Integrating Streamed Sensor Data into a Distributed Model of a Complex System

A report submitted in partial fulfilment of the requirements for the degree of

Bachelor of Engineering (Hons) in Software Engineering

at

the University of Waikato

Bert Downs

Supervised by Tim Walmsley, Mark Apperley

October 2024

Abstract

Digital twins combine live factory data, historical state, and a model of the factory to predict future states. This project shows how the process of creating a digital twin lacks standardization. A abstract framework for creating digital twins of chemical plants is presented, that builds on top of existing chemical simulation and data processing tools. A prototype is created to assess the feasibility of the framework. Functionality is implemented for steady-state process models within the Ahuora Digital Twin Platform, and evaluated in line with the framework presented.

Acknowledgements

I would like to thank my supervisors, Tim Walmsley and Mark Apperley, for their guidance and support throughout this project. I would also like to thank the Ahuora project team for their assistance and feedback. Finally, I would like to thank my family and friends for their encouragement and support.

Contents

List of Figures

List of Tables

List of Abbreviations

IDAES-PSE	Institute for the Design of Advanced Energy Systems - Process Simulation Environment
SCADA	Supervisory Data Aquisition and Control
PYOMO	PYthon Optimisation MOdelling language
PDE	Partial Differential Equation
RBF	Radial Basis Function
PYSMO	Python-based Surrogate Modelling Objects
OMLT	Optimization and Machine Learning Toolkit

Chapter 1

Introduction

1.1 Motivation

The World Economic Forum ranked "Failure to mitigate climate change" as the number one threat to the world in the next ten years [1], due to the effects climate change has on extreme weather events, biodiversity, and climate-vulnerable economies. Decarbonisation is a crucial step in mitigating climate change. In New Zealand, process heat accounts for 8% of total greenhouse emissions [2].

Digital Twin technology shows promise in assisting decarbonisation, through efficient energy management [3]. A Digital Twin is a virtual model of a physical object or system that is connected to the real-world object or system. It can be used to monitor, control, and optimise the real-world object or system. Digital Twins are used in a variety of industries, including manufacturing, healthcare, and transportation [4]. However, most Digital Twins in literature are designed for very specific use cases, and there is a lack of standardisation in the field.

Ahuora is a research group focused on developing smart energy systems to decarbonise factories. They have developed a Web-based simulation platform that allows users to create a digital twin of their factory. This platform is based on steady-state simulation, which simulates a factory at a single point in time. All factory conditions are manually specified by the user. This platform is useful for modelling changes to a factory before construction, or understanding the factory's performance under different conditions.

At the current stage of development, the Ahuora platform cannot be considered a "Digital Twin" because it does not take into account the factory's real-time state. By integrating real-time sensor data into the simulation, the platform can monitor the factories' performance, and suggest tunings that will optimise resource efficiency. The data can also be used to predict and avoid failures and downtime, a key problem where many resources are wasted. Additionally, models created in the Ahuora Simulation Platform during the design phase could also be used during operation, minimising overhead costs.

Including real-time data in the simulation is needed to improve the usefulness of the Ahuora Platform in industry. The system needs to meet industry requirements for security, scalability, and reliability. As such, this project has been commissioned to develop a standardised framework to enable the integration of real-time sensor data into the Ahuora Digital Twin Platform.

1.2 Objectives

The objectives of this project are as follows:

- Conduct an Exploratory Analysis of techniques and tools for digital twin development, based on their applicability to the Ahuora Digital Twin Platform and the requirements of live data processing.
- Develop the Ahuora Digital Twin platform to a stage where support for live data processing can be added.
- Develop a standardised framework for integrating real-time sensor data into the Ahuora Digital Twin Platform.
- Develop a prototype implementation of the framework.
- Evaluate the prototype implementation in a case study.
- Identify areas for future work.

1.3 Scope

Full integration of real-time sensor data into the Ahuora Digital Twin Platform is out of scope for this project. This project will focus on identifying techniques and tools for simulation and modelling that will be needed in a industry setting, and developing a prototype live data processing system for Ahuora that is extensible enough to support those techniques and tools.

1.4 Literature Review

This section is a summary of the full Literature Review included in Appendix C.

1.4.1 Data Collection in Industrial Facilities

Real-time data is used extensively for monitoring and improving operational efficiency, particularly for schedule optimisation. It shows promise for more advanced diagnostics, fault classification, and control optimisation [5]. Real-time data is a crucial element of the emerging field of Digital Twins because it enables the virtual model to be aware of changing conditions in the real world.

Crucial factory tasks such as diagnostics, control, and optimisation must be peformed in real time. Software produced in this project must be reliable. The software must be scalable enough to work in large and small industrial settings and needs to provide interoperability with existing systems that perform some of these roles already.

1.4.2 Data Processing Pipeline

In a factory, raw data is collected from SCADA systems, IoT networks, and system logs. This data is organised and labelled, tagging it with relevant contextual information such as timestamps and source information. Sensor data from multiple sources can then be fused together to provide more meaningful, understandable interpretations, cotributing to an increase in accuracy and fault tolerance. These interpretations serve as a bridge between raw data and model simulation. Simulation is used to estimate other parameters of the system that may not be able to be measured directly by sensors, enabling a comprehensive view of the entire industrial process's state. Finally, manually defined or data-driven algorithms can be used to respond to the process's state, to optimise schedule, control, or maintenance tasks. This pipeline is summarised in Figure 1.1.

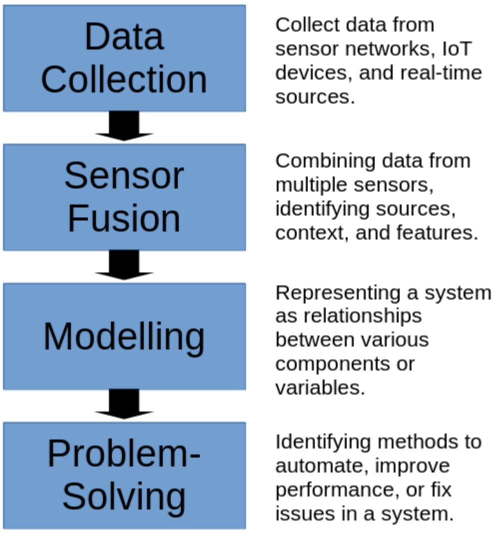

Figure 1.1: Pipeline for processing Live Data in Digital Twin Systems, as identified in the Literature Review (Appendix C)

This project's aim is to ingest live data from an industrial process and use a mathematical model to estimate other parameters in real-time. It focuses on the sensor fusion and model simulation areas of the data processing pipeline. Hence, the inputs and outputs of the project can be clearly defined.

- Input: Historical & live data from lower-level Data Collection Software, e.g SCADA systems or IoT Networks.
- Output: Real-time simulation results, for higher-level optimisation, control and reporting tools.

The software will be responsible for transforming measurements from sensors into a comprehensive model of the factory.

1.4.3 Sensor Fusion and Hybrid Modelling

Advanced methods for sensor fusion, such as Bayesian Analysis, Fuzzy Logic, and Theory of Evidence models, can help convert data into a more interpretable format that is appropriate for use in an analytical or mathematical model. Furthermore, Hybrid modelling has been shown to enable more flexible modelling that still retains the advantages of mathematical modelling. It is better suited to the variation and noise that is inherent in real-world sensor measurements. Hybrid Modelling also enables the modelling of complex dynamics that are difficult to capture via mathematical modelling alone. Additionally, live modelling augments models with the ability to update themselves in real time, so that the model can reflect changes in the real-world environment.

This provides a basis for developing the core features required in the Ahuora Platform. The platform needs to incorporate existing or novel sensor fusion technologies to prepare data for modelling. The Ahuora Platform is built on the IDAES process simulation engine. Further research is required to understand how IDAES supports hybrid modelling, and how to make hybrid modelling feasible in a real-time context. The optimal technologies and methods to use for hybrid modelling in real-time is also uncertain, so further research is likewise required to investigate this. This research needs to be conducted within the context of the needs and requirements of the Ahuora Platform, to ensure that the approach also supports the broader goals of the Ahuora project.

Chapter 2

The Ahuora Digital Twin Platform

The work presented in this report is part of a larger, multi-disciplinary project. Consequently, some of the presented work goes beyond the limited scope of this immediate research, but is still required to achieve outcomes relating to the integration of live data analysis into the broader software platform. As such, this content is still relevant within the context of this specific project. Additionally, some work involves research for future implementations that cannot be completed with the platform's current capabilities. This chapter provides an overview of the Ahuora Digital Twin Platform, to provide context for the rest of the report.

2.1 Background

'Project Ahuora' is a Ministry of Business, Innovation and Employment (MBIE) funded project that aims to decarbonise the process heat sector. By decarbonising, New Zealands' greenhouse gas emissions will be reduced. Cost savings from reduced energy consumption are anticipated, along with increased energy independence. This is a multi-disciplinary project that involves researchers from the University of Waikato, University of Auckland, Massey University, and other global universities. Chemical Engineers bring understanding of the chemical processes that are used in industry. Electrical Engineers bring understanding of the grid system and how to integrate renewable energy sources. Mechanical engineers bring understanding of how to design and build more efficient systems. Software Engineers bring understanding of how to model, simulate, and monitor complex systems.

A key objective is to develop a digital twin platform for the chemical processing industry. This platform will allow New Zealand factory operators to model their processes, simulate different scenarios, and monitor process state in real-time. This will enable factory operators to make data-driven and scientifically backed decisions on how to improve their processes. A digital twin can recognise where the factory is underperforming, suggest real-time improvements, and help plain future investments.

2.2 The Ahuora Simulation Platform

A key deliverable of the Ahuora project to date is the Ahuora Simulation Platform. This is a web-based platform that allows users to model a factory or other energy system, and simulate its per-

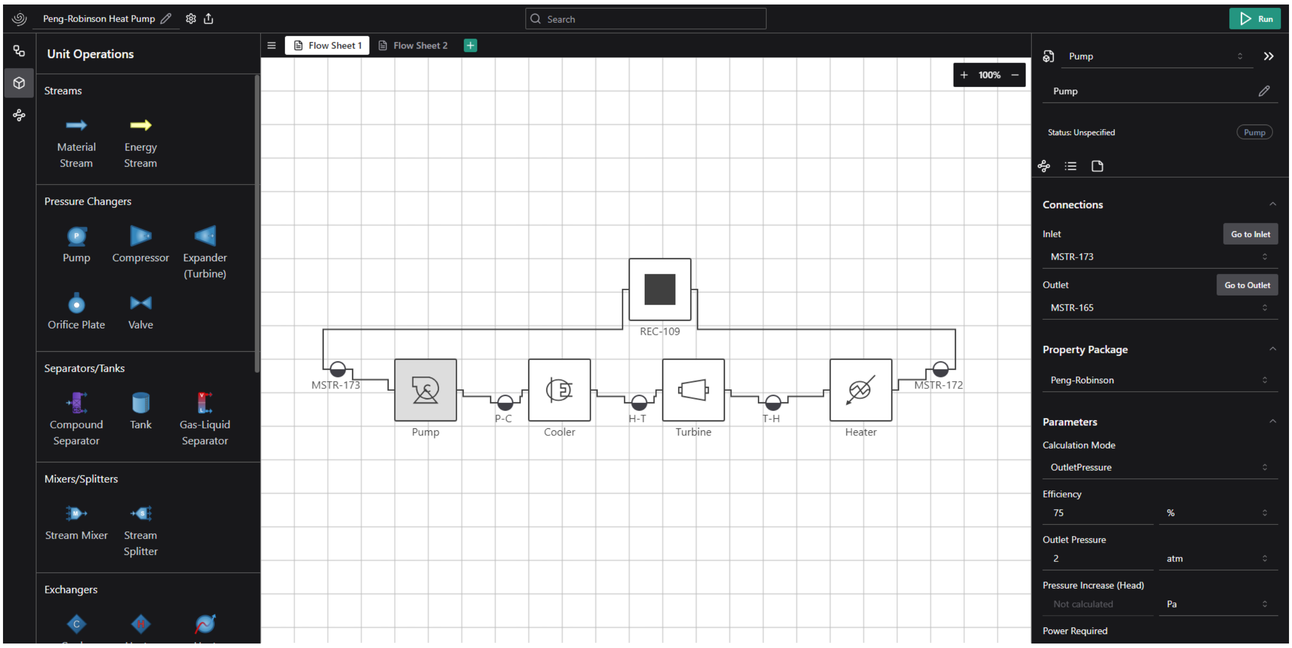

Figure 2.1: Example Flowsheet in the Ahuora Simulation Platform

formance. Much of the analysis functionality is achieved by leveragin the IDAES Process Systems Engineering framework within the backend. IDAES is a Python library that provides tools for modelling and simulating chemical processes.

Currently, the platform can model a factory at a single point in time. The user specifies the properties of the factory, such as the flow rates of different materials, the temperature and pressure of different streams, and the efficiency of different unit operations. The platform then simulates the factory and provides the user with a report on the factory's performance.

2.2.1 Flowsheet Interface

Figure 2.1 shows a screenshot of the Ahuora Simulation Platform, as at August 2024. The user interface is divided into three main sections. The left-hand panel shows a list of unit operations from a factory, including pumps, heaters, heat exchangers, reactors, and material streams. The user can drag and drop these unit operations onto the canvas in the centre of the screen. The user can then connect the unit operations together to create a process flow diagram. The right-hand panel shows the properties of the selected unit operation, such as the flow rate of the material stream, the temperature and pressure of the stream, and the efficiency of the unit operation. The user can edit these properties to simulate different scenarios.

The displayed flowsheet shows a simple heat pump cycle. The block on the top is a "recycle", specifying that the output of the cooler is fed back into the pump. A more complex flowsheet would replace the cooler and heater with heat exchangers, which exchange heat with their environment, but this provides a simple example.

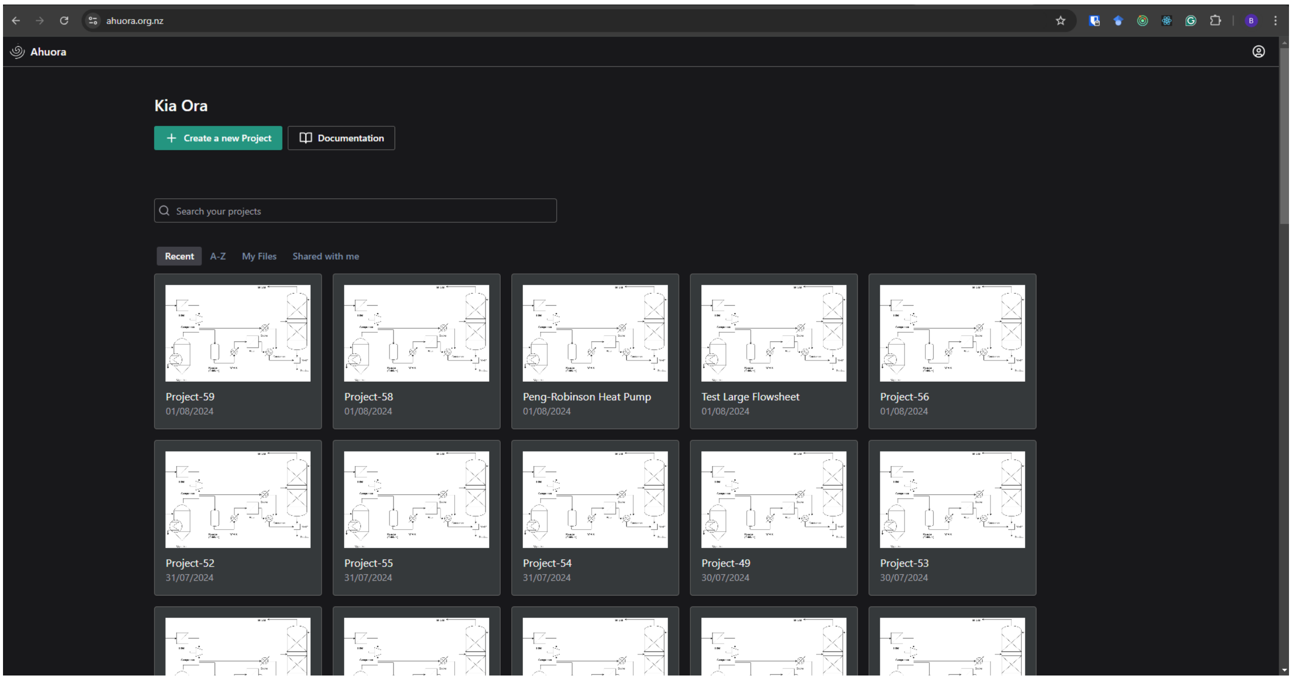

Figure 2.2: Homepage of The Ahuora Simulation Platform

2.2.2 Online Integration

The Ahuora Simulation Platform is designed as a web-based multi-user platform. This offloads processing and data storage to the server, allowing users to access the platform from any device with a web browser. Simulation can be very computationally expensive, particuarly in advanced models, so this is a key feature. It enables simulation to be run in parallel on powerful servers, as the simulation is not done within the local web browser, but rather via distributed server infrastructure. This allows the platform to be used in industry without requiring significant upfront investment in hardware.

In future, this will also enable the platform to be used for real-time collaboration between multiple users. Its API allows it to be integrated with other software, enabling enhanced functionality, real-time updates, and broader interactions.

The home page of the platform, shown in Figure 2.2, provides a list of saved simulations, and allows the user to create a new simulation. This is not publicly acessible, as the platform is still in development, and user account functionality is not yet complete.

2.3 Platform Architecture

The platform is hosted in a Kubernetes cluster, located on-site. The Kubernetes cluster handles all web traffic, deployments from the private Github Repository, and scaling of services.

Within the platform there are various containers running through the Docker platform, that can be replicated as required to scale based on service demand. The Database stores all flowsheets and model data. The solving interface uses the IDAES Process Simulation software [6], built on the Pyomo equation-oriented modelling language [7], to simulate a factory in real time. The User Interface is written in React Typescript, and uses the ShadCN UI Library and Redux Toolkit for state management

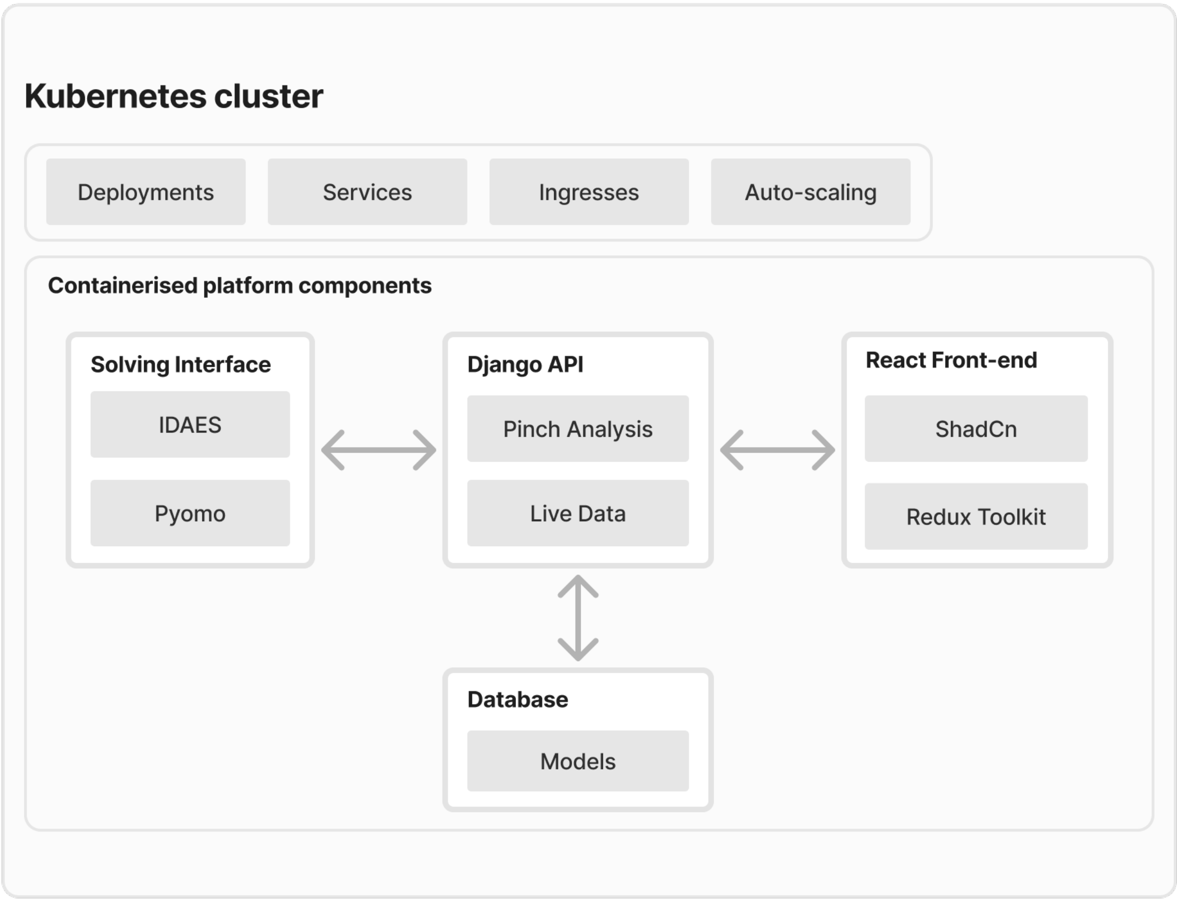

Figure 2.3: Architecture of the Ahuora Digital Twin Platform

and communication with the Django Backend. The Django API links all the services together, and handles the business logic for functionality such as Pinch Analysis and Live Data processing.

2.4 Ongoing Projects

Multiple other projects are being completed in parallel by other team members. Each project has a distinct topic, and is largely independent, but collaboration is required to ensure that each other's work is compatible.The projects cover a variety of topics, from analyis and data processing, to usability and deployment.

2.4.1 Live Data Processing

Team Member Responsible: Bert Downs

The Ahuora Platform currently supports building and simulating a factory, but it has no functionality to link live data from a factory into the Ahuora Platform. By integrating real-time sensor data into the simulation, the platform can monitor the factories' performance, and suggest tunings that will optimise resource efficiency. The data can also be used to predict and avoid failures and downtime, a key problem where many resources are wasted. Additionally, models created in the Ahuora Simulation Platform during the design phase could also be used during operation, minimising overhead costs.

2.4.2 Pinch Analysis

Team member responsible: Ethan MacLeod

One of the key optimisation tools the Ahuora Platform provides is the Pinch Analysis module. Pinch analysis is the process of identifying heat recovery pockets in a given system, and indicating where heat can be exchanged. The end goal of pinch analysis is reducing the overall heat consumption of a process, which in turn results in lower operational costs for a plant, and less greenhouse gas emissions produced. From the processes that are modelled on the Ahuora Digital Twin Platform, there is a distinct need for integration with process optimization tools like Pinch Analysis to inform the decision-making process for operators and engineers.

2.4.3 User Interface Usability

Team member responsible: Shean Danes Aton

This project aims to deliver an intuitive user interface and quality user experience in the Ahuora Digital Twin Platform through improving its UI through employing user centred methodologies including A/B testing, interviews, card sorting, the thinkAloud method, and surveys. Additionally, modern diagramming tools were reviewed and end users' knowledge were leveraged to the development of the Ahuora Simulation Platform UI. This project utilises software technologies and existing design components,ShadCn, for design implementations. Enhancing Ahuora Simulation Platform's UI increases user satisfaction, increases productivity, and minimises operational error.

2.4.4 Distributed Platform Deployment

Team member responsible: Caleb Archer

Each of the software components making up the platform need to be deployed together within a distributed environment that allows for efficient allocation and use of computational resources. This is achieved within the context of a Kubernetes cluster, on which several replicas of each platform component may run at any given time based on system load. The ability to scale workloads in response to demand increases the capacity of the platform to perform process modelling and simulation, and handle a greater number of users.

2.5 Other Long-term Platform Objectives

The projects listed pave the way for future work on the Ahuora Digital Twin Platform. This includes expanding the number and type of unit operations supported, and supporting a wider range of chemical processes. It is intended that the Ahuora Platform also support Hybrid modelling, where

Machine Learning Models are used to represent complex unit operations that do not have an exact mathematical representation. Analysis functionality to be added includes Dynamic Simulation, to model the factory over time and predict the effect of changes in the system, Process Variable Optimisation, to calculate optimal operating conditions, common diagramming and reporting functionality, and scheduling tools.

Because of the long-term context of the Ahuora project, work in this dissertation is constrained by the future requirements of the system. Care has been taken to ensure that the methods outlined can be extended to support the future objectives of the Ahuora Platform.

For clarity in this report, the term "Ahuora Digital Twin Platform" will refer to the complete solution, keeping in mind all these future objectives. The term "Ahuora Simulation Platform" will refer to the current state of the platform, which is based on steady-state simulation, and does not yet support the features required for a Digital Twin.

Chapter 3

Methodology

Software Engineering principles are notoriously hard to follow in academia [8], partially because of of competing criteria for success. In academia, the criteria is publication of results, but in Software Engineering, creating and publishing a quality piece of software is quintessential to success. Additionally, since this project is conducted as part of a wider effort to improve the Ahuora Platform, research and development often had to account for a changing environment. Thus Agile principles were followed for both research and development, focusing on implementing one feature at a time.

This project follows the "Action Research" methodology reviewed by Wohlin et al. [9]. The Action Research methodology is an iterative process, which includes five phases:

1. Diagnosing - Identifying a problem to be addressed. The larger problem has already been identified in the Introduction and Literature Review. Each chapter also starts by identifying specific research questions, or the purpose of implementing a feature.

2. Action Planning - Deciding what approach will best solve the problem. This is done by deciding on a prototype to build or a feature to implement, which is done at least once in each chapter.

3. Action Taking - This involves setting the planned actions into practice - by building a prototype for research purposes, or implementing a feature in the Ahuora Platform.

4. Evaluating - Studying and discussing the consequences of an action. The effectiveness of a prototype or feature, along with any other insights from the development process, is discussed at the conclusion of each chapter.

5. Specifying/learning - Identifying general findings related to the problem under study. This is done explicitly in some chapters, such as by identifying the requirements of different types of users. The report is concluded with a chapter specifically devoted to generalising the findings of each previous chapter into a theoretical framework that can be used for implementing Digital Twins, and an implementation plan for the Ahuora Platform.

Table 3.1: Overview of chapters

Chapter	**Purpose**
Data Collection (Chapter 4)	Understand the process of collecting data from a thermodynamic system, using a Heat Pump Dryer as a test case.
Simulation Technologies (Chapter 5)	Identify potential analysis techniques for a Digital Twin system, and what is required to make use of them.
Prototyping (Chapter 6)	Experiment with linking live data to a simple simulation in the Ahuora Platform.
Recording History (Chapter 7)	Adding the ability to store past solves in the Ahuora Digital Twin Platform.
Data Preprocessing (Section 7.4)	Functionality for parameterising solves, to make it easier to develop Digital Twins.

3.1 Overview of Work

Each chapter has a distinct objective, yet each piece of work contributes to the same high-level objectives. Table 3.1 summarises the specific focus of each chapter. Chapters 4 & 5 focus on understanding the core tools and platforms that relate to Digital Twin solutions, establishing a holistic long-term view of the problem. These chapters are more research focused.

Chapter 6 assesses the conclusions from the previous chapters, by building a prototype live data collection system. A theoretical framework is presented that can be used generally to inform Digital Twin development efforts.

Chapter 7 develop some features in the Ahuora Simulation Platform that implement characteristics discussed in the earlier chapters. The features discussed in this section are now integrated into the Ahuora Platform codebase.

A case study of a heat pump dryer was used throughout the report to evaluate the prototypes built and features developed. This was achieved by developing a model of a heat pump dryer in the Ahuora Simulation Platform, and integrating real-time sensor data into the model. The case study was used to evaluate the feasibility of the framework, and to identify areas for future work.

Chapter 4

Data Collection: The Heat Pump Dryer

4.1 Purpose

The literature review identified that the inputs to the Digital Twin Platform would be sensor data from the lower-level data collection software, and the outputs would be results from the simulations performed by the platform. To be able to test our processes, input data was required, so a Heat Pump Dryer was chosen as a data source.

A Heat Pump Dryer provides a good test case for live data processing. Heat Pumps are a common unit operation in chemical plants. A heat pump models most thermodynamic operations, including heat exchangers, compressors, and expansion valves, with phase changes between liquid and gas. The inclusion of a refrigerant loop makes it a good test case as recycle processes are common in industry. Gathering data from the heat pump dryer provides some insights into the challenges involved in real-time data collection.

Since the Heat Pump Dryer includes its own control system, we cannot send controlling actions to it. Closed-loop control is out of scope of this project and will be a future area of research.

Figure 4.1: Heat Pump Dryer

Figure 4.2: InfluxDB Database

4.2 Method

To gather data from the heat pump dryer, Bluetooth sensors were used to measure temperature and humidity at various points in the dryer. A power meter was used to measure the total power consumption of the heat pump dryer. A Raspberry Pi was used to collect the data from the sensors, and store it in an InfluxDB time-series database. InfluxDB was chosen because it is one of the platforms identified in the Literature Review as being used for aggregating sensor data; it is also open-source and easy to use. This can be viewed as representative of a data collection system that would be used in industry.

4.3 Discussion

The process of gathering and aggregating data was relatively straightforward, reinforcing the finding from the literature review that there are a number of tools already available for collecting and processing sensor data.

Depending on the process, there is often only a limited amount of information that can be gleaned from the system. For example, we were not able to collect pressure information, as pressure sensors are more costly and require modification of the refrigerant loop. Likewise, we were only able to measure the total power consumption of the heat pump dryer, rather than the power consumption of individual components. Data processing techniques will need to be able to infer the state of the system from the limited data available.

Additionally, in operation, the heat pump dryer has a number of different operating cycles: it does not just continuously run all of the time: sometimes it switches direction, etc. This changes the state of the system, so more complete digital twin systems need to be able to account for this. We did not have access to the control system of the heat pump dryer, so we were unable to collect data on the operating cycles. Depending on the setup in industry, this data may or may not be avaliable.

Chapter 5

Architecture Research

The literature review also identified a variety of modelling techniques, including Steady-state modelling, Dynamic Modelling, Surrogate Modelling, and Optimisation.

The Ahuora Digital Twin Platform is currently built to support only steady state modelling. However, to explore the potential for integrating more advanced live data processing techniques, the IDAES Process Systems Engineering Framework was employed as a testbed.

The IDAES framework is a Python library that provides tools for modelling and simulating chemical processes. It is designed to be extensible, allowing for the addition of new modelling techniques.

Using the IDAES framework, experimentation with live data processing techniques was undertaken. This testing aimed to evaluate the feasibility of integrating these advanced techniques into the Ahuora Digital Twin Platform. By leveraging IDAES, the platform can be designed to support the incorporation of more sophisticated modelling techniques in the future.

5.1 Research Questions

The research investigation aimed to answer the following questions:

- *RQ1:* How does the Ahuora Simulation Platform need to be modified to support dynamic modelling, surrogate modelling, and optimisation?
- *RQ2:* What architecture would best support the integration of live data processing techniques into the Ahuora Digital Twin Platform?

RQ1 focuses on future needs and long-term vision. This is important to help minimise the amount of rework required when implementing new features. *RQ2* focuses on the immediate needs of the project, and is important for developing a pilot implementation.

5.2 Dynamic Modelling

The IDAES framework was used to develop a dynamic model of a steam tank, with a valve controlling the inlet and outlet pressure and flow rate. A PID controller was used to control the valve opening fraction to regulate the pressure in the tank.

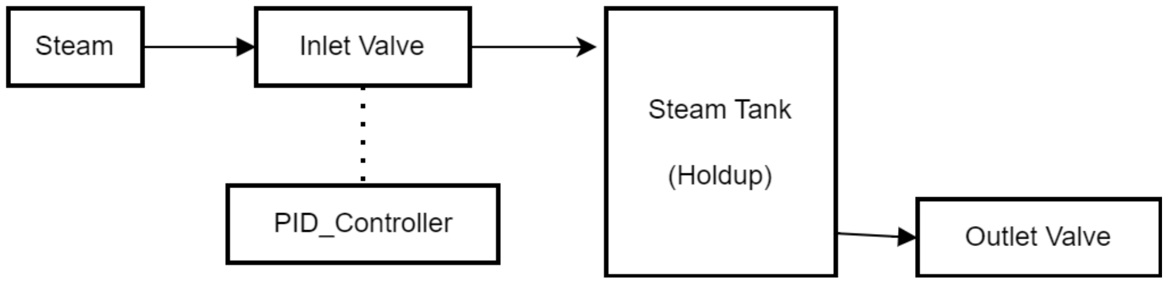

Figure 5.1: Dynamic Modelling of a Steam Tank

This provided a simple example of a dynamic system. The inlet and outlet valve and the PID controller were not dynamic models: From a mathematical perspective, this means their properties were fully determined by the inlet and outlet conditions. The only dynamic model was the Steam Tank. From a mathematical perspective, this means that the state of the steam tank was determined not only by the the inlet and outlet conditions, but also by the previous state of the tank, i.e how "full" the tank is.

5.3 Analysis

- The IDAES framework is well-suited to dynamic modelling, as it provides tools for creating and solving differential equations. It can easily model the same system at different time scales.
- The Ahuora Simulation Platform requires substantial changes to support dynamic modelling. Rather than storing a single value for each property, it will need to store the state of each property at each time step.
- This also necessitates significant UI changes to view the state of the system at different time steps. This could be achieved through some sort of time slider, and graph visualisations of properties over time.
- Specifying the initial conditions of the system is more complex, as the user needs to specify the initial state of dynamic properties, such as the initial tank level. Other properties, such as the valve opening fraction, need to be specified as functions of time.

5.4 Surrogate Modelling

Surrogate Modelling is the process of creating a simplified model of a complex system. This is usually done using machine learning techniques. This provides a good test case for implementing data-driven modelling techniques within the IDAES framework.

IDAES includes a framework for data-driven modelling called PySMO. This provides utilities for training polynomial, Radial Basis Function (RBF), and Kriging models to approximate the behaviour of a system.

To test the workflow for hybrid modelling, a simple surrogate model is created using PySMO. First, a set of data points is generated by solving a heater model at different pressures, temperatures, and flow rates. Then, this data is used to train a surrogate model, which can predict the outlet pressure and enthalpy from the valve based on the inlet conditions. This generated data for a steady-state simulation, predicting the outlet pressures and temperatures. An RBF network is trained to predict these values from the inlet conditions. The weights of the trained model are then saved to disk.

```
model = PysmoSurrogate.load_from_file('pysmo_heater_surrogate.json')
inputs = [self.inlet.pressure, self.inlet.temperature, self.heat_duty, self.
    inlet.flow_mol ]
outputs = [self.outlet.pressure,self.outlet.temperature,self.outlet_vapor]
self.surrogate = SurrogateBlock(concrete=True)
self.surrogate.build_model(model,input_vars=inputs, output_vars=outputs)
```

Listing 5.1: Using a surrogate model in IDAES

To use the surrogate model in a flowsheet, the weights of the model are loaded and the structure of the model is recreated as a set of algebraic constraints, relating the input conditions to the predicted output conditions. This can be combined with IDAES's UnitModel class and StateBlocks to create a new unit operation that can be used in a flowsheet.

As shown in Listing 5.1, each input and output is linked to a parameter in the unit operation. This works for steady-state models, but dynamic models have a set of parameters, so the surrogate model would need to be able to predict the next time step from the previous time step. This is a much more complex problem, as it would have to be time scale invariant.

5.4.1 Analysis

Surrogate Modelling can be achieved using IDAES's built-in PySMO libraries, or other similar libraries such as OMLT. It is reasonably straightforward to train a surrogate model to represent a non-dynamic unit operation, but dynamic unit operations get significantly more complex - instead of modelling a single value, the surrogate model must be able to model the entire time system. There are some methods of doing this, such as using neural ODEs, Residual Networks, Operator Networks, or some other sort of convolutional network. There is little research into applying these methods in the field of chemical and process simulation, especially in the context of mathematical modelling such as the IDAES framework.

The exact same process for surrogate modelling can also be used to model unit operations from historical data. This is useful when there is no mathematical model of the unit operation, but there is historical data available. This is very useful when considering the application of the Ahuora Digital Twin Platform to existing factories, where the exact mathematical properties of the unit operations are unknown but there is a wealth of historical data available. Online Learning techniques could be used to update the surrogate model in real-time, as new data becomes available. This is a key step in turning a "simulation" into a "digital twin", as it allows the model actively adapt to real-world conditions.

Because of the limited functionality of the Ahuora Digital Twin Platform, it is currently beyond the scope of this project to implement a surrogate model. However, the IDAES framework is well-suited to this task, and it is likely that a surrogate model could be implemented in the future.

Additionally, the Ahuora Digital Twin Platform will need a user interface to support creating these different types of models. As surrogate modelling is a complex process, the user interface will need to be able to guide the user through the process of creating a surrogate model from a dataset, and provide feedback on the quality of the model. This will require a significant amount of work, and will likely be a key focus of future development.

5.5 Optimisation and Control

Optimising a system involves adding an objective the model using standard Pyomo utilities, and then solving the model to find the optimal conditions. In the heater model, a cost function is added as a test objective, to find the ideal balance between heat duty and outlet temperature. The model is then solved to find the optimal heat duty that minimises the cost function.

```
def cost_objective(h):
return 3**(h.heat_duty[0]/5000) - (h.outlet.temperature[0]-350) * 33000
m.fs.heater.cost_objective = pyo.Objective(rule=cost_objective,
                                           sense=pyo.minimize)
```

Listing 5.2: Optimising the heater model in IDAES

This is shown in Listing 5.2. The model must be solved with degrees of freedom, i.e variables that the solver can adjust to find the optimal solution. In this case, the heat duty is the degree of freedom, but there can be multiple degrees of freedom in a model. In Figure 5.2, a setpoint function for the outlet temperature is added, and optimisation is used to find the heat duty for each timestep such that the resulting outlet temperature best tracks to the setpoint. This has one degree of freedom for each discrete time step that must be optimised: the heat duty at that time step. When this technique is used to control a system based on the optimal way to reach a setpoint, it is referred to as Model Predictive Control. Because the setpoint change is instantaneous, it is unable to perfectly follow the setpoint, but is able to minimise the error on either side of the setpoint through controlling actions.

The optimisation technique used here could be used iteratively for model predictive control, where the model is solved at each time step to find the optimal control action.

5.5.1 Analysis

Implementing Model Predictive Control in IDAES is relatively straightforward, as long as there is a dynamic model of the system, a cost function, and the optimisation problem is well-posed. The Ahuora Digital Twin Platform does not support optimisation yet, but this will be supported in the future.

In order for model predictive control to be useful, IDAES needs to be paired with a real-time data processing system. The real-time data processing system will need to be able to actuate the suggestions of the MPC simulation in real-time, and then inform the MPC simulation of the system's

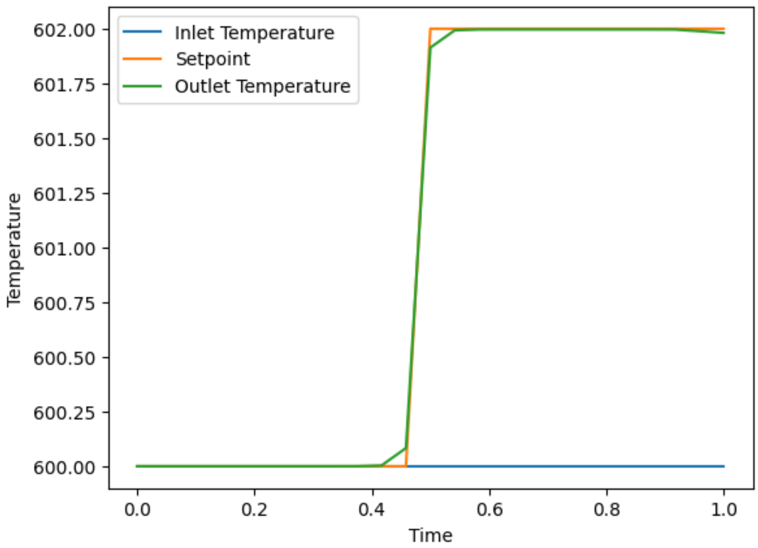

Figure 5.2: Optimising a dynamic model to follow a setpoint.

response. This requires integration with the industry-specific SCADA systems that are used to control factories.

5.6 Theoretical Architecture

Conventional simulation platforms do not have built-in support for live data processing. Likewise, conventional factory SCADA[1] systems do not have built-in support for complex simulation. To integrate a simulation platform with live data, a software system must be created that can merge the two systems. This can be considered as an intermediate layer between the simulation platform and the factory SCADA system.

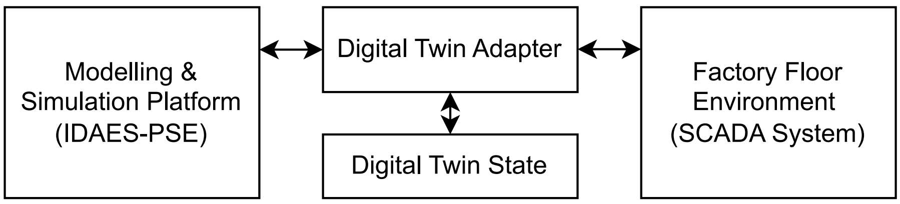

Figure 5.3: Theoretical framework of how a Digital Twin can be implementend on top of existing simulators and factory control systems.

The concept of a “Digital Twin” refers to a simulation of something in the physical world, which is kept up to date with a physical system using real-time data [3]. In Figure 5.3, this intermediate layer is what converts the simulation platform, and the data from the factory, into a digital twin.

By breaking the Digital Twin up into these core components, we are able to make use of existing software systems for live data processing and simulation, and only need to build the components that are unique to the Digital Twin use case. This limits the complexity of the system, and means that implementing a Digital Twin in a factory can be done with minimal disruption to the existing systems.

[1]SCADA systems: Supervisory Data Aquisition and Control systems.

5.6.1 Modelling and Simulation Platform

Being able to model and simulate a process is one of the foundational components of a Digital Twin platform.

Conventionally, simulation platforms are used when designing and planning a factory, and are not used during operation. This makes most conventional simulation platforms unsuitable for use in a Digital Twin, without substantial modification. In the Ahuora Platform, IDAES-PSE provides the core modelling and simulation capabilities. It is built on top of Pyomo, an Algebraic Modelling Language, enabling for flexible model definitions. This flexibility is key in enabling a Digital Twin platform to be built on top of IDAES. Further flexibility is enabled through the Ahuora Platform, which can be modified to support whatever features are required.

5.6.2 Factory Floor Environment

In the factory, there are many sensors and control systems that monitor the state of the factory. These systems are often connected to a SCADA system, which is responsible for collecting and displaying this data. Additionally, there are often data processing systems that are used to store and process this data, as historical data is often used in later analysis to improve performance or make maintenance and upgrade decisions.

There are also automated and manual control systems, which are used to adjust the factory's operation based on the data collected by the SCADA system. These systems are already implemented in factories, and are critical to the operation of the factory.

5.6.3 Digital Twin Adapter

As such, rather than making a new data processing system, it is better that a Digital Twin Platform can be built such that it can be implemented on top of established data processing systems. Likewise, rather than making a new simulation platform, it is better if the framework of an existing simulation platform can be reused or adapted to the digital twin use case.

The concept of a "Digital Twin Adapter" is used to define this functionality. It is responsible for converting the data from the factory into a format that the simulation platform can understand, and to provide results from simulations back the factory.

5.6.4 Digital Twin State

Key to the sucessful operation of the the Digital Twin Adapter is the fact that is also has access to the "Digital Twin State". The Digital Twin State represents an estimate of the current state of the factory, based on the simulation results and the live data. This represents the ability of a Digital Twin to "learn" how the physical system behaves, and adjust the inputs to the simulation accordingly.

5.7 Discussion

The architecture presented generalises the concept of a Digital Twin, enabling software to be built that can be used in a wider variety of factories. However, the exact implementation of the Digital Twin Adapter may need to change depending on the factory's existing systems. Different SCADA systems and data processing systems will have different ways of communicating, and different data will be avaliable depending of the factory. Likewise, different methods of simulation may or may not be avaliable depending on the system.

The Digital Twin Adapter will need to be built in a way that it can be customised or set up differently depending on the factory.

The current Ahuora Simulation Platform is split into three main parts:

- The Frontend UI, which is written in Typescript/React and runs in the user's web browser. This is responsible for rendering the flowsheet, and allowing the user to interact with the simulation.
- The Backend API, which is written in Python/Django and runs on the server. This is responsible for storing the simulation data, orchestrating calls to run the simulation, and returning the results to the user.
- The IDAES solving engine, which is written in Python and runs on the server. This is responsible for solving the simulation, and returning the results to the API. It has been seperated out from the API to allow it to be scaled independently.

Over time, the Ahuora Simulation platform will be expanded to support dynamic modelling, surrogate modelling, and optimisation.

Additionally, there are also a number of other tools that are used in industry for collecting and processing sensor data. There may be an integrated solution or a number of different tools that are used together, but they can be grouped by their functionality:

- Data Collection: These tools are responsible for collecting data from sensors and storing it in a database, as well as cleaning and preprocessing the data. This includes IoT networks and SCADA systems.
- Data Processing: These tools are responsible for processing the data to extract useful information. This includes machine learning models, statistical analysis, or other data processing techniques. Generally, these tools will sit within some sort of framework or pipeline, such as Apache Kafka, Flink, or a custom solution.
- Data Storage: These tools are responsible for storing the data in a way that is accessible to the other tools. This includes databases, data lakes, or other storage solutions. In Model-Based Systems Engineering, this is often referred to as a Knowledge Base.

Thus, a future requirement of the Ahuora Simulation platform is the connection to these tools to access the sensor data.

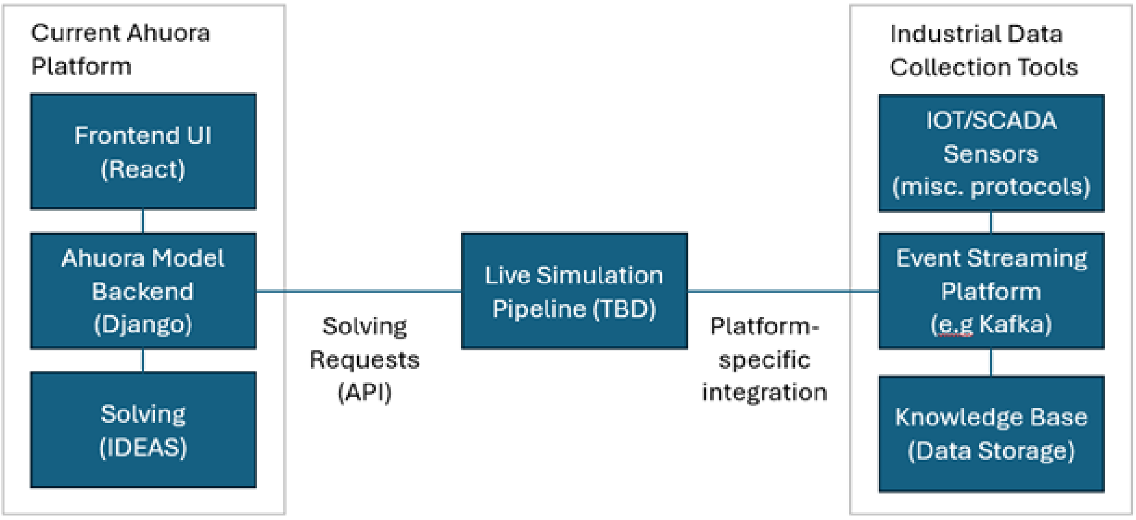

Figure 5.4: Anticipated method of implementing the Ahuora Simulation Platform into an industrial system.

Deploying the a model in production would likely only be done by a trained engineer who manages the plants' SCADA systems. The deployment would have to be custom for each plant, depending on the SCADA system in use, the sensors available, and the model being used. Making a User Interface for this process would be very challenging, and would limit support to only a certain number of protocols. Thus, it can be considered that the interface, pipeline, or service between the Ahuora Simulation Platform and the SCADA system is a custom software component.

There is a discrepancy in the use cases of the Ahuora Simulation Platform and the standard industrial stack used in live data processing. The Ahuora Simulation Platform is designed with experimentation and analysis in mind, which are usually offline[2] tasks performed by expert engineers. The standard industrial stack is designed for real-time prediction and control, which are online tasks performed by operators. Both the engineer and operator workflows will require all the modelling and simulation techniques identified, but the operator's workflow uses a fixed set of models.

These two types of users, Engineers and Operators, can be considered archetypes of user behaviours and needs.

Table 5.1: Comparison of Engineer's and Operator's Requirements

Requirement	Engineer	Operator
Use Case	Design, Retrofit,	Maintenance, Monitoring, Control
Modelling	Creates and modifies multiple models	Fixed set of models, only certain parameters may change
Data Usage	Historical data, likely preprocessed or cleaned	Real-time raw data
Reliability	It is acceptable if incorrectly configured models don't solve	Model must be correct; Requires 100% reliability

Because of the different needs between these two archetypes, some requirements in the platform

[2]Many use cases of the Ahuora Simulation platform include designing new factories or modifications to a factory. These tasks mostly use historical data, and are a distinctly seperate problem from live prediction and control.

appear to conflict. While the Ahuora Simulation platform needs to be designed so that it is easy to design and build a model, this functionality is not required in operation. The model could be considered "frozen" at this point, where only certain parameters can be changed based on the real-time data. It does not make sense to expose functionality to edit the model to an operator. Further testing is required to understand the different deployment procedures that could be used.

Chapter 6

Prototype: Live Data Simulation

6.1 Purpose

The previous chapter identified that the best way to implement a Digital Twin system is to implement an adapter, or bridging system, between the simulation platform and the factory SCADA system. This chapter describes a prototype of an adapter layer. This prototype is developed for demonstration purposes, illustrating how the Ahuora Simulation Platform could be integrated with a real-time data processing system. As such, this prototype reflects a minimum-viable product, assessing the feasibility of the architecture developed in the research stage. Insights from the prototype are used to refine the requirements of subsequent work.

6.2 Method

The Ahuora Simulation Platform has a REST API that is used internally to communicate between the frontend and the backend. This API already includes endpoints for updating properties, and retrieving properties after a simulation. This API could also be used for real-time data processing systems to communicate programmatically with the platform.

The flowsheet needs to be set up ahead of time with the relevant unit operations, and properties. Each property field has a unique ID, generated by the database. To solve the flowsheet repeatedly based on real-time data, the properties in the flowsheet are updated programmatically to reflect the real-time state of the system. Following this, a call is made to the API endpoint to solve the flowsheet. Relevant properties can then

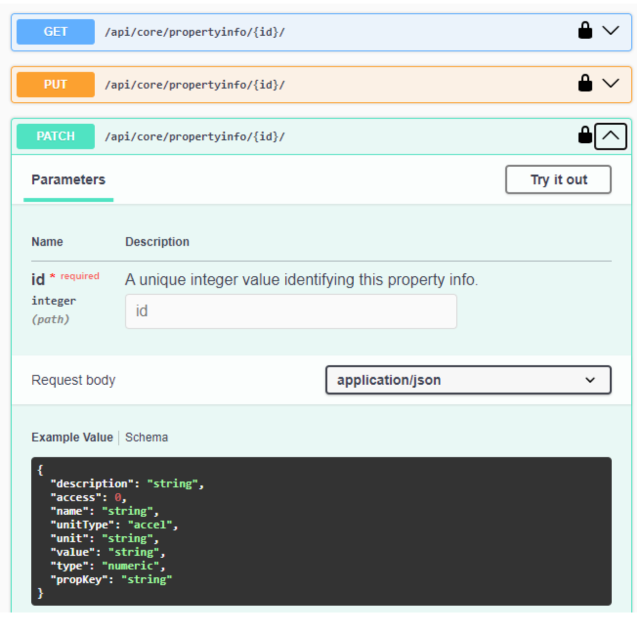

Figure 6.1: Swagger definition of API endpoints for updating properties.

Figure 6.2: Prototype UI designs for setting a property as a "real-time" property.

```rust
pub const SENSOR_DEFINITIONS: &[SensorDefinition] = &[SensorDefinition {
    location: "Wall Plug",
    measurement: "current_power",
    unitop: "my_pump",
    propkey: "PROP_PU_3", // Power Required Property
}];

pub const CALCULATED_PROPERTIES: [CalculatedProperty;2] = [
    CalculatedProperty {
        unitop: "pump_outlet",
        propkey: "PROP_MS_1", // Pressure
        display_name: "Pump Outlet Pressure",
        ..CalculatedProperty::default_vals()
    },
    CalculatedProperty {
        unitop: "pump_outlet",
        propkey: "PROP_MS_0", // Temperature
        display_name: "Pump Outlet Temperature",
        ..CalculatedProperty::default_vals()
    },
];
```

Figure 6.3: Definitions file for mapping properties in the Ahuora Simulation Platform to the real-time data processing system.

be queried to get the results of the simulation.

As the property IDs are generated by the database, it does not make sense to hard-code them into the real-time data processing system. Rather, a more generic way of referring to properties that need to be updated is required. Some prototyping of a UI method to set properties as "real-time" in the existing interface was done, as shown in Figure 6.2. This allows the user to select which properties should be updated in real-time, and establishes a mechanism to refer to such properties within the real-time data processing system. The work within this project stopped short of implementing full real-time data integration. This decision was made in part because there is not yet a clear delineation between the workflows of an engineer and a plant operator within the Ahuora Simulation Platform. Furthermore, for the purposes of a proof of concept implementation required within the scope of this work, simulating live inputs from dummy data was considered adequate.

A simple definitions file was used to map the properties in the Ahuora Simulation Platform to the properties in the simulated real-time data processing system. As shown in Figure 6.3, the file included the unit operation name, the property key (used by the Ahuora Simulation Platform to determine property types), and the sensor ID from the real-time data processing system. The data processing platform could use this information to find the property IDs in the Ahuora Simulation Platform, and update them with the real-time data.

Simulation of real-time data was achieved through the use of CSV file with dummy data points. This approach was appropriate for development purposes within the scope of a proof of concept prototype, as functional changes and debugging were able to be undertaken in a rapid fashion without being limited by time constraints associated with restarting a physical system. It is worth noting that the format of this data was consistent with historic data recorded from real-time sensors collected in Chapter 4. Therefore it is reasonable to extrapolate that the success of the implemented solution across multiple timesteps defined within the dummy data is demonstrable of the potential of the prototyped approach for live data situations.

This prototype also used a very simple model: a pump with an inlet stream and an outlet stream. The power used by the pump was calculated based on the "live" data, and the inlet streams were already specified. The simulation was used to calculate the outlet pressure and temperature of the pump. This is a simple model, but it is sufficient to demonstrate the feasibility of the architecture.

When the script file was run, for each new data point, the script updated the properties in the Ahuora Simulation Platform, and then called the API to solve the simulation. The results of the simulation were then printed to the console.

This worked suprisingly well, and was able to be done in only around 300 lines of code. It was implemented in Rust, using an API client SDK generated from the OpenAPI specification of the Ahuora Simulation Platform. This made it fully type-safe and reliable. Because of the configuration file format, it is trivial to add a longer list of sensors and calculated properties, as is required in more complex properties.

6.3 Discussion

This prototype demonstrated that the Ahuora Simulation Platform can be integrated with a real-time data processing system with relatively little effort, as long as there is a standardised API. Depending on the type and quality of the sensor data, there may need to be some data cleaning and preprocessing, specific to a particular use case. Hence, any data preprocessing pipeline or service for the Ahuora Digital Twin Platform should be kept as simple as possible. To confirm this, the heat pump dryer was then used as a case study in linking a more complex system to the Ahuora Simulation Platform. This provides a more realistic test of the architecture.

This approach, which closely follows the architecture described in Figure 5.4 in Section 5.7, also outsources the processing, visualisation, and control actions to third-party systems, as all data would be stored in the factory's existing knowledge base after processing, completely external to all Ahuora Systems. The only data that the Ahuora Digital Twin Platform would need to store is the current state of the system.

This makes it easier to build a "headless" version of the Ahuora Simulation platform that can be deployed as a single frozen model, into a factory's existing systems. This is beneficial for stability and reliability reasons, limiting the complexity of the system. This is the unix "do one thing and do it well" philosophy applied to software architecture.

However, there are disadvantages to this approach. One of the key value propositions of the

Ahuora Digital Twin Platform is that it provides one place to define a factory's architecture, and then multiple types of analysis can be used on it. The same structures could be helpful for automatically creating visualisations of the real-time state of the factory, fault diagnostics, control, and surrogate modelling. As the platform currently stands, it cannot be considered a "Digital Twin", as true digital twins include multiple fidelities of simulation, and dynamic "state" that adapts to real-world conditions via a feedback loop. The next steps in development balance these two approaches, to find an architecture that includes the advantages of each.

Chapter 7

Updating the Ahuora Simulation Platform

Up until this point, the presented work has focused on experimentation, research, and prototyping. Chapters 4 through 6 identified key use cases, requirements, and architectures for processing live data. Chapter 4 identified that while data collection is straightforward, there are limitations to the data that can be collected. Often, custom logic for inference or sensor fusion will be required. Chapter 5 concluded that IDAES is capable of supporting dynamic modelling, surrogate modelling, optimisation, and control, but that these features mean different things in the context of live data processing compared to offline simulation. Chapter 6 demonstrated that the Ahuora Simulation Platform could be integrated with a real-time data processing system for steady-state modelling with relatively little effort, as long as there is a standardised API, and provided techniques to do so. However, much more complexity arises when combining all these concepts together, and when balancing the needs of the engineer's workflow with the operator's workflow.

This chapter focuses on iteratively developing the Ahuora Simulation Platform to support live data processing, and testing these developments on the Heat Pump Dryer Model.

7.1 Recording History

7.1.1 Purpose

To better understand the tradeoffs of a more integrated platform, it was decided to implement a solve history system. This would allow the user to view the results of previous simulations, and compare them. It is a useful feature for the Ahuora Simulation platform as a standalone tool, but it can also be used to record the results of the real-time data processing system. A simple dashboard is created to view past solving results, which can show how the system has changed over time. Significant advantages revealed by this approach demonstrate the potential for development of a more integrated system. Beyond such a development, this approach also provides insight into how to break the system apart while retaining similar functionality.

7.1.2 Development

The Ahuora Simulation platform stores all simulation results in a database after they are returned from IDAES. Initially, only one entry was maintained and the data was overwritten by each subsequent flowsheet solve. To enable storing history of previous solves, a table was created to store the results of each previous solve, and new entries were automatically added each time the simulation was run. Then, a user interface was created to view the results of the simulations.

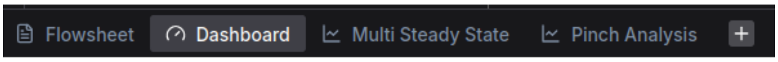

Figure 7.1: A tabbed interface was used to switch between the flowsheet view and the graphs.

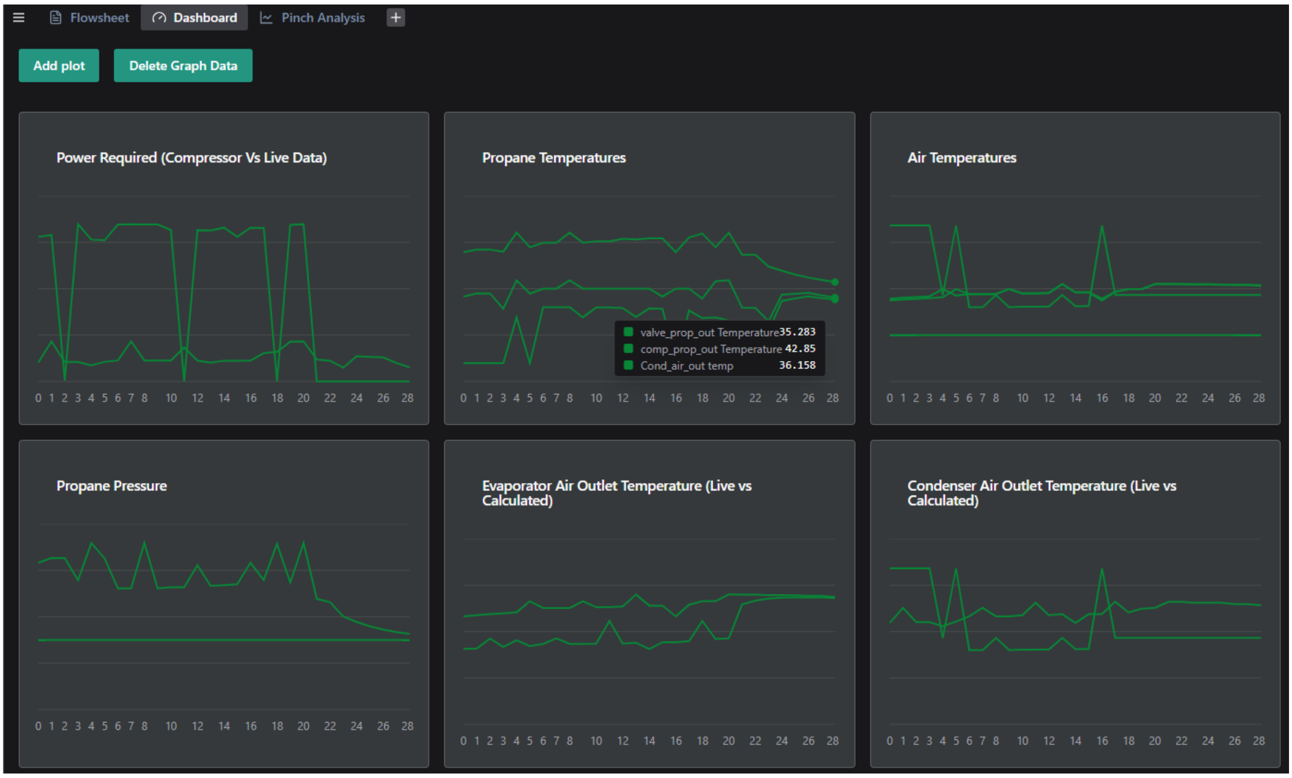

Figure 7.2: Live Data Dashboard in the Ahuora Digital Twin Platform

A seperate tab has been added to the Ahuora Simulation Platform to show the graphs of previous simulation results, as shown in Figure 7.1. This tabbed interface is visible on all pages of the application so it is always easy to access. Figure 7.2, shows the user interface that was created as part of the development, with a number of manually created graphs for the heat pump dryer. Functionality to edit and create new graphs was also added, and a screenshot of the modal to edit a graph is shown in Figure 7.3.

In the Django Backend multiple new tables are added to the Database. A table was added to store a Historical entry, to store a Plot, and to store a Series in a Plot. Django is a declarative framework, and Python classes are used to define the structure of the database, which is then converted into tables in the backend Postgres Database. Django also automatically handles generating migrations between database versions, ensuring that this functionality would not break existing work.

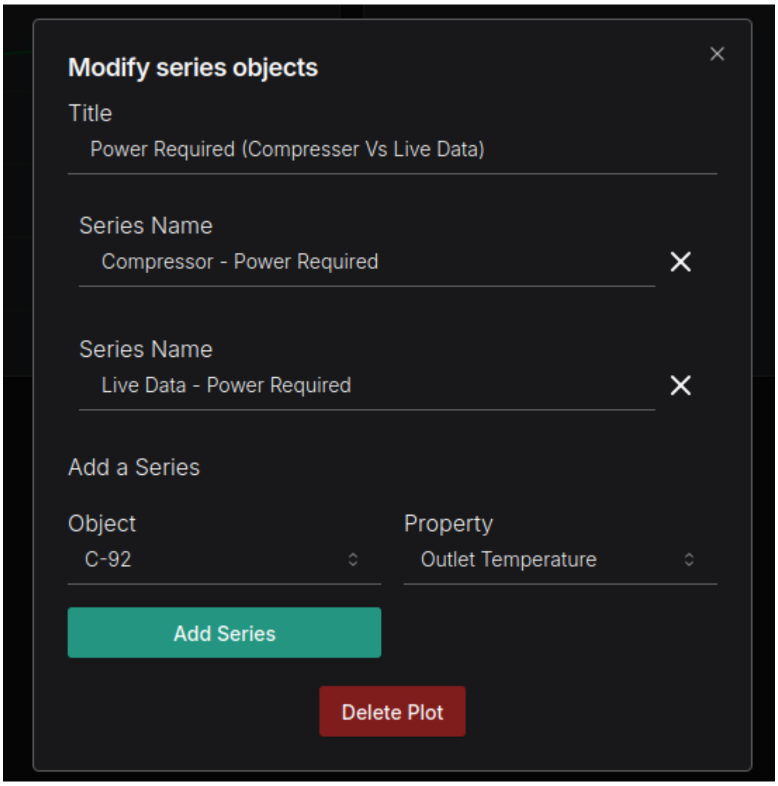

Figure 7.3: Editing a graph in the Live Data Dashboard to show multiple series

7.1.3 Testing

The data collection scripts from Chapter 4 were updated to store the results in the Ahuora Simulation Platform instead of InfluxDB, using methods previously outlined in Chapter 6. A heat pump was modelled in the Ahuora Simulation Platform. At this stage, the Ahuora Simulation Platform was able to reliably solve heat pumps with pure chemical components, and as the heat pump dryer used propane as a refrigerant, it was a good test case.

Figure 7.4 shows the heat pump model built in the Ahuora Simulation platform for this test. There are four unit operations in this model:

- The compressor pressurizes the propane gas, heating it up at the same time.
- The condenser cools down the propane, and the propane liquifies. The heat that leaves the propane moves into the internal drum of the dryer.
- The propane then moves through the valve, which reduces the pressure significantly. This causes the propane to vaporise, and the temperature further decreases to well below room temperature.
- Lastly, the propane moves through the evaporator, where it collects heat from the ambient surroundings, back to its original temperature.
- The recycle block is not a unit operation, but it specifies that the propane from the evaporator is fed back round into the inlet of the compressor. This means that the temperature, pressure, and flow rate out of the evaporator is the same as into the compressor.

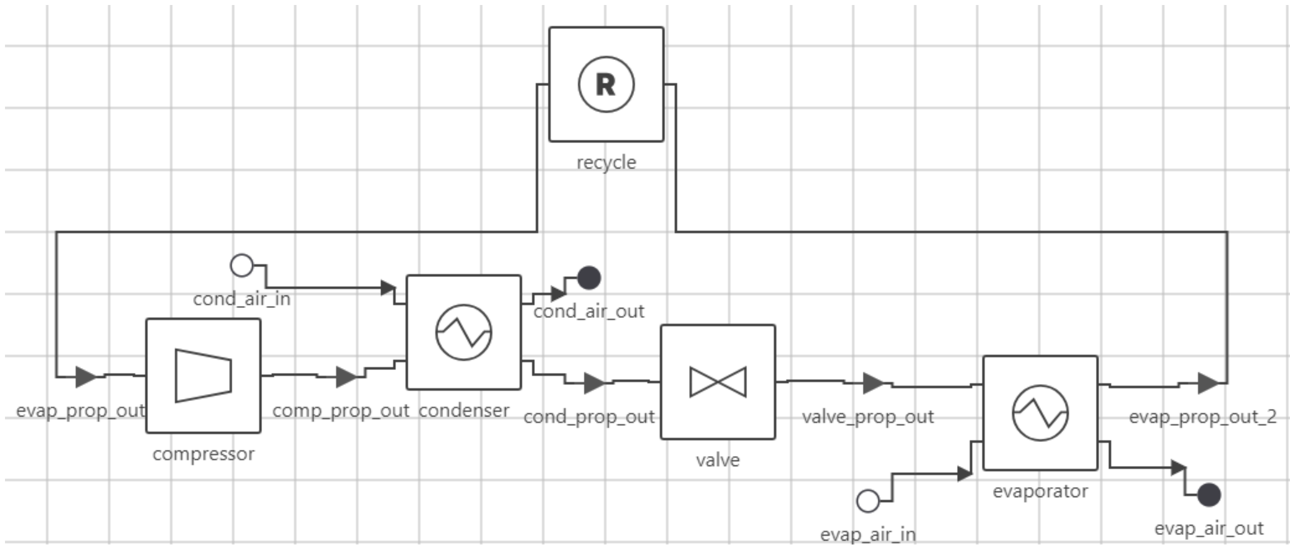

Figure 7.4: Heat Pump Model used for live data

The complexity arose in linking the data from the real-world sensors to the data from the simulation. In the data collection configuration the dryer was set up for, a sensor is used to read the temperature of the propane leaving the compressor, and out of the evaporator. Additionally, air temperatures are also measured, and the total power of the dryer is measured. Other properties required to calculate the flowsheet needed to be hardcoded, as shown in Table 7.1. The sensor readings that were measured, but not used in calculating the flowsheet are shown in Table 7.2. The reason not all sensor data was used is twofold: first, the air temperature readings would also require knowing the flow rate of the air and propane, which were unknown. Second, they would over-define the simulation which would make solving infeasible. As there were no pressure sensors, the pressure of the propane could not be measured. Thus, hand-calculations were performed to calculate what feasible pressures the propane would be at for it to boil below room temperature and at approximately 70 ° C. The model performance did not always exactly reflect real-world conditions, but it was accurate enough for proof of concept purposes.

The Ahuora Simulation Platform did not initially support specifying the outlet temperature of a compressor or valve, when these make more sensible calculation modes than outlet pressure for our avaliable live data. These were added to support this workflow.

Table 7.1: Specified properties in heat pump model

Unit Operation	Property	Source
Compressor	Outlet Temperature	Probe Temperature Sensor
Compressor	Isentropic Efficiency	Hardcoded to 100%
Condenser	Air Inlet Temperature	Hardcoded to 20 °C
Condenser	Propane Outlet Temperature	Hardcoded to 40 °C
Valve	Outlet Pressure	Hardcoded to 6 bar
Evaporator	Air Inlet Temperature	Dryer Temperature Sensor
Evaporator	Propane Outlet Temperature	Probe Temperature Sensor

As the Ahuora Simulation Platform can only store solving results, additional “dummy” unit operations were added to the flowsheet to store the other sensor readings mentioned in Table 7.2.

Table 7.2: Other sensor readings, not used in solving

Unit Operation	Property	Reason
Condenser	Air Outlet Temperature	Can be calculated from propane outlet temperature
Evaporator	Air Outlet Temperature	Can be calculated from propane outlet temperature
Entire Dryer	Power Usage	Doesn't map cleanly to one property

The live data processing script was then started and run with the developed model as specified. While the model was thermodynamically valid as it was built, and could solve correctly, the integration of live data quickly resulted in the model failing to solve. Sometimes, IDAES would fail to initialize the models. Other times, the IPOPT solver found that either the model was infeasible, or the problem failed in restoration phase (which could be from poor scaling, inaccurate results, or it could not find a feasible starting point for solving).

There are several potential causes for such failures. One possibility is hardcoded data. A likely source of failure is the discrepancies between the dynamic nature of the physical heat pump dryer and the steady state nature of the model. It takes time for the dryer to reach steady state, as it initially starts at room temperature. The model is trying to solve everything as a steady state - wheras in reality it takes time for changes in temperature to propagate through the entire system.

Additionally, the dryer has it's own built-in control system, and it is not always running. When the dryer is not running, the simulation model is not correct. This may result in infeasible solutions.

7.2 Results

This identified a number of problems with the current ways of handling live data.

Firstly, there needs to be a better way to handle failed solves. The only data that the flowsheet history stored was data on successful solves. When debugging solving, it would be much more useful to store successful and failed solve states. However, it may not make sense to store failed solve results in the Ahuora Simulation Platform, as the results from idaes are useless in that case. A potential solution can be found in the form of an external data pipeline to manage this. Additionally, the Ahuora Simulation Platform only stores properties of unit operations in the solve, making it hard to add additional data such as timestamps, or other data collected from sensors. It doesn't provide a fully-fleged data processing platform. In the test example, additional "dummy unit operations" disconnected from the main flowsheet were used to store the data, but this is a workaround and should not be viewed as a long term solution. It also would not be appropriate for other types of data, e.g if images were taken of the plant over time. From this, we can conclude that it is not appropriate for the Ahuora Digital Twin platform to form the entire knowledge base of a plant.

Secondly, the methods of specifying unit operations can vary, and the Ahuora Simulation Platform will need to support a wide number of calculation modes - in this case, an additional calculation mode (Outlet Temperature) had to be added to the compressor in order to solve the model. On discussing these results with other colleagues who are chemical engineers, they mentioned that it is common to have to calculate properties in roundabout ways. For example, you can calculate temperature by how much a pipe expands, or the pressure of a valve outlet from its temperature, assuming it's in the two-phase region (transitioning between liquid and gas) and is a pure fluid. The Ahuora Simulation Platform has “Specification Blocks” on it’s roadmap of future development, which allow specifying arbitrary relationships between variables. This may help with such scenarios. Additionally, it may be worth introducing a preprocessing step into the workflow prior to feeding the calculated values into the Ahuora Simulation Platform.

There was a lot of data that could not be used in the Ahuora Simulation Platform due to limited flexibility of existing model specifications. With greater flexibility, the inlet and outlet air temperatures could have been used to calculate the heat capacity of the air streams, or efficiency of the heat exchangers. Alternatively, overspecified data could be used for sensor fusion as explained in the Literature Review (Appendix C), using Bayesian statistics to calculate what the most probable true values are and the uncertainty associated with them. However, this would significantly increase the complexity of the model; an alternative is to just compare the simulation output to the live data, and flag any significant variations as a problem in a concept drift approach. Sensor fusion would likely require bayesian methods to be implemented into the Ahuora Simulation Platforms, comparing variances could be done as a postprocessing step in a seperate software.

7.3 Improving Solving Reliability

As the solving for the heat pump dryer was not always successful, some additional time was spent to improve the reliability of solving. This helped to identify relevant preprocessing steps to implement in the platform. There are a couple of reasons why the solving of the flowsheet was not always successful in the previous section.

The first reason is that the heat exchangers could be failing to initialise. They are the most complex unit operation, and require appropriate fluid volumes, heat deltas, flow transfer areas, and heat transfer coefficients to be set. If these are not set correctly, the heat exchanger may not initialise.

As we did not have flow rates for the air inlets in this model, approximate values were being used. Thus including heat exchangers in the model did not provide any more accuracy, and only added complexity. The model was updated with heaters and coolers instead of heat exchangers, which can be solved much more reliably.

The second reason why the model may not have been solving is that the recycle loop was not being initialised correctly. The recycle stream was being set to the same value as the outlet stream, but this creates a continuous loop that the solver may not be able to solve at a steady state as it is classed as a degenerate problem. In IDAES, tear guesses are used to decompose continuous loops sequentially until the solutions start to converge. For the next test, the recycle was removed, and the tear guess values were used as the initial conditions. This achieves a similar simulation result, with slightly fewer constraints and much less complexity.

When both of these changes were made, the model was able to solve reliably. The results of the simulation were much more consistent, and better reflected the expected behaviour of the heat pump dryer.

While implementing this, another issue was discovered. Because stream properties were being used to store additional live data that was not part of the simulation, sometimes the live data was invalid as a stream condition. This was done so the part of the platform that recorded solve history could be used to display all live data results, not just the simulation results. However, this was not a good solution, as it made the simulation less reliable. This means that either the solve history feature should be updated to handle storing additional metadata, or the live data should be stored in a different way.

7.4 Data Preprocessing

As concluded in Chapter 7, it is required that a comprehensive Digital Twin Platform support the workflows of process engineers, who are responsible to link the raw sensor data to the model. This workflow is seperate from the workflow to design a model, and thus requires a seperate set of tools. In this chapter, a simple data preprocessing pipeline is added to the Ahuora Platform to support this workflow.

When developing a way to link a simulation to real-time data, raw historical data is generally used to test the data processing pipeline, because it allows for testing against a standardised set of conditions. Then, the same pipeline is used to process the real-time data when the solution is deployed. Thus, the tools to process historical data should be the same as the tools to process live data. Building these tools in the Ahuora Digital Twin platform will provide a foundation to evaluate how to implement more advanced data processing techniques.

As with the history feature added in Chapter 7, this feature also is a core requirement of the Ahuora Simulation platform. It is required to support multi-steady-state simulation, a common type of analysis for chemical engineers. Thus it is an appropriate place to continue development.

7.4.1 Design

The simplest implementation that fits these requirements is a way to set properties to many different values, from some external data source (e.g a CSV file, or a live data stream). Thus the simulation could be run with many different conditions. Each successive simulation would be added to the history, so all simulations could be compared. This requires a way to load in data, and a way to specify which properties should be updated. It needs to be done in a way that can be easily extended from the simple “CSV file” case to a live data stream.

7.4.2 Loading Data

On consultation with other members of Project Ahuora, the tool that chemical engineers would be most familiar with for this task is a spreadsheet. Thus, it was decided to design the platform in a similar structure. The user would need to upload a csv file, and all the data could be viewed in a table. As in a spreadsheet, additional ”columns” could be added to the table, with calculated values based on the row.

This paradigm makes sense from a chemical engineer's standpoint, and from a live data processing standpoint. Each ”row” can represent one point in time, with all the data avaliable at that point in time. The ”columns” can represent different properties, and can be linked to properties in the simulation. In a live data processing system, due to performance reasons you often are only able to access the current row, and perhaps aggregation functions or other simple calculations from previous rows. This can be easily replicated in this ”spreadsheet” view, as we can limit the user to only being able to access the current row, and the previous row.

7.4.3 Linking Data to Properties

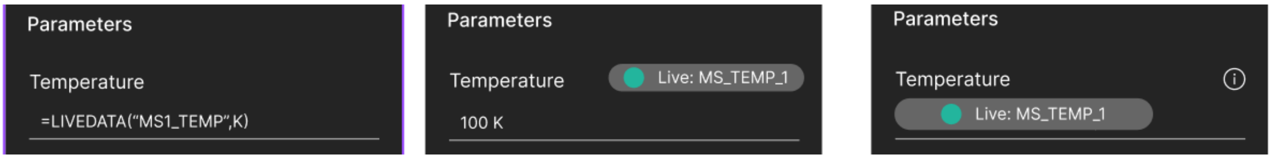

Figure 7.5: Design concepts for linking a property to a live data source.

Some conceptual designs for linking data to properties were created using the Figma prototyping tool, as shown in Figure 7.5. These designs were modifications of the sidebars and property panels already implemented in the platform. The first design drew inspiration from spreadsheet formulas. The advantage of this approach is flexibility in constructing more advanced formulas to link to a property. The second and third designs were variations on a menu system that allows for choosing a property.

On consultation with the others working on the Ahuora Platform, it was decided to most closely follow the second design, because it has the least impact on existing functionality, and is relatively simple to implement. The flexibility in constructing formulas can still be added in the “spreadsheet” view, keeping the main flowsheet view as simple as possible

7.4.4 Implementation

To keep the implementation simple, the CSV file was stored in the browser's local storage, and a seperate request was made to the server to solve the flowsheet at each timestep. The solve request would include the values from the CSV, for that timestep. Though it may not be the most efficient method for batch processing, this most closely mirrors the way that a live data processing system works.

The frontend was modified to allow the user to upload a CSV file, and to view the data in a table. The user can add new columns to the table, and specify a formula for the column. This process is shown in Figure 7.6. The data in the columns of the CSV file are shown in the “efficiency” and “power” properties. The calculated property does not show a list of values, instead it shows a formula that can be custom-defined by the user. An example usage of the calculated property expressions would be to take an average of multiple temperature readings from different sensors, so the average can be used in the flowsheet. This enables workflows of live data preprocessing to be

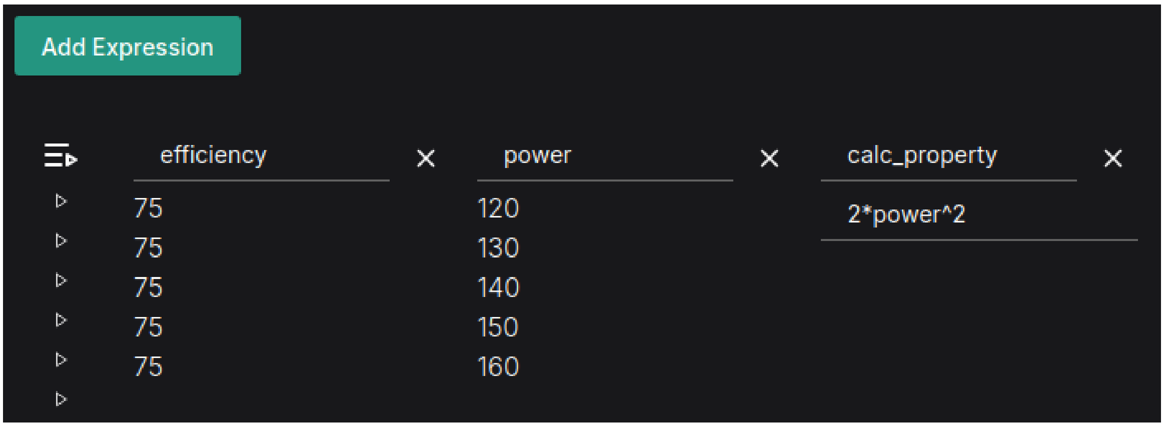

Figure 7.6: "Spreadsheet View", showing some example properties to be set on each solve.

constructed within the Ahuora Platform, rather than externally in a manually defined preprocessing script. Additional expressions can be added via the button at the top, if more calculated properties are needed.

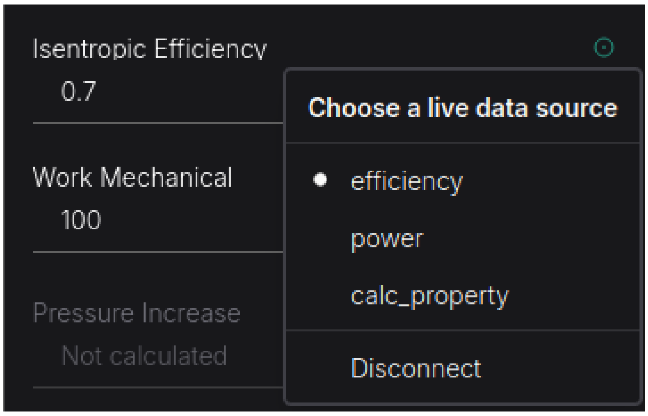

Figure 7.7: Linking a property from the spreadsheet view to the flowsheet.

A column is then linked to a property in the simulation via a new ”link” button on the flowsheet page. This is similar to the design in Figure 7.5, however it was decided to make it more subtle with a simple small icon on each property. As shown in fig. 7.7, the green target icon opens up a dropdown where the user can choose what live data source to connect to. If the property is not connected to any live data source, the icon goes grey.

Since any field can be connected to any column in the spreadsheet view, it is very easy to create complex data preprocessing routines with this interface.

In the backend, the solving functionality was updated to allow for parameters to be passed when a call to solving is made. These parameters are populated with the data from that row of the spreadsheet view. In this manner, seperate solves can be performed without having to update the properties of unit operations manually.

The most complex part of the backend implementation is the formula processing. Formulas need to be able to reference only certain other properties, and need to be able to be parsed from strings. SymPy was chosen as the library to do this, as it is a well-known library for symbolic mathematics in Python, so it supports the Ahuora Backend, and has support for complex enough expressions for this use case. It also provides a layer of security over using Python's built-in eval function, as it only evaluates mathematical expressions. Sympy's parsing libraries support specifying a dictionary of variables that can be used in parsing. This is used to pass the parameters of the solve into the query.

Results of the simulation can be viewed through the flowsheet, and through the history page. Future work could focus on a more integrated way of viewing results, such as including results of simulation on the "spreadsheet" view.

7.4.5 Testing

The functionality was tested by making a simple flowsheet in the Ahuora Simulation Platform, and uploading an example CSV file. The flowsheet was able to successfully solve at each timestep, and the results of each solve were viewable through the history page.

This same technique can be used in processing data from the Heat Pump Dryer, to avoid manually having to specify what unit operations each live data source needs to link to externally. This simplifies the process for those who are responsible for linking a simulation model with a live data source.

7.5 Discussion

The functionality for recording solve history and preprocessing data is a valuable addition to the Ahuora Platform, because it complements existing features, while making it more suitable to also be used in an online environment in the future. It also provides additional insight into the user archetypes and theoretical framework identified earlier in this dissertation.

7.5.1 Reflection on User Archetypes

The development conducted has been able to meet the needs of both an engineer and an operator, as identified in the user archetypes in Section 5.7, Table 5.1.

While both the engineer's use case and the operator's use cases are still valid, there was an assumption that the person setting up the digital twin would be able to easily port an engineer's model (e.g, from the design of a plant) into a digital twin that processes live data; that it would be just a matter of linking the model up with a live data source. From the development of the history and data preprocessing functionality, it is clear that finding the correct parameters to define a model, and finding a way to link sensor readings up with simulation parameters, is a non-trivial task: it is not as straightforward as a one-to-one mapping. Thus, two other user archetypes should be considered. A "Data Engineer" may be a Software Engineer responsible for aggregating data from different sensors (and simulation results) into a single SCADA system or Industrial IoT system. A "Process Engineer", for want of a better term, may be a Chemical Engineer who is responsible for linking existing simulation models with the IoT data, using their domain specific knowledge to convert the IoT readings from sensor outputs into valid parameters for a simulation model, using sensor fusion or other techniques. These need to be considered alongside the requirements of a chemical engineer only focused on design, and an operator only focused on control.

Table 7.3: Comparison of a Data Engineer's and a Process Engineer's workflow

Requirement	Data Engineer	Process Engineer
Use Case	Integrating simulation results into company knowledge base	Convert sensor readings into model parameters
Skillset	Software, Networking, Data Storage, and Data Processing	Chemical Modelling, Data Processing
Modelling	Doesn't do chemical modelling	Extends the model with sensor fusion & preprocessing techniques
Requirements	Good API to interface with the platform	Easy to use tools and techniques for sensor fusion
Historical Data	Needs to store historical data	Needs to test models on historical data

Table 7.3 summarises the requirements of each of these new use cases. The data engineer is most concerned about integration with existing systems. This would be best serviced by ensuring the data processing platform in Ahuora has simple, clear APIs to ingest data and extract data after solve. The Process Engineer wants easy access to all the sensor data in the factory, and sufficient tooling to be able to convert the sensor data into model parameters. This could be achieved by adding additional model properties to the Ahuora Simulation Platform, but more complex cases may require additional pre/postprocessing outside of the IDAES solver. Exactly what this could entail is highly implementation-specific.

7.5.2 Reflection on Architecture

In the previous chapter, Section 6.3 discussed the tradeoffs between doing everything in the Ahuora Platform, and doing as much as possible in an external system. The insights gained through implementing the solving history feature suggest that the Ahuora Simulation platform alone is insufficient, and that linking with external systems would be required to store results in line with other company data. Nonetheless, the solving history feature still is valuable for Process and Chemical Engineers who are testing different types of models offline.

Storing simulation results outside the platform also enables more advanced processing techniques to be implemented through custom logic outside of the Ahuora Platform. A seperate suite of tools to handle a data processing pipeline would best support this, and a full implementation would be a significant undertaking that is beyond the scope of this project.

However, specific techniques that involve extending the mathematical model of the system would be best done in the Ahuora Simulation Platform. Additionally, a Process Engineer, who is not as skilled at software engineering or programming, would be best served by having some data preprocessing tools built into the platform. As an integrated solution, they could test their models on historical data, using platform tools for data processing. A Data Engineer could then be responsible for providing the required live data to the platform, and storing simulation results, without having to deal with the intracacies of the model itself.

The implementation of the data preprocessing pipeline provides valuable insight into this. The preprocessing functionality provides a method to parameterise the simulation, abstracting away the details of the internal setup of the simulation. The previous methods involved making multiple API calls to set each property, and was more heavily dependent on the exact implementation. This method allows Process Engineers to have more control over how live data is processed, without needing to rely on the way that a Data Engineer would link it up with the live data source, because it allows them to decide what they will expose to the live data processing system. It also enables different models to be constructed from the same data source, which could be helpful for debugging and testing, or modelling different operation modes.

As much as possible, incorporating preprocessing functionality into the Ahuora Digital Twin platform will provide the best workflow for the largest range of user archetypes. Even in cases where the external processing before or after is necessary, the preprocessing functionality will still make integration easier.

Chapter 8

Conclusions

The research conducted in this project has identified where the key areas of work are in developing a Digital Twin Platform for Ahuora. This has resulted in a framework being proposed for the system, wherein the Digital Twin Platform is built on top of a data processing pipeline and a simulation platform. It has been tested via prototyping to show that it is appropriate for those who are involved in both design and operation of the factory, from a data or chemical engineering perspective.

This has been implemented in the Ahuora Digital Twin Platform for steady-state modelling. The ability to store solve history was added to visualise the results of simulations, and functionality to preprocess data was added to make it possible to convert input data to the format required in chemical modelling.

As a test case, a model of a heat pump dryer was created and the platform was used to simulate the dryer's performance. This showed that the platform is capable of simulating a factory's performance, and identified some edge cases where care must be taken to specify the model in a way that it can be reliabily solved.

This work addressed some of the main challenges in implementing digital twins in the industry, namely, the complexity and cost of building a Digital Twin system from scratch. Through the continued development of Digital Twin Platforms as specified in this report, Project Ahuora's goal to decarbonise the process heat sector will be fully realised.

8.1 Future work

Future work could focus on adding support for dynamic models, hybrid modelling, and optimisation to the Ahuora Platform. Then the live data processing system could be developed to support these features. This would increase the usefulness of the platform for factory operation and control.

The research conducted so far has provided the context required to build a high-level roadmap of future development. This is provided in Appendix A.

Further work should also be performed on the feasibility of creating a standalone 'Deployment' version of the Ahuora Platform, specifically focused on control and optimisation. The research presented in this report implies that such a product would better fit the constraints of a real factory, in cases where cloud platforms may not be appropriate.

Bibliography

[1] World Economic Forum. Global Risks Report. January 2023. Accessible at https://www.weforum.org/publications/global-risks-report-2023.

[2] Ministry of Business, Innovation & Employment. Low Emissions Economy: Decarbonising Process Heat. August 2022. Accessible at https://www.mbie.govt.nz/building-and-energy/energy-and-natural-resources/low-emissions-economy/decarbonising-process-heat/.

[3] Wei Yu, Panos Patros, Brent Young, Elsa Klinac, and Timothy Gordon Walmsley. Energy digital twin technology for industrial energy management: Classification, challenges and future. *Renewable and Sustainable Energy Reviews*, 161:112407, 2022.

[4] M. Mazhar Rathore, Syed Attique Shah, Dhirendra Shukla, Elmahdi Bentafat, and Spiridon Bakiras. The role of ai, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities. *IEEE Access*, 9:32030–32052, 2021.

[5] Isuru A. Udugama, Carina L. Gargalo, Yoshiyuki Yamashita, Michael A. Taube, Ahmet Palazoglu, Brent R. Young, Krist V. Gernaey, Murat Kulahci, and Christoph Bayer. The Role of Big Data in Industrial (Bio)chemical Process Operations. *Industrial & Engineering Chemistry Research*, 59(34):15283–15297, August 2020. Publisher: American Chemical Society.

[6] Andrew Lee, Jaffer H Ghouse, John C Eslick, Carl D Laird, John D Siirola, Miguel A Zamarripa, Dan Gunter, John H Shinn, Alexander W Dowling, Debangsu Bhattacharyya, et al. The idaes process modeling framework and model library—flexibility for process simulation and optimization. *Journal of advanced manufacturing and processing*, 3(3):e10095, 2021.

[7] Michael L. Bynum, Gabriel A. Hackebeil, William E. Hart, Carl D. Laird, Bethany L. Nicholson, John D. Siirola, Jean-Paul Watson, and David L. Woodruff. *Pyomo - Optimization Modeling in Python, 3rd Edition*. Springer, 2021.

[8] Andrew Connolly, Joseph Hellerstein, Naomi Alterman, David Beck, Rob Fatland, Ed Lazowska, Vani Mandava, and Sarah Stone. Software engineering practices in academia: Promoting the 3rs readability, resilience, and reuse. *Harvard data science review*, 5(2), 2023.

[9] Claes Wohlin and Per Runeson. Guiding the selection of research methodology in industry–academia collaboration in software engineering. *Information and software technology*, 140:106678, 2021.

Appendices

Appendix A

Roadmap of Future Development

Figure A.1 breaks future development down into phases. Each phase is broken down into subtasks, identifiying the key challenges to overcome.

Phase 1 provides the core functionality: ingesting data, solving the simulation, and displaying the results. This is the minimum viable product, and is completed as part of this dissertation.

Phase 2 adds support for physics based modelling, optimisation, and control. The key challenges anticipated, including specifying input of the the time domain, holdup, and visualisation, come from the research conducted in Section 5.3. Likewise, the key challenges from optimisation and control originate from the research in Section 5.5.1. Based on research from the literature review and Section 5.4.1 surrogate modelling moves beyond IDAES alone, and may also include online learning methods, specified in the figure as "Live Surrogate Modelling". This was seperated out as Phase 3, because it is a wider goal, and less defined at this stage.

Phase 2 and 3 are not developed in this project, but this long-term roadmap is a key research outcome. It proves that development has been done in an engineering context, with a clear understanding of future requirements and an anticipation of further work.

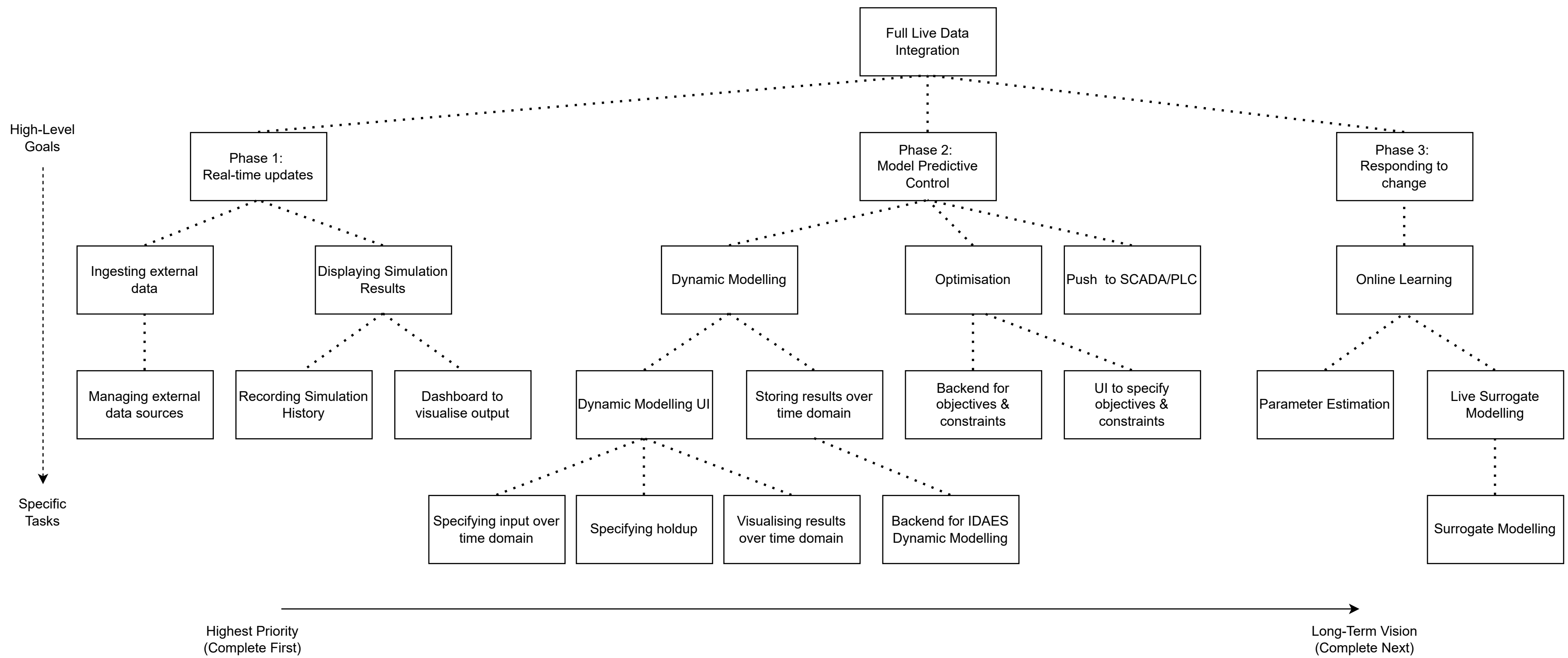

Figure A.1: Roadmap of future development, broken down into tasks. This extends beyond the scope of this project, into the broader vision of the Ahuora Digital Twin Platform. Each task is a key milestone in creating an industry-ready live data processing platform.

Appendix B

Project Proposal

The project proposal is included as an appendix on the following page.

ENGEN582-24X Honours Research and Development Project Proposal

Integrating streamed sensor data into a distributed model of a complex system

Bert Downs
Supervised by Tim Walmsley, Mark Apperley

February 2024

1 Proposal Summary

Process Engineering is a discipline of Engineering that focuses on designing, controlling, and operating systems that convert raw materials into useful products. A key challenge in this field is ensuring that factories and systems can maintain optimal operating conditions, even with varying external conditions such as weather, electricity price, and feedstock quality. One method of doing this is to incorporate real-time data from on-site sensors, to report abnormal conditions. However, while this can report inconsistencies, this cannot predict the result of the conditions on downstream processes. This is a gap in current research.

It is proposed that by implementing a mathematical model of the process engineering system and integrating live sensor data, predictions of downstream processes can inform decisions on adjusting to varying external conditions. This will ensure that factories can maintain optimal operating conditions.

This will be done by developing APIs to integrate live data into the Ahuora Adaptive Digital Twin Platform and working on implementing dynamic model analysis. Project Ahuora will provide resources required for a test case with a heat pump system.

This project has aspects of Sustainability and Vision Mātauranga through contributing to the aims of Project Ahuora, which focuses on decarbonisation through efficient energy usage.

2 Background

The manufacturing economy takes up 12% of New Zealand's GDP. Manufacturing has intense overseas competition, because New Zealand's small domestic market makes it hard to produce things at scale. At the same time, there is a growing consumer and political push to become carbon neutral, by reducing energy usage or switching to sustainable alternative. As fossil fuel and energy prices rise, efficient energy usage is increasingly becoming an economic advantage as well.

Downtime also has a significant effect on manufacturing sustainability and efficiency. When a factory line needs to be stopped, process heat is wasted. Restarting a factory can involve a significant increase in energy usage.

Digitisation and automation are increasingly being relied on in order to make factories efficient enough to compete. Digitisation can help to decrease the cost of running a factory by replacing manual tasks, detecting abnormal conditions to prevent downtimes, and helping to optimise processes. This is becoming more prevalent with the rise of "Industry 4.0" and the Internet of Things. Factories are able to gather more data on operations than ever before. However, the technology to make use of the gathered data is still developing.

Project Ahuora, a research team based in Waikato, Auckland, and Massey Universities, is working on methods to make better use of this technology. Their aim is to "develop a novel energy-technology platform based on adaptive digital twin technology" [1]. Digital Twin technology involves creating a digital model of a physical system. This allows for virtual simulation and testing, which is often not possible in physical sites. *Adaptive* Digital Twin technology means the model can also adapt to changing real world environments, which is made possible by integrating live data from IoT sensors back into the digital model. This enables for new types of analysis. For example, when abnormal conditions are detected in a factory, the digital twin can be used to simulate different mitigation strategies, and the impacts on downstream processes.

In order for these new analysis methods to be used, the IoT data needs to be connected and represented on the Digital Twin. Project Ahuora is developing software that can create mathematical models of a system, and therefore the IoT data needs to be represented as mathematical constraints on the model. This project will solve this issue.

3 Overall Aim of the Project

How can live data be integrated into digital twins in order to improve model and factory performance?

The aim of the project is to research novel methods to integrate live sensor data into a complex mathematical model. This comes with a number of specific objectives, which are modeled on similar research in related fields[3]. These include:

- Finding appropriate methods to gather relevant data;
- Creating a static and dynamic knowledge base and relevant querying methods;
- Researching and developing methods to relate raw data to properties of a mathematical model;
- Designing user interfaces that allow for access, setup, and use of sensor data in mathematical modelling.

This research will be done by developing an extension to the Ahuora Adaptive Digital Twin Platform.

4 Research and Development

The extension to the Ahuora Adaptive Digital Twin Platform will be the main deliverable of this project. The initial stage of the project will involve studying existing literature, and some novel research, in order to find appropriate methods to integrate live data into the mathematical model. This will be implemented into the Ahuora Digital Twin Platform, using industry standard practices to create maintainable code and a user-friendly interface. The result of this development will be used as a case study to prove the efficacy of the system.

4.1 Literature Review and Prototyping

The literature review will provide an overview of existing research, in order to determine what has already been done in the field, or in related fields. This will be used to plan a strategy to design and prototype the IoT systems, and the integration with Ahuora's mathematical model. The prototyping will be conducted in conjunction with the literature review, which will enable the literature review to be focused on the largest barriers to implementation.

4.2 Development

After being informed by the literature review and initial prototyping, development on the integration with the Ahuora Digital Twin will commence. This is expected to take the majority of the project's time. The Ahuora Digital Twin platform has to be modified in order to support different custom "extensions" such as live data processing. Then a platform to ingest data from IoT sensors needs to be created. The data will then need to be converted into "constraints" on the mathematical model, using machine learning techniques or other means, as influenced by the literature review. Finally, a user interface needs to be created in order to enable easy implementation of IoT in a variety of process systems.

Once the development is nearing completion, it can then be tested in a real-world case study. For example, a heat pump can be modelled in the Ahuora Platform, and then live temperature data can be imported in order to predict efficiency, energy usage, report abnormal conditions, etc. Some hypotheses that may be investigated include:

- *Integrating live sensor data into mathematical models will improve predictions of abnormal conditions in a system*
- *Using a mathematical model and live temperature data, other thermodynamic properties of a process can be accurately estimated.*

The exact details of this testing and research phase will be decided based on learnings from the literature review and initial prototyping.

4.3 Additional Information

A Gantt chart with an expected timeline can be found in Appendix A.

5 Resources

During the initial phases of the project, the main resources required are expertise in Chemical Engineering and Software Development. This is available through others involved in the Ahuora project and the Computing and Mathematical Sciences department at Waikato University. Additionally, development resources and space has been allocated as part of Project Ahuora. Access to simple IoT devices, such as temperature and humidity sensors, for initial prototyping has been provided by Project Ahuora. Finally, Ahuora has provided access to the source code of the Digital Twin platform to develop on.

During the later stages, more practical testing will be required. Access to a physical process will be needed to enable research to be conducted on the efficacy of predictions. The details of this have not yet been determined, however this may be done in conjunction with other Honours students who are working on heat pump testing, as creating a digital twin of a heat pump will provide a good controlled real-world test case. Space and time on-site will be required to conduct this testing.

6 Sustainability and Vision Mātauranga

The industrial sector of New Zealand uses around 170 PJ of energy every year [4]. This is around 30% of the total energy usage. Being able to optimise processes, such as by reducing waste heat or downtime, is able to make a significant impact in the total energy usage of the country. Less energy usage means there will be less reliance on carbon-based fuels, and will speed up the process of electrification. Energy production and consumption also has a significant impact on the land and ecology of New Zealand.

As this is project based around decreasing energy usage, it is related with the “Te Taiao” theme of Vision Mātauranga. The environment is an important part of New Zealand’s identity and history, particularly for Māori. Sustainable resource management is specifically mentioned as a point of focus in New Zealand’s Vision Mātauranga policy [2], which includes sustainable energy usage.

References

[1] 2024. Ahuora - Center for Smart Energy Systems, University of Waikato. https://ahuora.waikato.ac.nz/.

[2] 2024. Vision Matauranga: Unlocking the Innovation Potential of Māori Knowledge, Resources and People, Ministry for Business, Innovation, and Employment. https://www.mbie.govt.nz/assets/vision-matauranga-booklet.pdf.

[3] Dagimawi D. Eneyew, Miriam A. M. Capretz, and Girma T. Bitsuamlak. Toward smart-building digital twins: Bim and iot data integration. *IEEE Access*, 10:130487–130506, 2022.

[4] 2024. Ministry for Business, Innovation, and Employment. https://www.mbie.govt.nz/.

Appendices

A Gantt Ghart

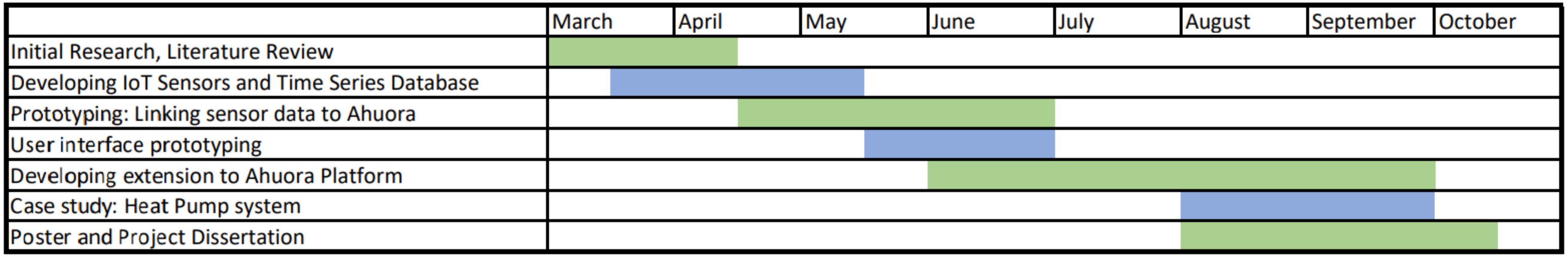

Figure 1: Gantt Chart with estimated project progress.

This Gantt Chart provides an estimated project timeline, and provides a standard to measure progress against. Many things can be worked on concurrently, as they will be related tasks. A significant amount of time will be spent prototyping, but the longest task will be developing the extension to the Ahuora Platform, because of the level of detail required.

Appendix C

Literature Review

The literature review is included as an appendix on the following page.

ENGEN582-24X Literature Review

Integrating streamed sensor data into a distributed model of a complex system

Bert Downs
Supervised by Tim Walmsley, Mark Apperley

April 2024

“Live Data integration in Industry 4.0”. Generated by Bing AI.

Contents

1 Introduction

1.1 Background

The World Economic Forum ranked "Failure to mitigate climate change" as the number one threat to the world in the next ten years [1], due to the effects climate change has on extreme weather events, biodiversity, and climate-vulnerable economies. Decarbonisation is a crucial step in mitigating climate change. In New Zealand, process heat accounts for 8% of total greenhouse emissions [2].

Digital Twin (DT) technologies can inform efficient scheduling, optimisation, maintenance, and retrofit procedures. Thus, DT technology has the potential to decrease process heat emissions, particularly in site-wide and site-edge applications [3].

1.2 Purpose

This literature review is being conducted in conjunction with a software development project. The project is to develop a system that ingests real-time sensor data from chemical processing plants and uses a mathematical model of the plant to estimate other characteristics in real time. The project proposal may be found in appendix **??**. However, the project proposal is very high-level, explaining *what* the project is but not *why* the project is necessary or *how* the project can be implemented.

This literature review supports the software development project by answering the *why* and *how* questions about the project. It reviews the current state of research integrating real-time industrial sensor data into digital twins. It confirms and updates the overall aim and specifications of the software development project, by identifying knowledge gaps requiring further research, and methodologies or tools that would support development.

1.3 Scope

This literature review will broadly cover topics related to Big Data including Data Collection in Factories, the Internet of Things, Data Stream Processing, and Machine Learning for Data Streams. It will focus more specifically on Digital Twins (DTs) and Chemical and Process Engineering (CAPE).

1.4 Research Questions

The sources used in this literature review were selected to focus on the following research questions:

- *RQ1*: Where is real-time data collection most used in industrial processing facilities, and where is there potential for it to be used better?
- *RQ2:* What techniques and processes are required to turn real-time data into useful insights that can inform and automate decisions in an industrial facility?
- *RQ3:* How can real-time data be integrated into a Digital Twin that is based on mathematical, first-principles simulation techniques?

RQ1 focuses on the general goals and purpose of the project. By reviewing case studies and other literature that apply real-time data collection in an industrial processing facility, an overview of the most applicable areas can be achieved, enabling the project to focus on areas where the greatest improvement over existing techniques is expected.

RQ2 focuses on the pipeline to process live data in an industrial factory. This will be used to define the technical environment the software development project will pull from and contribute to. From this, the inputs and the outputs of the software development project can be clearly defined, enabling the project to be clearly situated as a part of the existing data processing pipeline.

RQ3 discusses the major technical challenge this project will need to overcome. Real-time data is by nature dynamic and imperfect. Mathematical and first principles simulations are well-defined and cannot always handle variation, missing data, and inaccuracies. By discussing methods that others have used to process real data using mathematical simulation techniques, areas for future work can be identified. This can be used to inform the development of the core algorithms of the software development project.

1.5 Methodology

Digital Twins are by nature highly integrated and context-dependent. Thus, the research and technology underpinning them can differ greatly depending on the application. Because this project is focused on industrial live data, most sourced literature is related to Digital Twins in industrial settings. This literature will identify the specific technologies and techniques used to build digital twins in industrial settings. The technologies and techniques are then reviewed in greater depth outside of the context of digital twins, to gain a general understanding of the limitations and strengths of each. This can then be used to draw conclusions about its use in Industrial Digital Twins.

Because this literature review will primarily be used to inform future development, some sections include a review of relevant products and tools that apply the techniques identified. These products have been identified based on their applicability to industrial processes and settings. As a convention, sources from product websites and documentation are referenced as footnotes, to differentiate from academic citations included in the bibliography.

The three research questions identified each focus on very different areas of industrial data processing and analysis, spanning the spectrum from big-picture needs down to challenges in developing specific algorithms. Each could be the subject of a literature review on its own. Additionally, Big Data, Artificial Intelligence, and Digital Twins have recently become very popular topics in academic literature. The diversity of sources and availability of research means there are a large number of papers that are related to this literature review. As such, it is a challenge to filter out all research on a topic to find those that are the best fit for this project - there are simply so many articles and technologies that are relevant and could be used. Care has been taken to ensure that the sources used cover the spectrum of available technologies and techniques, on each level of analysis. However, this cannot be considered a comprehensive review of all relevant research, algorithms, and implementations.

2 An Overview of Digital Twins

The concept of Digital Twins has been rising in popularity in recent years and is used in a number of industries, including Manufacturing, Energy, Transportation, Healthcare, and Networking [4]. The concept of a “Digital Twin” refers to a simulation of something in the physical world, which is kept up to date with a physical system using real-time data [5].

2.1 Purpose of Digital Twins

There are number of reasons Digital Twins may be used. These generally fall into one of the following categories:

- **Testing**. The digital twin can then be used as a test environment, as modifications can be applied to the digital twin before applying them to the physical system [6].

- **Validation**. Data can be collected to validate that a factory is working correctly. Data can also be collected to validate that a digital model accurately reflects the conditions of the factory [7].

- **Diagnostics**. Live data feeds from each stage of operations can help diagnose issues, by tracing the problem back to where it first occurred. They can also be used to measure the health of the physical system as the digital twin changes over time[7].

- **Fault Classification**. This goes a step further than diagnostics, as fault classification involves interpreting the data to find an exact cause and type of fault [8].

- **Automation**. A software system can be set up to trigger actions in response to changes in process conditions. For example, the system can initiate corrective measures autonomously[3].

- **Schedule Optimisation**. Data from operations can be used to optimise plant schedules, such as avoiding peak power usage when the electricity price is high [9].

- **Control Optimisation**. Live data can be used to identify what conditions provide the highest cost-yield ratio [10] [11].

- **Asset Optimisation**. Plant data can be useful when evaluating the return on investment of capital expenditure during expansion, retrofit, or replacement of equipment [12].

2.2 Conceptual Model of Digital Twins

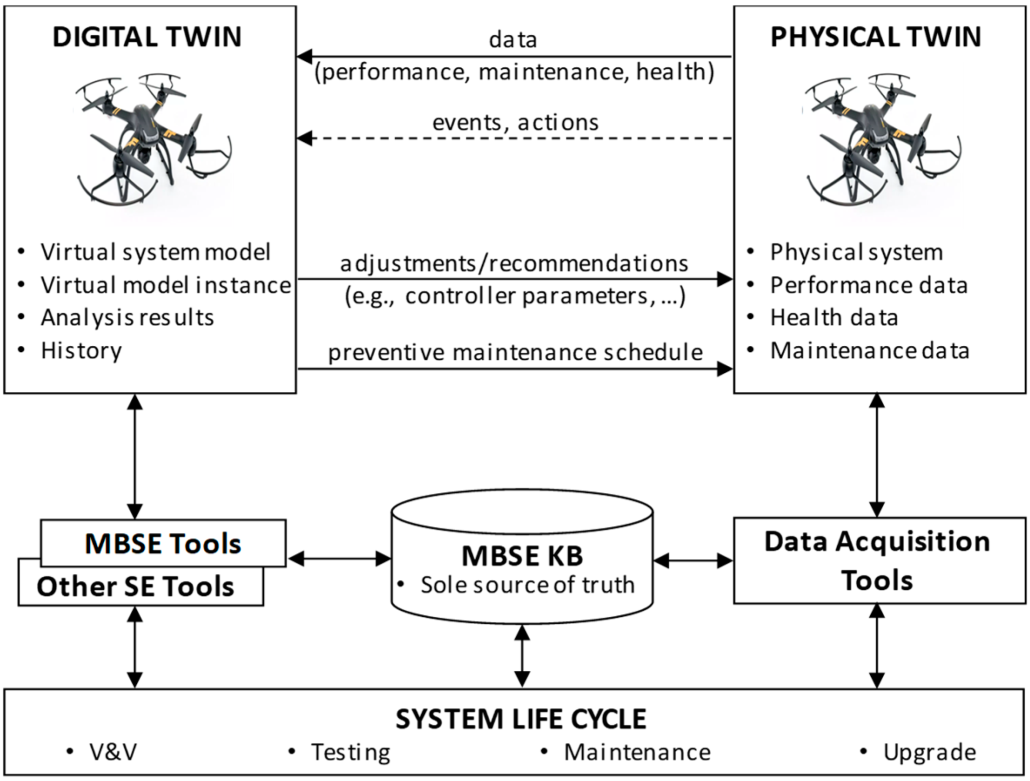

Figure 1: Digital twin concept, using the Model-Based Systems Engineering Framework (Madni et al. [7])

Madni et al. analysed the concept of digital twins from a Model-Based Systems Engineering (MBSE) perspective [7]. Figure 1 shows how the physical twin is able to send information to the digital twin, such as performance data and events. The digital twin is then able to be used to calculate recommended actions and inform decisions using simulated data beyond that which is physically possible to collect. It is able to do this by using the MBSE Knowledge Base, a database that records simulation results based on real-time data. As the digital twin is a model of a specific instance of a physical asset, rather than a generic model, it is more accurate than traditional approaches.

Yu et al. [5] classified DTs on varying levels of detail. They identified that there was a lack of research into higher-level digital twins, which contained accurate process simulations and operated at a larger scale over an entire factory or site. They also identified that much less research has been done on process and energy Digital Twins in comparison to their mechanical counterparts.

2.3 Conceptual Model of Live Data Processing

There are many steps to processing live data. One framework others have used to view the processing of live data is the Knowledge Pyramid [13]. Also referred to as the Data-Information-Knowledge-Wisdom hierarchy or Data Pyramid, this consists of four layers of abstraction:

- **Data**: Discrete, objective facts about something. Simple values or measurements.
- **Information**: Data in context - structured to provide a useful story or meaning, such as linking to a specific location, or time.
- **Knowledge**: Something that provides the why or how - explaining the relationships between information.
- **Wisdom**: Placing knowledge in a broader context, and applying it to different or novel situations.

This hierarchy is used to illustrate the progression from raw data to meaningful insights and actions [14]. It is analogous to the progression of live data in a digital twin, which starts out as raw sensor data and eventually is applied to improve the physical system. Figure 2 illustrates this flow from data to meaning as follows:

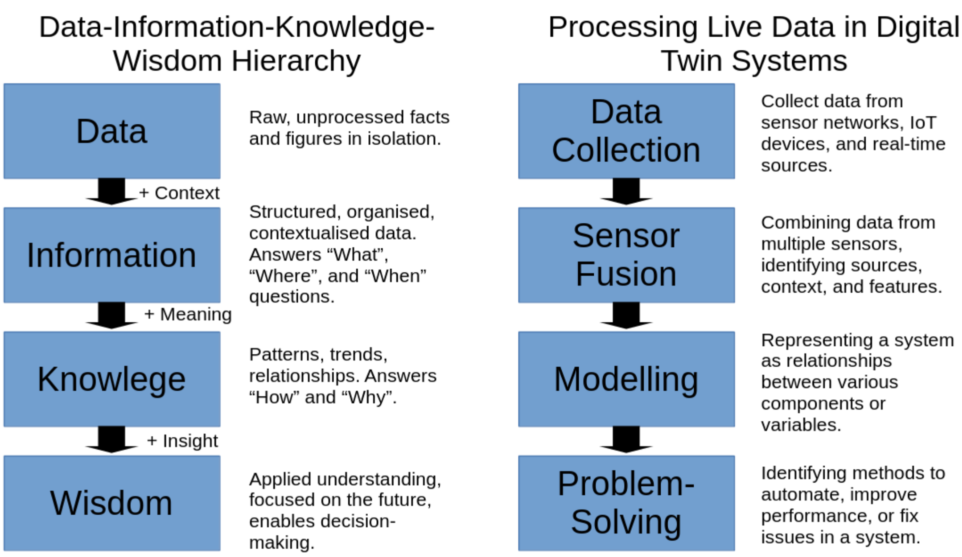

Figure 2: Comparing and applying the Data-Information-Knowledge-Wisdom Hierarchy with the flow of data through Digital Twin Systems.

- **Data Collection**: Raw Data must first be collected from the industrial process. This forms the basis of all future insights but is meaningless in isolation. This data is collected in real time.
- **Sensor Fusion**: Context is given to the data, such as type of measurement, time, and/or location. More advanced sensor fusion techniques can be used to integrate data from multiple sensors to overcome limitations such as noise, uncertainty, or individual sensor limitations. This can be viewed as converting the "Data" into "Information".

- **Modelling**: Modelling is the core of the Digital Twin. Modelling creates a "representation or abstraction of the system" [15], linking distinct pieces of information together by creating or finding relationships between them. This means modelling can be viewed as helping to convert "Information" into "Knowledge". Models enable the simulation of hypothetical scenarios.

- **Problem-Solving**: Wisdom is the least well-defined portion of the data pyramid. However, it is generally seen as future-focused, applying current knowledge to make decisions. In Digital Twins, this can be seen as the process of applying live data models in order to find insights that aid in or make decisions. This could include automating actions, optimising performance, or identifying faults in advance.

The remainder of this literature review analyses literature related to each level of data processing in this pipeline in turn. This provides an end-to-end understanding of how data is processed in a digital twin. Section 3 will cover Live Data Collection, section 4 reviews literature related to sensor fusion, section 5 covers various methods of modelling with live data, and section 6 considers how digital twins can be used to solve various real-world problems.

3 Live Data Collection

The first part of processing live data in a Digital twin is to collect the raw data from the industrial processes. This section discusses the data sources available in industrial plants, and what technologies enable collecting data from them. This information is used to discuss and update the requirements the software development project will have.

3.1 Data Sources

In Industrial Process operations, there are a number of data sources, as shown in fig. 3 [16]. These include:

- **Process Data**, which is data from on-site sensors and control systems. This can include the state of operations, the current control configuration, temperature, humidity, vibration, position, or chemical composition. This data is collected in real-time, can be automatically analysed, and is the main focus of our study.

- **ERP Databases** contain time-stamped information about usage, breakdowns, plant configuration, capacities, and external factors such as the supply chain or product specifications. This data is also collected in real-time, and can usually be automatically analysed, and thus is also central to our study.

- **Engineering Data** refers to datasheets and models, usually created during plant design. This is not a live data feed, but can be used to inform the development of sensor fusion algorithms as discussed in section 4. Factory models can also be used for real-time simulation, and this will be discussed in section 5.

- **Manual Logs** refer to manually recorded information about the plant operations. In general, these cannot be automatically analysed.

- **Quality Logs** refer to factory samples that are taken and analysed offsite. These results are not a live data source, but real-time algorithms based on plant conditions and past quality logs could be developed to target plant conditions that provide the highest quality results.

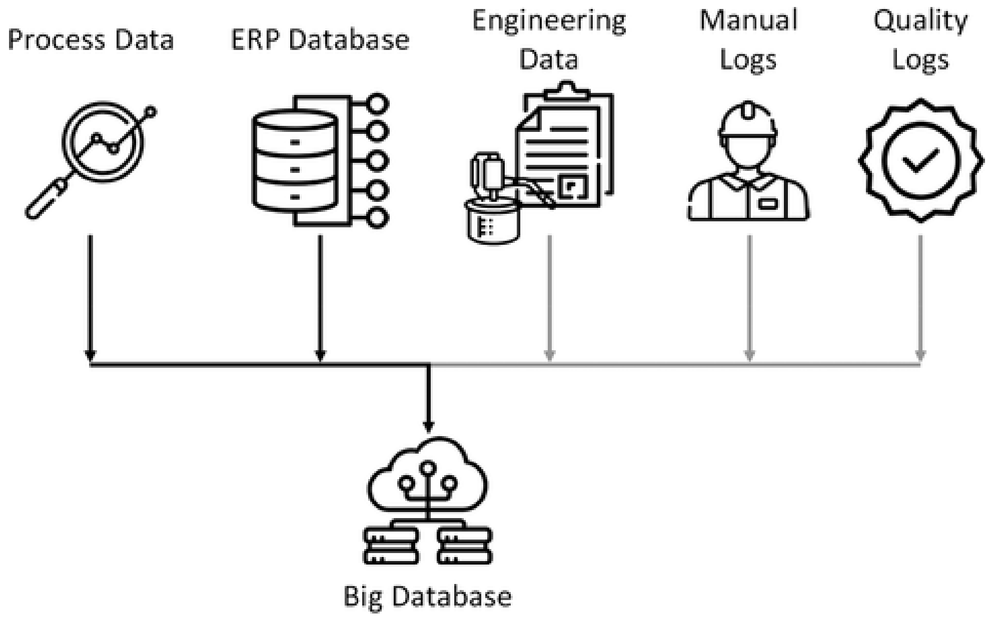

Figure 3: Typical data repositories in an industrial plant [16].

3.2 Technologies

There are a number of tools and technologies that are used to collect live data from factories. It is important to identify what industry tools exist and are already being used. Understanding these tools provides a framework future tools can build on and avoids recreating work that is already done.

IoT Communication tools provide standardised ways to interact with a variety of IoT devices. These include:

- MQTT [1], which is a publish/subscribe system that enables other services to communicate with IoT devices
- Modbus [2], an open standard to connect to Programmable Logic Controllers (PLCs) in factories.
- Telegraf [3], which can collect metrics from various protocols, transform, aggregate, and filter them, and send them to a centralised database.
- OPC [4] is a standard that abstracts PLC protocols such as Modbus into a standardised interface. This enables interoperability between devices from different vendors.

In an industrial context, Supervisory Control and Data Acquisition (SCADA) systems are used to gather and process data. It is industry standard to use SCADA Systems to help chemical plants work autonomously, optimally, and efficiently [17]. Examples of commercially available SCADA systems include (in no particular order) WinCC, Genisis64, iFix and Cimplicity. These systems perform a similar role to IoT-based solutions, and data from either can be used to inform a Digital Twin.

3.3 Discussion

From the above sources reviewed, it is clear that data is readily available from industrial sites. Therefore, in the context of the software development project, it is most appropriate to use existing sources

[1] https://mqtt.org

[2] http://www.dankohn.info/projects/Fieldpoint_module/Open_ModbusTCP_Standard.pdf. See also https://www.modbustools.com/modbus.html.

[3] https://www.influxdata.com/time-series-platform/telegraf/

[4] https://opcfoundation.org/

of data rather than deploying additional sensor networks. Sensor data is not the only source of live data - so the project should also process external information such as usage demands. Furthermore, there are many different platforms and protocols that data is collected from, so the ability to support multiple distinct data sources is a key requirement for the project. In particular, integration with SCADA systems will be required.

Because the project will need to aggregate so many different sources of data, it will also need to be performant enough to process the data in real-time, and reliable enough to do so without interruptions. It will need to be scalable enough to support large and small factories.

4 Converting Data to Information via Sensor Fusion

Once access to real-time data is available, the next step to processing is to make the data meaningful to a digital twin system. This is an application of the field of sensor fusion. Data needs to be tagged with relevant context information, such as source and timestamp information [16], and data streams need to be aggregated together in a way that enables extracting meaning. This is the process of converting raw data into information.

4.1 Sensor Fusion Techniques

Sensor fusion is the process of combining multiple different sensors' data in order to come up with some more meaningful parameters[18]. For example, a Virtual Reality headset can fuse accelerometer, gyroscope, and magnetometer data together to find the device's rotation - even though there is no one sensor that can perform this task [19]. In essence, this creates a "digital twin" of the headset with the same orientation parameters in the virtual environment that the physical headset has in the real world.

When building a digital twin from sensor data, it is unlikely that the measurements we are able to get will always be exactly the same as what is modelled in process dynamics. For instance, it may be hard to get the temperature of a material stream, but it may be possible to get the temperature of the pipe the stream is in - and then sensor fusion techniques can be used to estimate the temperature of the material inside based on this and other data.

Alam et al. [20] explored how IoT and data fusion technologies are related. They identified three methods of data fusion:

- Probability-based methods, such as Bayesian Analysis, Markov chains, and the Monte Carlo Method. This fuses sensor data by mathematically modelling the relationship between the various sensor readings, and weighting the probability of inaccuracies in each in order to find the most likely solution.
- AI-based methods, ranging from fuzzy logic to Artificial Neural Networks. This works better than Probability-based methods when modelling complex interactions.
- Theory of Evidence-based methods, which are a generalisation of Bayesian Probability theory that is not as complete but works well in practice, and is faster than AI methods.

Bayesian Analysis was used by Renganathan et al. [21] to fuse field measurements with simulation data. Figure 4 shows the overall methodology. Bayesian Inference is used to find the value that has the highest likelihood of being correct. The main limitation identified with Bayesian Interference is that specifying the appropriate hyperparameters, such as the prior probability distribution and the likelihood function. This requires a detailed understanding and modelling of the data source.

Fuzzy logic is an extension of boolean logic to express the concept of partial truth - allowing for representing values in between true and false. Sensor values can be expressed in terms of fuzzy logic,

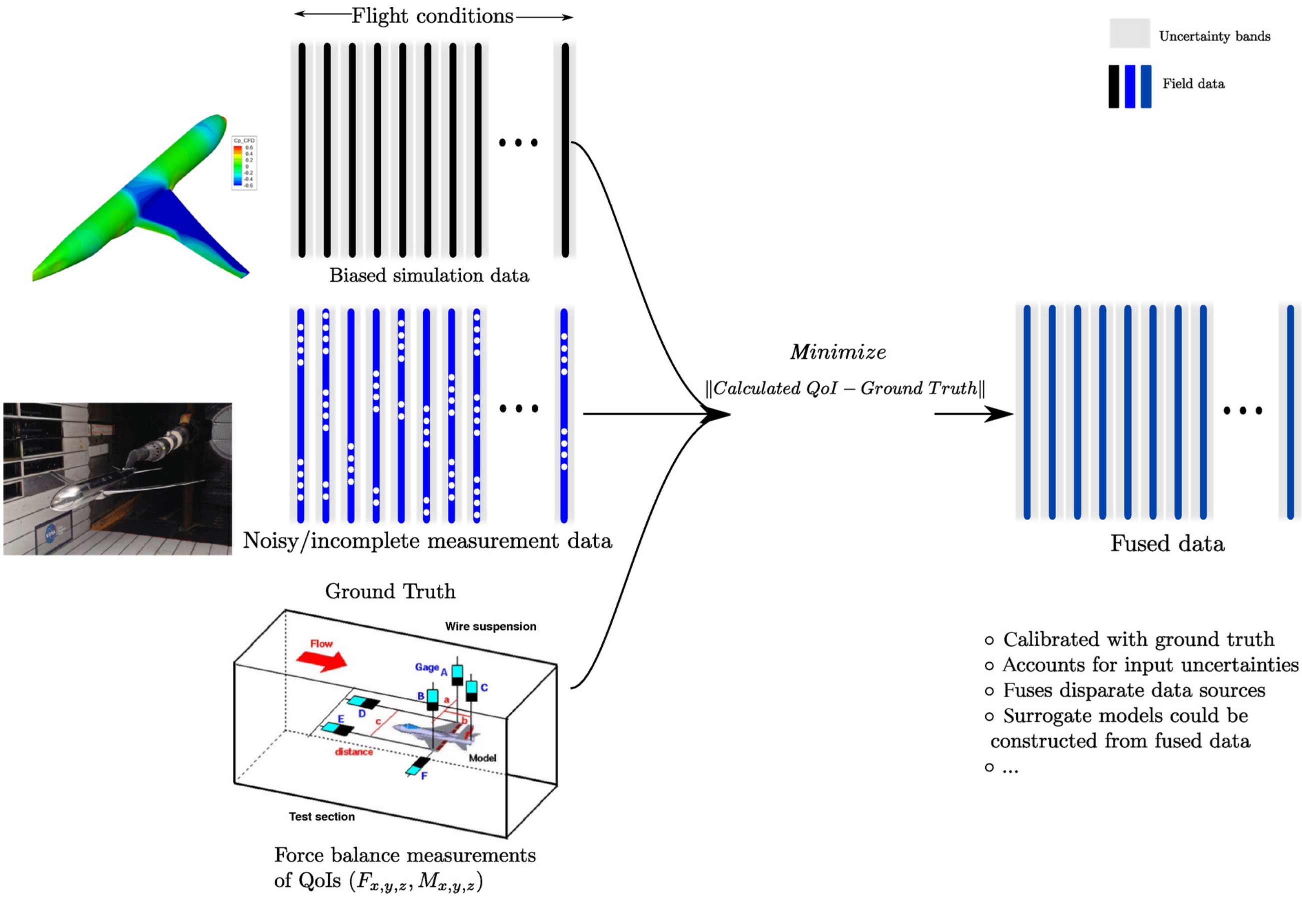

Figure 4: Sensor Fusion of data from a simulated aerodynamic model and field measurements. [21].

and then fuzzy rules can be used to fuse the signals. This multi-sensor fusion increases accuracy because decisions are not based completely on any one input [20].

Lermer and Reich [22] used fuzzy logic to augment Artificial Neural Networks, by using fuzzy rule sets to generate additional parameters. This improves training when there is little data, as the fuzzy rule sets can encode knowledge from domain experts.

Artificial Intelligence methods can be used for sensor fusion, such as to predict the ground truth from multiple competing sensors, or using Principle Component Analysis to reduce the dimensionality of the data [23]. However, when AI is used for sensor fusion, it is often also used for modelling and decision making [22]. Munir et al.[24] classified these different methods as *Sensor Fusion*, *Knowledge/Feature Fusion*, and *Decision Fusion*. Using Artificial Intelligence for Knowledge/Feature Fusion is a type of modelling and will be discussed in section 5. Using Artificial Intelligence for decision Fusion will be discussed in section 6.

Theory of Evidence models are based on Dempster-Shafer theory (DST). The main advantage Theory of Evidence models have over similar techniques such as Bayesian Analysis is that they include methods to model missing values and incomplete data[20]. Saeidi et al. [25] used DST to process data from Light Detection and Ranging (LiDAR) systems by fusing it with multispectral satellite imagery to convert it into land types, and found it was cost-effective and time-effective, because of the ability to use logical reasoning to develop the algorithm to process the data. This also means it is more interpretable than AI-based methods.

4.2 Technologies

Regardless of the methods used for data fusion, the first step is to gather data from multiple sources and tag data with additional metadata such as time or location. Data-gathering tools aggregate data from various IoT devices into one central database. Some platforms include:

- Influxdb [5], a time-series database that integrates well with a variety of platforms and protocols for storing IoT data.
- DynamoDB, [6], a NoSQL database that works well for storing flexible data from IOT sensors.
- Apache Kafka [7], an event streaming platform that enables storing and manipulating data streams.
- SCADA systems also include functionality to tag and store data, or to pass it on to other systems for further processing.

4.3 Discussion

The software development project will be able to make use of data streaming platforms such as Apache Kafka and database systems such as Influxdb to enable the collection and processing of data in real time. These technologies should be chosen to allow more advanced methods of sensor fusion, such as Bayesian Analysis, Fuzzy Logic, Artificial Intelligence, and Theory of Evidence models to be used to process raw sensor data. The exact method to use will depend on the application. Hence the project should include preprocessing functionality that is adaptable to generic or custom implementations of these algorithms.

Sensor fusion techniques also help to increase the accuracy and reliability of source data, as data from multiple sources are cross-validated. This enables higher-level modelling and simulation algorithms to be more precise, enabling better decision-making.

5 Turning Information into Knowledge via Modelling

5.1 Modelling and Simulation

Modelling is the process of defining relationships between entities. There are a number of different methods and platforms for modelling a digital twin system. This literature review will focus on those most relevant to processing live data for industrial digital twins.

The primary purpose of modelling is to enable simulation, which is the process of using a model to generate predictions of what might occur under certain conditions. De Paula Ferreira et al. [26] conducted a literature review of simulation in Industry 4.0. Simulation is used in many contexts, but some approaches were identified that are applicable to the context of processing live data. These include Analytical methods such as Agent-based modelling, Discrete Event Simulation, System Dynamics, and Data-Driven methods such as Artificial Intelligence. The literature review identified that Hybrid Modelling and real-time simulations provided opportunities for further research. They also identified that incompatibilities between simulation platforms were a major barrier to use.

The following sections first discuss Analytical modelling methods, including standards for model formulation, and Data-Driven modelling. Then research into hybrid modelling approaches and real-time simulations will be reviewed.

5.2 Analytical Modelling methods

Agent-based modelling is the process of defining rules for individual agents, or entities in a system. When simulated, emergent properties can be identified from the interactions between the entities. Abar

[5] `http://influxdata.com/`
[6] `https://aws.amazon.com/dynamodb/iot/`
[7] `https://kafka.apache.org/`

et al. [27] conducted a review of tools that provide functionality for agent-based simulation. In industry, agent-based modelling is most helpful for logistics, to optimise factory plans, and for 3d simulation. However, it is not usually appropriate for modelling chemical processes.

Discrete Event Simulation is the process of simulating a process of events as if they occurred one at a time. This method of simulation is used in industrial settings primarily related to the optimisation of scheduling and supply chains [28] such as to ensure full capacity utilisation of processing equipment [29].

System Dynamics models the change of a system over time, by modelling the relationships between components of a system. In an industrial setting, Process Systems Engineering (PSE) uses mathematical modelling and data analytics to model a chemical system. Using Mathematical Modelling in Process Systems Engineering can often be considered an application of System Dynamics. System Dynamics is useful for Process Control and enables modelling interactions such as chemical reactions or thermodynamic interactions such as thermal coupling [30].

5.2.1 Model Formulation Standards

Blaž Rodič [31] conducted several case studies on the use of simulation modelling in the industry, and found that standardised methodologies would allow better integration between platforms. Consistent ontological models allow higher-level applications to interact with different data types.

There are some efforts in the industry to come up with standardised data models. Some are open, enabling easier adoption across platforms, and some are proprietary, but provide high compatibility with other products made by the same company.

Fiware[8] is an open-source platform that enables accessing, processing, and storing data using standardised NGSI-LD models. These models have standard definitions for a variety of entities (such as buildings or IoT devices), properties (such as temperature) and relationships between entities [32]. These can be extended to handle specific use cases[9] but provide the most benefit if they are standardised.

A study on NGSI modelling for smart cities found that the added context made the data more understandable and navigable [33]. It is likely the concepts are applicable to other digital twin situations.

Thingworx [10] is a similar platform to Fiware, that uses a proprietary information model rather than NGSI. It is made by PTC and caters towards the Industrial IoT market.

Siemens Insights Hub [11] also provides a similar service. It is focused on connecting to PLCs in factories and creating a data model that can then be analysed and used in other services.

Modelica [12] is an object-oriented language for modelling cyber-physical systems. It enables defining objects and mathematical relationships between objects. The Modelica specification language is used to model many different types of physical systems, including electrical, hydraulic, heat and fluid flow, state machines, and mechanical systems. [34] There are many libraries of models available which cover these systems to varying levels of detail. A number of commercial and open-source tools provide simulation environments that can run Modelica models.

5.2.2 Modelling Tools

Anylogic [13] is a modelling solution that supports Agent-Based Modelling, Discrete Event Modeling, and System Dynamics. A whitepaper published by The Anylogic Company [35] illustrates how all

[8] https://fiware.org
[9] https://ngsi-ld-tutorials.readthedocs.io/en/latest/understanding-@context.html#creating-ngsi-ld-data-models
[10] https://support.ptc.com/help/thingworx/platform/r9.5/en/
[11] https://documentation.mindsphere.io/
[12] https://modelica.org/
[13] https://anylogic.com/

three different simulation approaches can be integrated and simulated together using Anylogic. However, Anylogic is designed for logistical and economic modelling, and while custom logic can be programmed in Java it would not be as appropriate for modelling complex chemical processes.

MapleSim [14] is a similar commercial modelling solution. It supports Model Based Systems Engineering and Modelica models, enabling it to work with a wider variety of external services and libraries, including more complex processes.

5.3 Data-Driven Simulation Modelling

Machine Learning is also applied in order to predict output conditions of a complex industrial process, given readings of input conditions. One study created an Artificial neural network that was able to predict the power output, steam and exhaust flow rates, and other parameters of a coal biomass cofired combined heat and power plant based on input air, fuel, and water conditions [36]. This was shown to have an error rate of below 5% for most operating conditions.

5.4 Hybrid approaches

More recently, research has focused on hybrid modelling that combines mathematical models and machine learning models. There are two main ways of doing this [37]: Embedding a machine learning model inside a mathematical model, or constraining a machine learning model to more closely follow a specific mathematical formulation.

Embedding machine learning models inside a mathematical model is the more common method of hybrid modelling. Ceccon et al. [38] developed an open-source package named OMLT. It is able to convert Neural Networks and Gradient Boosted Trees into models in the algebraic modelling language Pyomo. This enables embedding a machine learning model into a mathematical digital twin in order to model system interactions that are too complex or expensive to solve from first principles. However, OMLT is only able to support machine learning models that can be written as a set of algebraic constraints. It currently supports Neural Networks and Gradient Boosted Trees.

The alternative approach, constraining a machine learning model to follow a mathematical formula, was discussed by Bikmukhametov and Jäschke [39]. They presented a framework to design physics-aware machine learning models, by passing the outputs of a first principles simulation as inputs into a machine learning model. They found that generally, this type of "feature engineering" improved results, but in some cases it did not. Nevertheless, the models created were much more transparent than models that only were trained on raw data, which helps when evaluation of algorithms is required to ensure consistent performance.

Modulus [15] is a new framework developed by Nvidia to train and deploy AI models that also follow physics-based principles. These models are trained based on simulation data and equations representing the first principles simulation of the model. A case study[16] used Nvidia Modulus to accurately model a Heat Recovery Steam Generator, without having to do full fluid dynamics calculations. This enabled a Digital Twin to be created that could accurately predict when corrosion occurs.

5.5 Live Modelling

Models can be informed based on historical data, or they can be updated and used in real time. Updating and simulating models in real-time is called Live Modelling. This technology is crucial for digital twins because the models in a digital twin must be updated in real time to reflect real-world changes in the system.

[14] https://www.maplesoft.com/products/maplesim/

[15] https://developer.nvidia.com/modulus

[16] https://www.nvidia.com/en-us/on-demand/session/gtcspring22-s41671/?playlistId=playList-c1c5c322-57be-46af-8841-e418a2f70c2c

Live Modelling can be viewed as a combination of conventional modelling and information gained from sensor networks. Simulations are run in real time based on the information gathered from sensor data, and the results of the simulation are stored. Lugaresi and Matta [40] noted that most live modelling applications in the industry focus on scheduling, so other applications of real-time simulation could be researched further.

Various commercial modelling solutions include functionality for live modelling. Simulink [17] is a modelling software with a focus on automation and live modelling. Its focus is more on robotics, artificial intelligence, and electrical and signal processing. Anylogic supports using live data for digital twinning on a logistical and operations level and provides a good case study on how live modelling can be used. A whitepaper on digital twins [41] used models based on live data from operation stages and status, and key performance and financial indicators. Their examples showed that while simulation and modelling can identify bottlenecks and constraints, live modelling could be used to manage solutions. While the whitepaper developed a higher-level scheduling and operations simulation, these same techniques could be applied to more technical simulation platforms.

5.6 Discussion

To best fit the requirements of the software development project, the models used need to be Live Models, able to update in real time to reflect changes in the physical system. A hybrid modeling approach would best suit this. Analytical modelling forms the mathematical basis of the project because it enables greater transparency. Transparency is important in the industry where reliability needs to be assessed and the model's accuracy needs to be validated. However, elements of machine learning still need to be applied. Online machine learning can adjust the parameters of the analytical model dynamically, to keep the model up to date with the physical system. The models chosen will also need to be performant enough to run in real-time. This means choosing the model resolution is crucial - if the system is modeled at too high of a level, accuracy will be sacrificed. However, if the model is very detailed it may take too long to process the live data, delaying analysis and decision-making on the results.

Modeling forms the core of live data processing for Digital Twins. The sensor fusion techniques discussed in section 4 need to be chosen in order to get data that fits the requirements of the DT model. Likewise, the accuracy and detail of the model define how useful it is in creating strategies to solve problems, the penultimate goal of DT modeling.

6 Problem Solving through Simulation Modelling

A digital twin, consisting of a computer-based model informed by real data, provides a form of "digital knowledge" about the system. This knowledge does not provide value or improvement unless interpreted in a way that solves some problem. In the Data-Information-Knowledge-Wisdom hierarchy, interpreting knowledge is the final step to developing "wisdom", or strategies to solve problems. The problems digital twins solve were outlined in section 2.1, "Purpose of Digital Twins". This section will discuss how the "digital knowledge" can be applied to solve the problems identified, including Diagnostics and Fault Classification, and Schedule, Control, and Asset Optimisation.

6.1 Diagnostics & Fault Classification

Traditionally, Machine Learning an effective method to transform raw data from an industrial process into meaningful results, such as for fault diagnosis. Many different algorithms have been used for this. Pang et al. [42] combined data from a wind turbine SCADA system and used various convolutional

[17]https://www.mathworks.com/products/simulink.html

neural network techniques to predict faults. They used a convolutional neural network to iterate over recent readings of all sensors simultaneously. They also performed this analysis using shallow machine learning methods and found that the deep learning methods performed relatively better.

These same methods may be applied to the outputs of digital twin models. Tao et al. performed a case study on a digital twin system for a wind turbine in a power plant [8]. They designed the Digital Twin to model the gearbox, factoring in the torque, vibration frequency, and variance from the physical model, with the maximum contact stress, maximum bending stress, and gear meshing number calculated by the Digital Twin. An Extreme Learning Machine (ELM), a single hidden layer neural network, was trained to predict fault causes on the output of the digital twin. This performed up to 30% better than an equivalent ELM trained on the data from the physical model, without the extra parameters provided by the Digital Twin.

6.2 Optimisation

Because digital twins replicate not just the system inputs and outputs, but also the internal structure, they have increased flexibility to accurately simulate and optimise systems. Schulse et al. introduced the concept of "Experimental Digital Twins", which is the practice of using a digital twin model (complete with information about the digital twin's current environment) to simulate different methods of performing the same task [43]. This involved creating a virtual "testbed" for the digital twin, in order to see how it performed under different conditions. Optimisation functions can be defined and multiple experiments can be run in identical virtual conditions to see which would perform the best. In one test, a 3-D digital twin of a robot was used, and simulations of various procedures were used to find the optimal position to install a camera to maximise the visibility of the work the robot was performing. They concluded that this same technology could be used in the development of future systems, and to improve the operation of existing systems. While the techniques discussed in this paper were applied to 3-D robotic simulation, the approach of creating a virtual test bed and simulating various optimisation functions can still be applied to other types of digital twins.

In a Process Engineering context, there are three main ways to optimise a system: Asset Optimisation, Schedule Optimisation, and Control Optimisation. The following subsections will discuss each of these in turn, and analyse how Digital Twins and Live Data can be used to assist them.

6.2.1 Asset Optimisation

Asset optimisation is the process of retrofitting or upgrading assets in an industrial process to reduce operation costs. A common way to perform asset optimisation is pinch analysis [44] which analyses streams that need to be heated or cooled to find ways to transfer heat from one to the other. This decreases the amount of external work required to provide heating and cooling. Other methods include analysing waste costs, reaction quality, mass balances, or safety indicators to compare theoretical optimums and current performance [45]. In these cases, data collection is important, and digital twins can enable more accurate data collection and estimation of system properties. However, Asset Optimisation is by nature an "offline" process, so real-time analysis is not critical.

6.2.2 Schedule Optimisation

Scheduling involves finding the most efficient way to run a process with minimum overhead between tasks. Georgiadis et al. [46] reviewed the theory and current real-life industrial applications of process scheduling. They concluded that while there was substantial theory behind schedule optimisation, more integration was needed between the mathematical models and industrial Enterprise Resource Planning systems. However, digital twin technology and live data were not specifically addressed as a way to improve this, which could be an area for future investigation.

Others have investigated how schedule optimisation can become self-adaptive using live data. Qiao et al. [47] proposed a framework based on Cyber-Physical systems, where the Cyber-Physical system is enabled to analyse live data (via Digital Twin technology or another method) to identify disturbances. The system could then automatically recompute its schedules to find the optimum method of continuing operations based on the disturbance. The Cyber-Physical system used the same pattern of information processing already identified in this literature review - converting *data* into meaningful *information*, updating scheduling *knowledge*, and creating a coherent *strategy* to continue operations.

6.2.3 Control Optimisation

In this context, "Control Optimisation" refers to adjusting factory controls (such as PID controller targets) to their optimal values. This is related to the field of Model Predictive Control (MPC). Maxim et al. [48] provided an overview of Control in Industry 4.0, and identified that tools to monitor and optimise the performance of PID controllers via parameter tuning are generally available, but that there is more room to iterate in the field of Model Predictive Control and plant-wide economic optimisation. They also concluded that Digital Twins combined with data-driven control design may improve performance. Flowsheet simulation tools such as IDAES [18] are able to perform model predictive control, as demonstrated by Parker et al. [49].

Alves de Araujo Junior et al. [23] created a digital twin using a machine learning model based on automatically extracting fuzzy rules. They demonstrated it could optimise the number of fans needed for a water cooling system based on the factory's power data, the radiator inlet temperature, and ambient temperature data. This was able to have a low error response even with a relatively small amount of training data, saving an estimated 1.44MWh of energy per day.

6.3 Discussion

Problem-solving algorithms are likewise highly context-dependent. The best way to support them is to ensure that real-time data streams from DT modeling is available through standard databases or data streaming protocols. This will enable piping data into other existing control platforms, and developing custom algorithms, or custom machine learning models.

The software development project will focus mainly on control optimisation. This forms a good first use case for live data processing, because it is well established, and factory control systems can be easily compared in a timely manner, using either simulated or real data. If enough time is available, fault classification and schedule optimisation algorithms can be explored, but asset optimisation does not benefit from live data processing in the same way. By making the data from DT modelling available through standard data-streaming protocols, it will not be a technical challenge to apply the project to these different areas.

7 Summary

This literature review covers the data processing pipeline through the lens of a Data-Information-Knowledge-Wisdom hierarchy. By doing so, a broad-spectrum survey of research and technologies that enable insights from live industrial process data is presented. These insights have been discussed in the context of the software development project to develop a system that ingests real-time sensor data from chemical processing plants and use a mathematical model to simulate the plant in real time. Knowledge gaps and additional requirements have been identified, and a high-level development plan has been outlined, fulfilling the purpose of the literature review as stated in section 2.1. The research questions in section 1.4 have been answered as follows:

[18] `https://idaes-pse.readthedocs.io/en/stable/`

RQ1: Where is real-time data collection most used in industrial processing facilities, and where is there potential for it to be used better?

Real-time data is used extensively for monitoring and improving operational efficiency, particularly for schedule optimisation. It shows promise for more advanced diagnostics, fault classification, and control optimisation. Real-time data is a crucial element of the emerging field of Digital Twins because it enables the virtual model to be aware of changing conditions in the real world.

This provides key requirements for the software development project. The project will need to provide reliable software because it needs to be able to work continuously to enable crucial tasks such as diagnostics and optimisation to take place. The software must be scalable enough to work in large and small industrial settings and needs to provide interoperability with existing systems that perform some of these roles already.

RQ2: What techniques and processes are required to turn real-time data into useful insights that can inform and automate decisions in an industrial facility?

This question has been answered by outlining the data processing pipeline. Raw data is collected from SCADA systems, IoT networks, and system logs. This data is organised and labelled, tagging it with relevant context such as timestamps and source information. Sensor data from multiple sources can then be fused together to provide more meaningful, understandable interpretations. This increases the accuracy and enhances fault tolerance. These interpretations serve as a bridge between raw data and model simulation. Simulation is used to estimate other parameters of the system that may not be able to be measured directly by sensors, enabling a comprehensive view of the entire industrial process's state. Finally, manually defined or data-driven algorithms can be used to respond to the process's state, to optimise schedule, control, or maintenance tasks.

This provides context for the software development project. As the project's aim is to ingest real-time data from an industrial process and use a mathematical model to estimate other parameters in real-time, this sits the project squarely in the sensor fusion and model simulation areas of the data processing pipeline. Hence, the inputs and outputs of the project can be clearly defined. It will need to be able to interface with lower-level data collection software, as they will provide the input data of the project. It will need to be able to output simulation results in real-time in a manner that enables higher-level algorithms for process optimisation to be developed on top of them.

RQ3: How can real-time data be integrated into a Digital Twin that is based on mathematical, first-principles simulation techniques?

Some key methods to overcome these technical challenges have been identified. Advanced methods for sensor fusion, such as Bayesian Analysis, Fuzzy Logic, and Theory of Evidence models, help convert data into a more interpretable format that is appropriate for use in an analytical or mathematical model. Furthermore, Hybrid modelling has been shown to enable more flexible modelling that still retains the advantages of mathematical modelling. It is better suited to the variation and noise that is inherent in real-world sensor measurements. Hybrid Modelling also enables the modelling of complex dynamics that are difficult to capture via mathematical modelling alone. Additionally, live modelling augments models with the ability to update themselves in real time, so that the model can reflect changes in the real-world environment.

This provides a basis for developing the core features of the software development project. The project will need to incorporate existing or novel sensor fusion technologies to prepare data for modelling. A mathematical modelling framework will need to be chosen that supports hybrid modelling, and technologies will need to be developed to enable live modelling in that environment. Further research and review will be required in this area to find the optimal technologies and methods to use to develop these features. The optimal solution will depend highly on the external constraints and requirements of the stakeholders.

These insights enhance the project proposal in appendix **??** by specifying *why* and *how* the project can be built. By updating the specifications and requirements of the project based on the discussions in this literature review, the software development project is most likely to achieve its purpose to minimise industrial process energy use via efficient scheduling, optimisation, and maintenance procedures.

References

[1] World Economic Forum. Global Risks Report. January 2023. Accessible at `https://www.weforum.org/publications/global-risks-report-2023`.

[2] Ministry of Business, Innovation & Employment. Low Emissions Economy: Decarbonising Process Heat. August 2022. Accessible at `https://www.mbie.govt.nz/building-and-energy/energy-and-natural-resources/low-emissions-economy/decarbonising-process-heat/`.

[3] Timothy Gordon Walmsley, Panos Patros, Wei Yu, Brent R. Young, Stephen Burroughs, Mark Apperley, James K. Carson, Isuru A. Udugama, Hattachai Aeowjaroenlap, Martin J. Atkins, and Michael R. W. Walmsley. Adaptive digital twins for energy-intensive industries and their local communities. *Digital Chemical Engineering*, 10:100139, March 2024.

[4] M. Mazhar Rathore, Syed Attique Shah, Dhirendra Shukla, Elmahdi Bentafat, and Spiridon Bakiras. The role of ai, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities. *IEEE Access*, 9:32030–32052, 2021.

[5] Wei Yu, Panos Patros, Brent Young, Elsa Klinac, and Timothy Gordon Walmsley. Energy digital twin technology for industrial energy management: Classification, challenges and future. *Renewable and Sustainable Energy Reviews*, 161:112407, June 2022.

[6] Yiwen Wu, Ke Zhang, and Yan Zhang. Digital Twin Networks: A Survey. *IEEE Internet of Things Journal*, 8(18):13789–13804, September 2021. Conference Name: IEEE Internet of Things Journal.

[7] Azad M. Madni, Carla C. Madni, and Scott D. Lucero. Leveraging digital twin technology in model-based systems engineering. *Systems*, 7(1), 2019.

[8] Fei Tao, Meng Zhang, Yushan Liu, and A. Y. C. Nee. Digital twin driven prognostics and health management for complex equipment. *CIRP Annals*, 67(1):169–172, January 2018.

[9] Chenhan Zhang and Zhenlei Wang. A data-driven strategy for industrial cracking furnace system scheduling under uncertainty. *Chemical Engineering Science*, 277:118865, August 2023.

[10] Yong Zhang, Ying Yang, Steven X. Ding, and Linlin Li. Data-Driven Design and Optimization of Feedback Control Systems for Industrial Applications. *IEEE Transactions on Industrial Electronics*, 61(11):6409–6417, November 2014. Conference Name: IEEE Transactions on Industrial Electronics.

[11] Krupa Prag, Matthew Woolway, and Turgay Celik. Toward Data-Driven Optimal Control: A Systematic Review of the Landscape. *IEEE Access*, 10:32190–32212, 2022. Conference Name: IEEE Access.

[12] Mohammed Alkatheri. *Data-driven Optimization: Applications to Energy Infrastructure and Process Industry*. Doctoral Thesis, University of Waterloo, December 2021. Accepted: 2021-12-20T15:59:47Z.

[13] Murray E. Jennex. Big data, the internet of things, and the revised knowledge pyramid. *SIGMIS Database*, 48(4):69–79, nov 2017.

[14] Jennifer Rowley. The wisdom hierarchy: representations of the dikw hierarchy. *Journal of Information Science*, 33(2):163–180, 2007.

[15] Louis G. Birta and Gilbert Arbez. *Modelling and Simulation: Exploring Dynamic System Behaviour*. Simulation Foundations, Methods and Applications. Springer International Publishing, Cham, 2019.

[16] Isuru A. Udugama, Carina L. Gargalo, Yoshiyuki Yamashita, Michael A. Taube, Ahmet Palazoglu, Brent R. Young, Krist V. Gernaey, Murat Kulahci, and Christoph Bayer. The Role of Big Data in Industrial (Bio)chemical Process Operations. *Industrial & Engineering Chemistry Research*, 59(34):15283–15297, August 2020. Publisher: American Chemical Society.

[17] Omar Chamorro-Atalaya, Dora Arce-Santillan, Guillermo Morales-Romero, Nicéforo Trinidad-Loli, Adrián Quispe-Andía, and César León-Velarde. SCADA and Distributed Control System of a Chemical Products Dispatch Process. *International Journal of Advanced Computer Science and Applications*, 12(12), 2021.

[18] Sirish L Shah. Sensor-fusion strategies for process and performance monitoring. 2008.

[19] Baptiste Delporte, Laurent Perroton, Thierry Grandpierre, and Jacques Trichet. Accelerometer and Magnetometer Based Gyroscope Emulation on Smart Sensor for a Virtual Reality Application. *Sensors & Transducers.*, 14-1(Special Issue ISSN 1726-5479):p32–p47, March 2012.

[20] Furqan Alam, Rashid Mehmood, Iyad Katib, Nasser N. Albogami, and Aiiad Albeshri. Data Fusion and IoT for Smart Ubiquitous Environments: A Survey. *IEEE Access*, 5:9533–9554, 2017. Conference Name: IEEE Access.

[21] S. Ashwin Renganathan, Kohei Harada, and Dimitri N. Mavris. Aerodynamic Data Fusion Toward the Digital Twin Paradigm. *AIAA Journal*, 58(9):3902–3918, 2020. Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.J059203.

[22] Matthias Lermer and Christoph Reich. Creation of Digital Twins by Combining Fuzzy Rules with Artificial Neural Networks. In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 5849–5854, October 2019. ISSN: 2577-1647.

[23] Carlos Antonio Alves de Araujo Junior, Juan Moises Mauricio Villanueva, Rodrigo José Silva de Almeida, and Isaac Emmanuel Azevedo de Medeiros. Digital Twins of the Water Cooling System in a Power Plant Based on Fuzzy Logic. *Sensors*, 21(20):6737, January 2021. Number: 20 Publisher: Multidisciplinary Digital Publishing Institute.

[24] Arslan Munir, Erik Blasch, Jisu Kwon, Joonho Kong, and Alexander Aved. Artificial Intelligence and Data Fusion at the Edge. *IEEE Aerospace and Electronic Systems Magazine*, 36(7):62–78, July 2021. Conference Name: IEEE Aerospace and Electronic Systems Magazine.

[25] Vahideh Saeidi, Biswajeet Pradhan, Mohammed O. Idrees, and Zulkiflee Abd Latif. Fusion of Airborne LiDAR With Multispectral SPOT 5 Image for Enhancement of Feature Extraction Using Dempster–Shafer Theory. *IEEE Transactions on Geoscience and Remote Sensing*, 52(10):6017–6025, October 2014. Conference Name: IEEE Transactions on Geoscience and Remote Sensing.

[26] William de Paula Ferreira, Fabiano Armellini, and Luis Antonio De Santa-Eulalia. Simulation in industry 4.0: A state-of-the-art review. *Computers & Industrial Engineering*, 149:106868, November 2020.

[27] Sameera Abar, Georgios K. Theodoropoulos, Pierre Lemarinier, and Gregory M. P. O'Hare. Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33, May 2017.

[28] Wilson Trigueiro de Sousa Junior, José Arnaldo Barra Montevechi, Rafael de Carvalho Miranda, and Afonso Teberga Campos. Discrete simulation-based optimization methods for industrial engineering problems: A systematic literature review. *Computers & Industrial Engineering*, 128:526–540, February 2019.

[29] Anupam Keshari, Anand N. Sonsale, Brij K. Sharma, and Sanjay D. Pohekar. Discrete Event Simulation Approach for Energy Efficient Resource Management in Paper & Pulp Industry. *Procedia CIRP*, 78:2–7, January 2018.

[30] E N Pistikopoulos, Ana Barbosa-Povoa, Jay H Lee, Ruth Misener, Alexander Mitsos, G V Reklaitis, V Venkatasubramanian, Fengqi You, and Rafiqul Gani. Process systems engineering – *The generation next? Computers & Chemical Engineering*, 147:107252, April 2021.

[31] Blaž Rodič. Industry 4.0 and the New Simulation Modelling Paradigm. *Organizacija*, 50(3):193–207, August 2017.

[32] Gilles Privat and Alexey Medvedev. Guidelines for modelling with ngsi-ld. *ETSI White Paper*, 42, 2021.

[33] Víctor González, Laura Martín, Juan Ramón Santana, Pablo Sotres, Jorge Lanza, and Luis Sánchez. Reshaping Smart Cities through NGSI-LD Enrichment. *Sensors*, 24(6):1858, January 2024. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[34] *Modelica ® -a unified object-oriented language for systems modeling: language specification*. 2023.

[35] The Anylogic Company. Multimethod Simulation Modelling for Business Applications: Overview with guided model building example. *Anylogic.com*, 2019.

[36] S. De, M. Kaiadi, M. Fast, and M. Assadi. Development of an artificial neural network model for the steam process of a coal biomass cofired combined heat and power (CHP) plant in Sweden. *Energy*, 32(11):2099–2109, November 2007.

[37] Joel Sansana, Mark N. Joswiak, Ivan Castillo, Zhenyu Wang, Ricardo Rendall, Leo H. Chiang, and Marco S. Reis. Recent trends on hybrid modeling for industry 4.0. *Computers Chemical Engineering*, 151:107365, August 2021.

[38] Francesco Ceccon, Jordan Jalving, Joshua Haddad, Alexander Thebelt, Calvin Tsay, Carl D. Laird, and Ruth Misener. OMLT: Optimization & Machine Learning Toolkit. *Journal of Machine Learning Research*, 23(349):1–8, 2022.

[39] Timur Bikmukhametov and Johannes Jäschke. Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models. *Computers Chemical Engineering*, 138:106834, July 2020.

[40] Giovanni Lugaresi and Andrea Matta. REAL-TIME SIMULATION IN MANUFACTURING SYSTEMS: CHALLENGES AND RESEARCH DIRECTIONS. In *2018 Winter Simulation Conference (WSC)*, pages 3319–3330, December 2018. ISSN: 1558-4305.

[41] The Anylogic Company. An Introduction to Digital Twin Development. *Anylogic.com*, 2019.

[42] Yanhua Pang, Qun He, Guoqian Jiang, and Ping Xie. Spatio-temporal fusion neural network for multi-class fault diagnosis of wind turbines based on SCADA data. *Renewable Energy*, 161:510–524, December 2020.

[43] Michael Schluse, Marc Priggemeyer, Linus Atorf, and Juergen Rossmann. Experimentable Digital Twins—Streamlining Simulation-Based Systems Engineering for Industry 4.0. *IEEE Transactions on Industrial Informatics*, 14(4):1722–1731, April 2018.

[44] Ian C. Kemp and Jeng Shiun Lim. *Pinch Analysis for Energy and Carbon Footprint Reduction: User Guide to Process Integration for the Efficient Use of Energy*. Butterworth-Heinemann, August 2020. Google-Books-ID: yW_KDwAAQBAJ.

[45] Ana Carvalho, Rafiqul Gani, and Henrique Matos. Design of sustainable chemical processes: Systematic retrofit analysis generation and evaluation of alternatives. *Process Safety and Environmental Protection*, 86(5):328–346, September 2008.

[46] Georgios P. Georgiadis, Apostolos P. Elekidis, and Michael C. Georgiadis. Optimization-Based Scheduling for the Process Industries: From Theory to Real-Life Industrial Applications. *Processes*, 7(7):438, July 2019. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

[47] Fei Qiao, Juan Liu, and Yumin Ma. Industrial big-data-driven and CPS-based adaptive production scheduling for smart manufacturing. *International Journal of Production Research*, 59(23):7139–7159, December 2021. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207543.2020.1836417.

[48] Anca Maxim, Dana Copot, Cosmin Copot, and Clara M. Ionescu. The 5W's for Control as Part of Industry 4.0: Why, What, Where, Who, and When—A PID and MPC Control Perspective. *Inventions*, 4(1):10, March 2019. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[49] Robert B. Parker, Bethany L. Nicholson, John D. Siirola, and Lorenz T. Biegler. Model predictive control simulations with block-hierarchical differential–algebraic process models. *Journal of Process Control*, 132:103113, December 2023.