *cloud computing* environment.

In contrast with embedded sensing applications, in case of industrial IoT systems in the smart manufacturing domain, the physical nodes could be, for example, various manufacturing machinery in an industrial plant, sometimes referred to as a *physical asset*. For these types of applications, the physical node is itself a CPS with safety-critical real-time performance requirements, and non-trivial amount of compute power may be available at the disposal. Figure 1 presents a typical connectivity architecture. The smart physical asset (CPS) itself handles performs computations that often require hard real-time guarantees, such as control tasks; edge computing systems close to the physical asset for time-sensitive online computations of the order of seconds, such as fault detection and isolation; operation technology (OT) infrastructure for computations of the order of minutes and hours, such as coordinating the overall operation of of a plant; and information technology (IT) systems enabled by the cloud for computations over the days and months, such as business analytics.

In this abstract, we focus on edge computing where, in contrast to traditional MBD for CPS, the use of computational models is increasingly becoming useful at operation time (*after deployment*) for model-based online monitoring and analytics. Examples of such computations, in addition to fault detection and isolation as stated above, include prognostics and health monitoring, and other analytics such as remaining useful life identification. This workflow usually makes use of a computational model—called a *digital twin*—of the physical asset for online monitoring. We consider a hardware and software based demonstrator to showcase various workflows in this domain.

II. Overall Architecture

Figure 2 shows a high-level overview of the overall architecture considered in this abstract. Gear pumps form the physical model considered in this workflow. These pumps have a motor controlled by a programmable logic controller (PLC). The pump along with the PLC forms the smart asset. Additional computational machinery next to the PLC, such as a desktop computer, a real-time hardware such as the Speedgoat target machines[1], or other edge devices such as HPE Edgeline 1000[2] provide the edge computing functionality nearby the asset. The data gets fed to a cloud service provider, such as Microsoft Azure cloud and Amazon Web Services cloud. A remote client computer that is connected to this cloud can be used for visualization of the data as well as the analytics results.

III. Digital Twin Modeling

A *digital twin* is a computational model of a real physical asset in operation. Such a model is used in-operation to control and optimize behavior of the particular real asset by identifying anomalies, efficiencies, or the possibility of particular future events. There are two approaches to developing such a computational model of the pump.

A. Modeling From First Principles

First-principles based modeling approaches involve creating a physical model based on the laws of physics, an approach often called white-box modeling, e.g., of the gear pump dynamics [2]. We create such a first-principles based model of the pump in Simscape™, an acausal physical modeling formalism which provides a natural mechanism for creating such physics-based models.[3]

[1]https://www.speedgoat.com/products-services/real-time-target-machines
[2]https://www.hpe.com/us/en/product-catalog/servers/edgeline-systems.hits-12.html
[3]https://www.mathworks.com/products/simscape.html

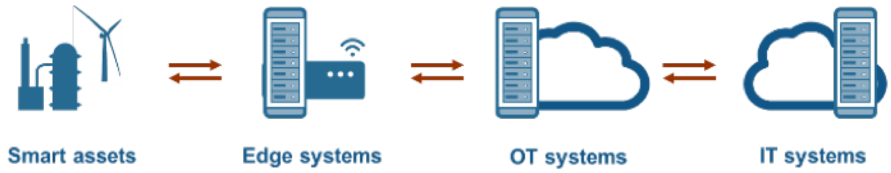

Fig. 1. Computation and Connectivity Structure in a Typical Industrial Internet of Things Application.

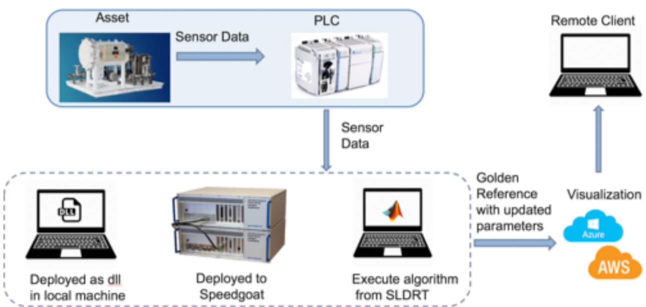

Fig. 2. Overall architecture of the demo.

B. Data Driven Modeling

At the other end of the spectrum are data-driven—or black-box—modeling approaches, based on machine learning or system identification techniques to learn or identify models from measured data about the physical asset in operation [1]. Textual modeling languages such as MATLAB® are well suited for modeling such data-driven models.

Our goal is to develop both physics-based and data-driven digital twin models of the pump to show the efficacy of the approaches. Supervisory control operational in the PLC can be modeled using a combination of Simulink® and Stateflow®, tools for block diagram modeling and statechart modeling respectively.

IV. Simulation and Analysis

There are two main modes in which digital models are simulated on the edge.

A. In Real Time: Synchronized with the Wall Clock

Real-time simulation of the digital twin model synchronized with the wall clock is useful simulating the physical asset 'in parallel' while in operation, for monitoring applications such as anomaly detection and fault-detection and isolation.

B. As Fast As Possible

Simulations that are run as fast as possible, but not necessarily synchronized with the clock, are useful for several scenarios including the following.

- Running the so called 'what if' scenarios where the digital twin is simulated in a scenario forward in time before putting the physical twin in the operating condition.
- Running several simulations in parallel in a batch mode to generate synthetic simulation data. Such simulation data is useful for further online learning, e.g., learning of set points and parameters, and reinforcement learning.

C. Simulation setup

The demo under development showcases three key workflows for streaming realistic simulation streaming into Azure IoT Hub Devices.

- The `parsim` command[4] in MATLAB for parallel simulation of Simulink models along with a MATLAB Distributed Computing Server[5] is used for the simulation functionality.
- A number of digital representations of pumps are set up to send data into Azure IoT Hub. Some pumps may be triggered to initiate failures, or be running in various states of age (new vs. old), etc.

D. Simulated Faults

Two kinds of faults, namely leakage faults and blockage faults, can be emulated on the physical pump. These fault conditions can be detected via monitoring a discrepancy between the simulated behavior of the digital twin and observed behavior of the physical twin.

V. Discussion

The overall goal of this demo is to develop a hardware- and software-based demonstration of various digital twin workflows from the industrial IoT application domain with an objective of simulating realistic scenarios, fault conditions, and realistic streaming data. Exploring the use of formal specification and monitoring approaches for anomaly detection and fault detection and isolation using temporal logics [4] would be an interesting future work.

Acknowledgments

We thank our colleagues Sarah Dagen, Heather Gorr, Ramanuja Jagannathan, Pallavi Kar, Pieter Mosterman, Adarsh Narasimhamurthy, Jim Tung, and Terri Xiao for their help in developing this demo.

References

[1] Black box modeling. https://www.mathworks.com/help/ident/ug/black-box-modeling.html.

[2] Massimo Rundo. Models for flow rate simulation in gear pumps: A review. *Energies*, 10(9), 2017.

[3] Hans Scharler. Counting cars and analyzing traffic, 2015. https://blogs.mathworks.com/iot/2015/09/18/counting-cars-and-analyzing-traffic-with-thingspeak/.

[4] Zhe Xu, Sayan Saha, and Agung Julius. Provably correct design of observations for fault detection with privacy preservation. In *Proceedings of the 2017 IEEE 56th International Conference on Decision and Control (CDC)*, December 2017.

[4] https://www.mathworks.com/help/simulink/slref/parsim.html

[5] https://www.mathworks.com/products/distriben.html