A Generic, Four-Layered, Functional Architecture Based Framework for Rapid Development of Multi-Domain Cyber-Physical Systems

Cham Da Ming
Bachelor of Engineering in the field of
Mechatronics Engineering

A thesis submitted for the degree of *Doctor of Philosophy* at
Monash University in 2020

Department of Mechatronics Engineering
School of Engineering

Copyright notice

Abstract

Cyber-Physical System (CPS) is the seamless integration of physical systems with a computational core across a network. One of CPS's main goals is to improve productivity by collecting, analysing, and controlling data from the connected systems. CPS design principles can be applied to various domains such as manufacturing, medical, and transportation. Most CPS frameworks designed currently are application and domain-specific and hence are not able to integrate systems from different fields, which limits the functionalities and applications. A functional framework for generic usage is thus highly desired. However, designing the CPS framework to be functional in all domains and applications is a challenging task. The generic framework must encompass different requirements from various fields of research. For example, the smart grid requires the transmission of real-time sensor data across thousands of kilometers with a significantly high-security level. Another often ignored challenge is implementing a CPS in time-critical systems. The physical system must sense and react immediately when an event occurs, which requires effective latency management. To accommodate various requirements from different fields of applications and overcome the challenges mentioned above, this Ph.D. study has developed a Generic Cyber-Physical System Framework (GCPSF). The GCPSF is validated for two domains of research - Manufacturing and Intelligent Transportation Systems. The proposed framework integrates the advantages of the previous CPS architectures and provides a step-by-step methodology in building a CPS in any domain. The methodology proposed is simple yet enables designers and integrators from different backgrounds to quickly develop a fully functional CPS. The proposed CPS framework addressed problems related to latency in the physical and communication systems. The seamless connections paved the way to utilize Digital Twin (DT) for CPS modelling and control.

Declaration

This declaration is to be included in a standard thesis. Students should reproduce this section in their thesis verbatim.

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature: ……………………

Cham Da Ming

Print Name: ………………….

1 / May / 2020

Date: ………………………….

Publication during enrolment

Published journal article

D.M. Cham, M. Shanmugavel, V.R. Sampathkumar, “Simulation of Vehicle Steering Angle and Lateral Acceleration in Mitigating Potential Run-Off-Road Crashes,” in Journal of the Society of Automotive Engineers Malaysia, 2017, Vol 1, No 3, pp. 251-261.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my main supervisor Dr. Veera Ragavan Sampath Kumar, for his endless support in my journey of completing my Ph.D. Without his motivation, patience, and immense knowledge, the goals of the research would not be realized. I would also like to thank my associate supervisor Dr. Madhavan Shanmugavel for his continuous guidance on the experimentation and writing of the thesis. I wish to show my gratitude to Professor Antonios Tsourdos for being my external supervisor and guiding my in the experimentations. I feel privileged to have all my supervisors constantly encouraging me throughout the adventure, and I am indebted to them for the rest of my life.

Besides my supervisors, I would like to thank my Ph.D research progress panel, Assoc. Prof. Tan Chee Pin (Edwin), Dr. Alpha Agape Gopalai, Dr. Darwin Gouwanda, and A/Prof Hung Yew Mun for their insightful comments and advice in the progress reviews. I would like to pay my special regards to Dr Mohammed Ayoub Juman for his advice and support in the progress review.

I am grateful to acknowledge the Ministry of Higher Education Malaysia (FRGS/2/2014/SG04/MUSM/03/1) and Monash University for providing financial aid to carry out this Ph.D. I also acknowledge Scribbr editor Michelle,a language specialist with technical editing experience for providing professional proofreading services. The intervention is restricted to language, consistency, and academic style. In required areas, advice is provided on the structure or clarity using exemplars only.

I wish to express my gratitude to Bathmanathan Arjuna, Paneer Arjun, and Tharmaa Varatharaja, who were always there for me in lab matters and technical support. Special thanks to my fellow Ph.D candidates Shavi, Jesh, Loong Yi, Shalini, Howgen, Junn, and Kosa for always giving moral support.

I want to thank my parents, siblings, and best friends Vincent and Zhi Wei, for their unconditional love and support. These few years leading up to the completion of the Ph.D was challenging. They understood my goals and aspiration throughout the entire journey.

Table of Contents

Abbreviations

3D	3 dimensions
A2A	Asset to automation
AI	Artificial intelligence
API	Application programming interface
CAN	Controller area network
CoAP	Constrained application protocol
COTS	Commercially available off-the-shelf
CPS	Cyber-Physical System
DDS	Data distribution service
D-H	Denavit-Hartenberg
DT	Digital twin
EtherCAT	Ethernet for Control Automation Technology
EtherNet/IP	Ethernet using TCP/IP
FARE	Failure analysis and reliability estimation
FPGA	Field-programmable gate array
GCPSF	Generic cyber-physical system framework
GUI	Graphical user interface
HTTP	Hypertext transfer protocol
I2C	Inter-integrated circuit
IIRA	Industrial Internet Reference Architecture
IoT	Internet of things
IT	Information technology
ITS	Intelligent transportation system
LWM2M	Lightweight machine to machine
M2M	Machine to machine
M2P	Machine to people
Modbus/IP	Modbus that uses TCP/IP networks
MQTT	Message queuing telemetry transport
NASS	Network-aware supervisory system
OEE	Overall equipment effectiveness
OPC UA	Object linking and embedding for process control unified architecture

OT	Operational technology
P2P	People to people
PCB	Printed circuit board
PHM	Product health management
PIC	Proportional-integral-derivative
PLC	Programmable logic controller
PLM	Product life management
QoS	Quality of service
RAMI 4.0	Reference architectural model industry 4.0
RPM	Rotation per minute
RTOS	Real-time operating system
SAP	Systems, application and products
SD	Secure digital
SMA	Shape memory alloy
SOA	Service-oriented interface
SPI	Serial peripheral interface
TCP/IP	Transmission control protocol and internet protocol
TIA	Totally Integrated Automation
UART	Universal asynchronous receiver/transmitter
V2I	Vehicle to infrastructure
V2V	Vehicle to vehicle
Wi-Fi	Wireless fidelity

Glossary

Abstraction	The action of removing detailed programming languages by using ideas and concepts instead.
Accelerometer	A sensor that measures the acceleration of the object for vibration or speed detection.
Ad hoc trial	Trials that are done only when needed.
Aggregation	The combination of several objects or systems.
Ambiguity	Uncertainties of the system.
Anomalous data	Data that is outside the normal range.
Application developer	An expert in creating software services.
Architecture	A multi-discipline approach in designing a plan for development of a system.
Authentication	The action of confirming the user has clearance to access the system.
Bayesian network	A methodology that uses probability to measure the relationships between variables.
Behavior model	A model that predicts the system's behavior when introduced to inputs.
Big data	Large amount of data that is produced by different sources for analysis and predictions.
Cloud computing	The action of processing data that includes data storage and analytics in a virtual computer.
Cloud platform	A virtual place that handles all the cloud services such as computing and storage.
Cloud server	A virtual computer that can be accessed using a network for cloud computing.
Communication bandwidth	The frequency range for a communication device to transmit data.
Communication overhead	The time taken on the transmission of data using a network.
Communication protocol	A standard set of methodology for communication over a network.

Communication system	A set of devices that bridges the communication between systems.
Computational core	The main data processing device for the entire system.
Concurrent model	A model that runs in parallel with other applications within the software.
Context aware	A system that can sense its environment and has the intelligence to act accordingly.
Context-adaptive synchronization method	A method that can ensure data is the same on two or more systems with the consideration of its environment.
Continuous-time system	A system that functions parallel to real-time.
Cross-layer interaction	The open communications between systems that are within a hierarchy of layers.
Crowdsensing	A large number of participating devices exchanging sensor information for various applications.
Cyber system	A virtual system that exists in computing systems for data storage and analytics.
Cyber-physical system	A combination of an actual system and a virtual system using seamless connections.
Data-driven analytical algorithms	A set of rules stating the methods of processing information or decisions based on data.
Decentralized control	A control method of using multiple controllers for different systems without taking other systems into account.
Design stage	The initial stage of a product lifecycle management process, that focuses on the details of the product such as structure, materials, and functions.
Deterministically	The idea of having the controllers react according to its environment without fail.
Digital shadow	A virtual representation of an existing object with one direction data flow from the physical to the digital model.
Digital thread	A concept that links digital twins with their corresponding physical twins.
Digital Twin	A virtual representation of a physical object or system.
Discrete-time system	A system that operates in time steps.

Distributed control	A control method that obtains information from various systems and sends control commands to each of the systems.
Domain	A field of research.
Drift	A continuous motion of the vehicle that opposes the intended direction.
Dubins path	A shortest path connecting two positions and their orientations using circles and straight lines.
Edge computing	The data processing of the system that is done in the individual controllers instead of a centralize processor.
Empirical observation	The results obtained from sensing, experiments, and analysis of the data.
Emulation	A copy of the original function using different means.
Fault-propagation	A spreading of a single error that could result in catastrophic failure.
Feedback control	A control method of using the sensor values as feedback to adjust the control commands.
Fidelity	A measure of how close the simulated model is to its corresponding object or system.
Fourth industrial revolution / Industry 4.0	The introduction of networking components in the automation industry to enhance existing and new systems.
Gateway	A device that connects different types of communication protocol devices to a computational core.
Generic framework	A framework that encompasses the requirements of a variety of domains.
Grayscale	An image made up of different shades of gray.
Grid	The electrical power grid that distributes energy to consumers.
Gyroscope	A sensor that measures the rate of rotation of the object.
Hall effect sensor	A sensor that detects magnets.
Hard iron distortion	The sensor noise introduced by materials with constant magnetism such as magnets.
Heterogeneous	A set of components that are of different content.

High-level programming	A programming language that simplifies the process of programming a controller using common language instead of bits and bytes programming.
Holistic intelligence	The knowledge of the system that includes every aspect of its structure, behavior, and environment.
Homogeneous	A set of components that are of the same content.
Internal clock	The measure of time using the microcontroller’s timers.
Interrupt service routine	A programming method of interrupting the main program of the system to operate a service.
Isogeometric analysis	A combination of designing tools (CAD and FEA) for product analysis.
Kalman filter	A filter that uses sensor measurements and calculations to predict a variable.
Kinematic model	A representation of a system using the motion of the objects without considering the forces.
Knowledge model	A model that is understandable by both computer and humans to instil knowledge.
Latency management	The action of controlling the transmission of information across networks in a timely manner.
Legacy system	An older machinery that are still being implemented on the factory floor.
Lifecycle	The stages of a product or system that starts from the birth to end of life.
Low-pass filter	A filter that removes high frequency noise.
Magnetometer	A sensor that acts as a compass in three dimensions.
Memory allocation	The action of retaining memory for specific variables in a program.
Metallurgical structure	A materials engineering term for the physical properties of metallic compounds such as alloys.
Microcontroller	A controller that is small in size for various applications.
Middleware	A software that connects physical systems to a cyber system.
Mobile agent	A program that can be operated in any computer within a networked system.

Modbus	A standard serial communication protocol for manufacturing systems developed by Modicon (Schneider Electric).
Motion capture system	A set of cameras that detect the motion of markers within the range of sight.
Moving median filter	A filter that uses the median of a set of sensor data to remove high frequency noise.
Multi-agent model	A model that uses different types of systems that includes discrete, continuous, and virtual systems.
Network topology	A layout of the communication devices and protocols in a system.
Networked system	Systems that are connected either using wired or wireless technology.
Normalize	A process of ensuring the data is organized in a certain manner.
Off-the-shelf component	A component that is ready made for implementation in specified systems.
Open-loop control	A control method that only uses the sensor input of the system.
Open-source	A software that is publicly available.
Packet switching	A method of communication that combines data into packets for transmission.
Peripheral	An external device that gives input or output signals to the controller.
Physical system	A system that is in the physical world.
Pneumatic linear actuator	An actuator or piston that moves in a linear motion using compressed air.
Powerlink	A common communication protocol for ethernet connections.
Predictive maintenance	The action of predicting the need for maintenance using sensor data and computation.
Probabilistic simulation	The simulation of variables using probability methods.
Product life management	The action of controlling the product throughout its entire lifecycle.
Profinet	A standard for industrial communications protocol.

Proximity sensor	A sensor that detects an object near the sensor.
Publish	A term used in MQTT protocol for transmitting data to a server.
Quantitative model	A method of representing systems using measurements and mathematical models.
Real-time system	A system that operates and can give a response at any given time.
Repository	A place of data storage.
Robust	A system that can function in changing environments.
Round-robin loop	A programming method of operating a list of programs in an orderly manner.
Safety-critical system	A system that prioritize safety.
Salient	The most important feature.
Schematic	A figure that describes the main features of an object or system.
Seamless connectivity	A connection between systems that is continuous.
Semantic	The study of meaning in programming languages.
Semi-flexible joint	A connection between two objects that allow some degree of motion.
Sequential workflow	A series of work objective that are set up in an orderly manner.
Service aware module	A software component that takes programmed services into account.
Service computing	A programming method of providing services using simple tools.
Soft iron distortion	The sensor noise introduced by materials with induced magnetism such as iron casings.
Steering angle	The angle between the front wheels of the vehicle and its forward direction.
Supervisory control	A control strategy that uses a computational core to oversee the controls of multiple systems.
Supplant	To replace.
Synchronization	The action of ensuring two or more systems operate in the same time.

Temporal logic	The logic of systems that are related to time.
Third industrial revolution	The introduction of automation using digitization of systems.
Time-critical system	The systems that prioritize timeliness.
Timesteps	The time interval whereby the system operates the next program.
Timing behavior	The ability of the system to adapt to environmental changes in the given time interval.
Tool suite	A set of tools for specified applications.
Trajectory planning and control	The action of deriving a path for a vehicle and sending control commands.
Turning radius	The radius of the circle when the vehicle is given a set steering angle and speed.
Uncertainty	The unpredictable conditions that may occur.
Validation	"An activity that ensures that an end product stakeholder's true needs and expectations are met." (IEEE-STD-610)
Verification	"A test of a system to prove that it meets all its specified requirements at a particular stage of its development." (IEEE-STD-610)
Vulnerability assessment	The action of determining the weaknesses of a system in a formal method.
Waypoint	A point on a planned path to guide a vehicle to its destination.
Wi-Fi module	A microcontroller that uses wireless communication protocols to connect offline devices to the internet.
Worst-case scenario time	The time taken to operate the program in the worst possible condition.
Yaw	The rotation along the vertical axis of the vehicle.

Table of Figures

Table of Tables

Chapter 1 Introduction

1.1 Background of Research Topic

The first industrial revolution began after the invention of steam-powered machinery. These machines powered factories and transportation. Then the first assembly line was developed in 1870 to increase the production rate of goods. The third industrial revolution is digitization and automation. Multitudes of machinery were built during that time to increase the production rate, efficiency, and effectiveness of manufacturing products [1]. These feats were made possible by system designers who built machinery that could manipulate sensors and actuators for production. Each machinery developed was limited to a singular control strategy, and any changes to the manufacturing process required assistance from the designers. However, a computational core networked to the production line can supervise and alter the entire manufacturing process from a single location. As of date, the introduction of networking systems to computational cores has facilitated the fourth industrial revolution. The evolution of manufacturing systems as explained is shown in Figure 1.

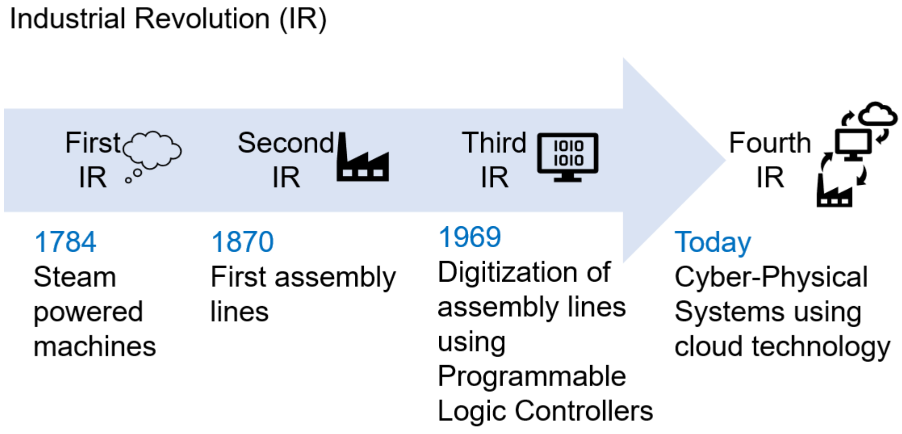


Figure 1 Timeline of Industrial Revolutions illustrating the evolution of manufacturing systems [1].

Cyber-physical system (CPS) is essentially the backbone of Industry 4.0. Dr. Helen Gill first introduced CPS in a presentation at the National Science Foundation [2]. The first definition of a CPS was the integration of computation with physical processes using feedback loops across networks that affect both physical and cyber systems. A CPS is a generic concept for seamless connectivity between systems for various applications. Thus, a CPS is neither an application nor a domain-specific theoretical concept, as shown in Table 1.

Table 1 List of domains that require a CPS for further innovation.

Domain	Need for the CPS	Value Added From the CPS
Electrical grid	To control the uncertainty of power generation from various renewable energy sources.	Real-time monitoring for grid-wide voltage control
Agriculture	To prevent forest fires in drought areas	Real-time temperature and humidity readings across the entire plantation
Healthcare	To improve outpatient care	Real-time health monitoring using microscopic sensors for patients outside of medical facilities
Transportation	To enable emergency vehicles to function more efficiently in congested areas	Real-time traffic monitoring and control over traffic lights for emergency vehicles
Home safety	To improve the safety of citizens in apartments	Real-time perimeter monitoring for suspicious activities using Artificial Intelligence (AI) and image recognition

There are limitless possibilities for implementing CPS in a variety of domains. However, at present, no research has built a functional framework for generic usage. In addition, CPS is categorized as a scientific study, and research on CPS has focused on architecture and theoretical concepts. Upon researching CPS, digital twin (DT) appeared to be a pragmatic solution for industry players and the development of a generic framework. A DT is an engineering study of models and data that virtually represents an actual product or system. The virtual representation acts as a computational core that uses data from the physical system to monitor, improve, and predict aspects of the physical system (Figure 2). Numerous obstacles must be dealt with before CPS and DT can become a pragmatic generic CPS framework (GCPSF) for Industry 4.0.

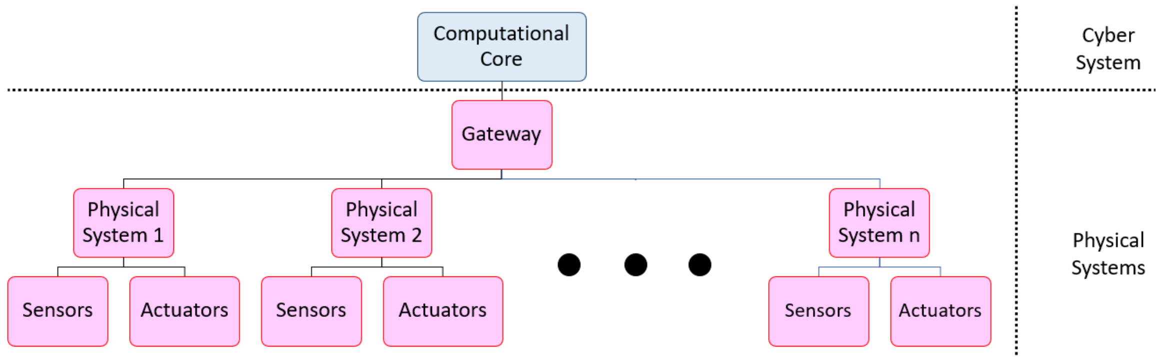


Figure 2 Schematic of a CPS to illustrate the division of physical and cyber systems.

1.2 Challenges of Building a Generic Cyber-Physical Framework

CPS is inherently a complex hybrid system, and its functionality spreads across multiple domains. Developing a CPS framework that meets the functional requirements of subsystems in all domains and applications is challenging. The generic framework must encompass different requirements from various fields of research. For example, developing a CPS for a smart grid application requires the data acquired from a multitude of geographically distributed sensors to be transmitted securely in near-real-time across thousands of kilometers of cyber-networks. In manufacturing applications, the developed CPS must be interoperable with legacy systems without standard communication protocols.

Besides, physical systems interact with the environment in continuous time. Any changes in the environment are observed by sensors in discrete time intervals triggering a continuous reaction from the controller. However, computer systems that control the CPS are typically discrete-time systems. These controllers operate in discrete time-steps and require high-frequency data captures to minimize the loss of information. Thus, the integration of continuous and discrete-time systems is essential for the reliability of the CPS.

Another significant challenge is implementing a CPS in time-critical systems. The physical system must sense and react immediately when an event occurs. An autonomous vehicle traveling at high speeds has only two choices when an obstacle is present: to avoid or to stop the vehicle. A timely decision from the navigation controller of an automated vehicle is crucial for success. Therefore, latency management is a critical requirement of time-dependent CPS. Latency management in a CPS, such as in modern-day cars, is a non-trivial problem. The autonomous control function in a CPS requires the seamless integration of multiple microcontrollers with limited computing power, sensors, and actuators. The communication of data across wired and wireless sensor networks increases the computing power demands on the microcontroller. Latency requirements and the communication overhead requires tightly coupling the physical and the cyber systems in real-time.

Once a CPS is networked, security becomes a challenge. CPS exists in both the physical and cyber worlds; attacks from either side endangers the security of the entire system. For example, tampering with a safety-critical sensor may result in catastrophic failure or the destruction of the entire system. Numerous researchers are currently working on general strategies to secure CPS and beyond the scope of this work. The GCPSF in this work only implements simple security modules; hence the security policies developed by researchers to prevent future vulnerabilities can be incorporated in this framework.

CPS spans across different domains and requires specialized knowledge to model each system. Hence, for developing a generic CPS framework, a hybrid modelling technique is needed. A digital twin (DT) is a suitable engineering solution for modelling different types of physical systems. DT modelling can integrate geometric, physical, behavior, rule, and process models, each optimally representing one part of the physical system. The integration of these hybrid models contributes to the many practical applications of the DT. However, modelling a DT is challenging because a seamless connection between systems is required.

1.3 Problem Statements

As widely reported in the research literature, there are multiple unresolved challenges for system engineers attempting to build a CPS. Consequently, a generic framework would ease the development of domain-specific CPS applications. The challenges of developing a CPS are considered in the problem statements for this work, which are described in the following sections:

- The need for Generic Cyber-Physical System Frameworks (GCPSFs)
- Latency Issues Affecting the CPS Performance
- Need for a Cross-Discipline Cyber System for the GCPSF
- Need to Verify and Validate the GCPSF

1.3.1 Need for Generic Cyber-Physical System Frameworks (GCPSFs)

The research timeline of the thesis, illustrated in Figure 3, describes the progression of the research direction. Based on the literature study from 2016 to date, CPS researchers [3, 4] have pointed out that a generic framework was missing in the literature. One of the main challenges of developing a generic framework is the seamless connections between systems. In 2017, DT that thrives on seamless connections between systems became a popular research topic. The development of a generic framework acts as the foundation for DT to enhance the cyber system of the CPS. Thus, this research focuses on the intersection of CPS and DT.

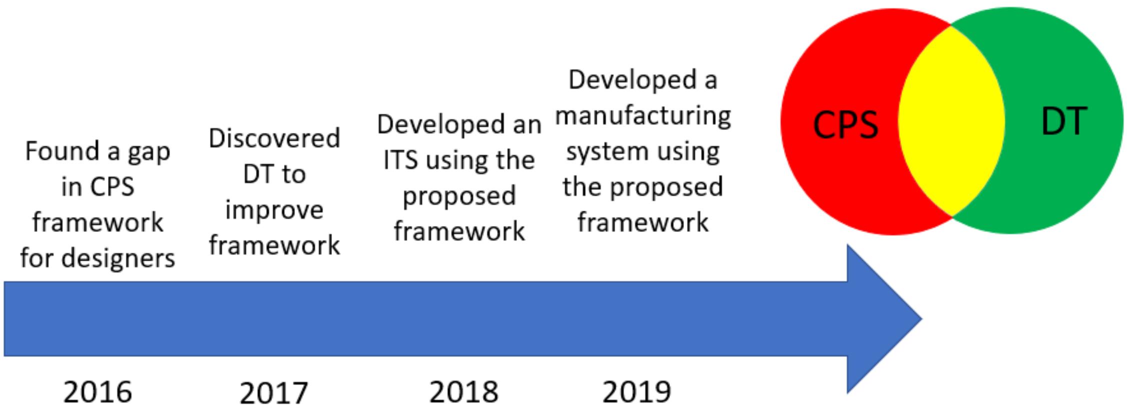


Figure 3 This research timeline is the time history of CPS and DT and the research topic's evolution with time.

The CPS and DT-integrated framework aim to ease the role of designers and system integrators in different domains. The verified framework acts as a foundation to reduce the cost and time of developing a CPS, resulting in the reduction of verifying every single component of the CPS. The implementation of a DT on top of the verified framework further reduces design and development costs. DT models of the physical system enhance the design stage by reducing the need for prototyping.

The GCPSF illustrates a simple step-by-step method of developing a CPS by adding generic and domain-specific modules in each layer to meet that domain's requirements. This provides an abstraction to hide complexity. By doing so, researchers can interactively program at higher levels based on the task's context and environment. The increase of the abstraction level increases the productivity of researchers because they can do more in less time. This will reduce the cost of researching a new topic and designing new experiments. Therefore, giving results that are difficult to obtain without the framework. GCPSF helps academics and researchers to iterate and explore different types of architectural and technical infrastructure. Thus, improving production, system integration, and interoperability of developing a CPS.

1.3.2 Latency Issues Affecting the CPS Performance

The GCPSF contains physical systems that interact with the environment. The uncertainties from the environment procreate the need for the physical system to perform specific tasks in real-time. However, real-time performance in networked systems such as CPS is especially challenging due to latency issues. The standard solution to design a system with real-time capabilities is to implement a real-time operating system (RTOS). However, researchers have stated that RTOS is an unreliable solution for complex systems [5-7]. The addition of networking systems further complicates the decision-making process in the RTOS. Therefore, the direct implementation of an RTOS in CPS would not provide a reliable physical system with real-time capabilities [5-7].

The physical systems require handling sensors, actuators, and networking control commands. The physical system needs to read, filter, process, compile and send sensor data across a network. After dealing with the sensors, control commands from a networked core prompt action on the actuators. The data transmission of a CPS requires reliability, timeliness, and security. These requirements solidify the need for reliable real-time performing physical systems.

The development of a near-real-time physical system with a limited computing power microcontroller showed the importance of latency management. Latency management starts from the physical system as communication systems introduce numerous challenges such as unstable connections and security concerns. Further research in communications could be beneficial for CPS and DT research.

1.3.3 Need for a Cross-Discipline Cyber System for the GCPSF

With the seamless connections set up by the physical system, the development of the cyber systems follows. In Figure 4, the cyber system in CPS acts as the computational core for central decision-making and control. The physical system of the CPS is operated using RTOS concepts while a DT forms the computational core. The gateway represents a device that connects the physical system to the computational core. Numerous methodologies exist for building cyber systems. However, a generic solution to develop the cyber system is needed for CPS.

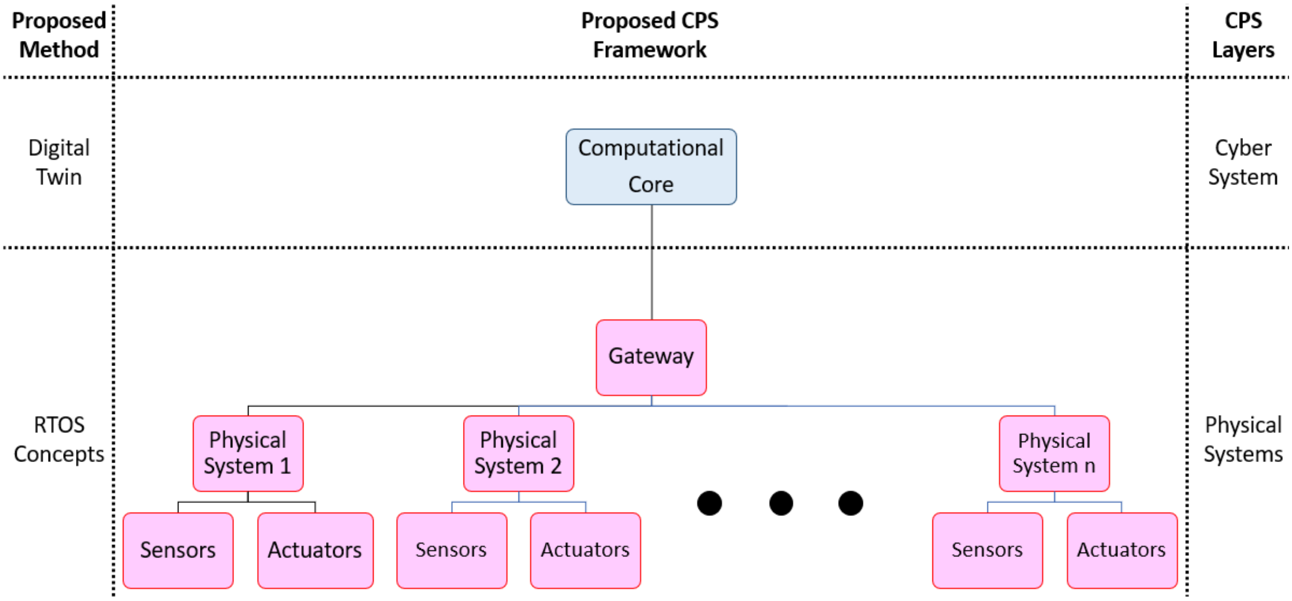


Figure 4 Schematic of the proposed generic CPS framework with the proposed methodology of implementation.

Because the cyber systems have been developed in numerous ways, as found in the literature, the proposed generic framework required a generic solution. The rise in popularity in DT research and its modelling and prediction capabilities were necessary for integrating the physical to the cyber system in the GCPSF. DT provides not only overall control and management capabilities but also integrates disparate models such as geometric, physical, behavior, rule, and process that describe the physical system in various viewpoints.

1.3.4 Need to Verify and Validate the GCPSF

Every system developed requires a verification process to ensure its reliability. The verification process for CPS is tedious and challenging. Various research has stated the need for a standard procedure for verifying and validating a CPS effectively [8-11]. System testing, verification, and validation of CPS are performed manually through experiments using integration test cases from multiple domain-specific standards. The experimental verification and validation of the proposed framework proved the framework is generic in two domains - ITS and manufacturing. With the validated framework, CPS designers and integrators have a practical solution for the rapid development of CPS.

1.4 Research Objectives

The aim of this work is to develop a fully functional Generic CPS Framework to fill the research gap. For verification and validation purposes, the developed generic framework was deployed to develop two domain-specific CPS applications – the ITS and manufacturing. In accordance with the problem statements, we have four objectives.

1. Develop a Pragmatic General Cyber-Physical System Framework.
2. Build Cyber and Physical Subsystems Capable of Near-Real-Time Behavior.
3. Derive a Digital Twin Model of the Cyber-Physical System.
4. Experimentally verify the models and the framework.

The two CPS applications in different domains were developed using the framework to illustrate the generic nature of the framework. The intelligent transportation system (ITS) CPS application demonstrates the functionality of a physical system with limited computing power but requiring near-real-time performing CPS. On the other hand, the manufacturing CPS application demonstrates the functional capability of a smart manufacturing system requiring modularity, interoperability, scalability (deployed using multi-vendor commercial off-the-shelf (COTS) components), and integration with legacy systems.

After developing functional CPS for two domains, the functionality was further augmented by developing DT models to scale the functionality through services made available in the cloud layer. First, the behavior model is the engineering behavior that provides an operational view of the manifested behavior. This model requires fixing a set of state variables and focusing on how they describe changes in the object's qualities and relationships. In this context, the object is the steering system of the vehicle and how the steering angle was affected by the steering inputs/commands. Steering systems in autonomous vehicles are the subsystems that essentially control the direction of travel. The kinematic model of the steering subsystem was parametrized and scaled to build the ideal DT model to predict the performance of the steering dynamics. The DT model was scaled because the size of the experiment was designed to be done indoors. Moreover, scaled experiments are cost effective as compared to using an actual vehicle. Therefore, the DT model was scaled to fit these criteria and to illustrate the enhanced value that DT technologies offer in the development of CPS.

1.4.1 Develop a Pragmatic GCPSF

Various architectures and guidelines were described in the literature, but a generic framework is still missing. Therefore, this work reviews the existing architectural concepts to determine the requirement of a generic CPS framework. The gaps in the existing framework were addressed to develop a pragmatic and generic framework for developing CPS.

1.4.2 Build Cyber and Physical Subsystems Capable of Near-Real-Time Behavior.

In an ITS application, the physical system is required to function in near-real-time (within an allowed latency). Due to the speed of the vehicles, one second could mean the difference between collision and obstacle avoidance. A scaled vehicle prototype with a microcontroller was used to demonstrate the real-time capabilities of the physical system. The limited computing resource of the microcontroller was used to exemplify the significant challenges posed in building a CPS with real-time constraints. The scaled vehicle was also used to demonstrate the ability to manoeuvre and send real-time data to a cloud server simultaneously.

For the manufacturing CPS, the physical system is required to function with different types of new and existing subsystems (legacy systems). In this domain, the subsystems of the manufacturing plant can be replaced by functionally equivalent subsystems and components. Integrating these heterogeneous entities was made possible by introducing a gateway architecture in the communication layer. This layer compiles data from various systems, sends the data to the cyber system, and enables scalability and interoperability, the basic requirements of manufacturing CPS. Therefore, the implementation of COTS components with GCPSF showed the framework to be modular, scalable, and interoperable.

1.4.3 Derive a Digital Twin Model of the Cyber-Physical System

Digital twin technology is the latest in the list of new technologies widely adopted by practitioners to mainstream digital transformation. The development of a DT starts from the design phase of product life management (PLM) and lives on until the end of the life of the product. The integration of models, such as geometric, process, and behavior models, can be added to provide real-time insight into the physical twin for optimisation, control, and product health management (PHM). The developed DT of the steering system on the scaled vehicle predicts the performance before the experiments began. For modelling of entities in the manufacturing domain, behavior DT model was built using a state machine to illustrate the importance of geometric and physical models. Without integrating the geometric and physical DT models, one cannot fully describe the entire manufacturing system.

1.4.4 Experimentally Verify the Models and Generic CPS Framework

Every component that will be integrated into the GCPSF must be individually verified from the sensors to digital twin models and services deployed on cloud servers. In this work, the verification process is performed for both the ITS and manufacturing domains to illustrate the generic nature of the framework and its ability to be adopted for domain-specific applications despite of its generality.

1.4.4.1 Cyber-Physical Intelligent Transportation System

The physical systems in ITS applications are time critical and domain-specific. Therefore, the developed physical system must be verified to function in near-real-time to meet the latency requirements. Developing a domain-specific CPS for ITS using the GCPSF becomes necessary. This ITS application highlights the importance of managing edge computing, and the DT models of the ITS subsystems, and the vast application potential.

1.4.4.2 Cyber-Physical Manufacturing System

Many factories still operate high-powered and expensive legacy systems, many of which fall into the safety-critical system category. Digitization of these legacy assets enhances the functionality, converting them into smart factories. Incorporating COTS IoT modules provides CPS developers with reliable, modular, and secure products to enhance features without much overhead. A famous example is the Overall Equipment Effectiveness (OEE) IoT modules for improving manufacturing productivity. However, the available COTS hardware uses different communication protocols to complicate the integration process. Therefore, a general framework is needed to simplify the integration process and assist designers in verifying manufacturing CPS.

1.5 Contributions

The significance of this work is the development of the GCPSF. The development of physical and communication systems within the framework enables the cyber system to function seamlessly. Moreover, the framework was implemented in different domains to show it is generic.

1.5.1 Proposed Four-Layer Functional Cyber-Physical System Framework

After reviewing the literature of existing CPS architectures, a four-layer CPS framework was developed and proposed. The framework is the fusion of different CPS and DT architectures. In order to prove the framework is pragmatic and generic, two CPS applications were developed for verification and validation. Moreover, the framework can be implemented in other fields of research as well.

1.5.2 Functional Near-Real-Time Operating Physical System

Most of the existing frameworks do not include real-time operation functionality. The interfaces to existing physical systems are abstract. Hence, many laboratory-scale vehicle prototypes were custom-built to provide the flexibility required for the data collection process and observation of internal subsystem behaviors for verification and validation of the framework. The most tedious activities of this research were the building of physical system prototypes. This meant incorporating generic near-real-time capabilities using real-time operating systems (RTOS) using custom programming microcontrollers hardware with limited computing power controlling critical processes. For example, each sensor and actuator connected through the microcontroller is checked for worst-case scenario time for reference. These methods can be implemented in small robots for the effective management of limited computing resources. Once the data is collected from the physical system, the next digitization activities can proceed. For example, developing virtual process models of the system, behavior model validations of the subsystem, and service orchestrations of the system functionality can be carried out.

1.5.3 Digital Twin of a Steering System

The developed DT model of a steering system demonstrates the implementation potential for full-sized vehicles. Detecting performance issues during the design stage and real-time data from the physical twin improve the behavior model's performance. Additional models can be supplemented on top of the behavior model for optimisation of the CPS.

1.5.4 Data Management of a Cyber-Physical System

With several sensors typically connected to the microcontroller, the implementation of sensor fusion increases the reliability of the data. These data are filtered and compiled into pre-allocated memories on the microcontroller to reduce processing time. Then, the microcontroller communicates the data to an SD card for backup and a Wi-Fi module for internet access to cloud servers. The details of the memory allocation and communication protocols are essential to reduce latency during communication between microcontrollers and cloud servers. Therefore, other networked systems can implement similar techniques for faster response times.

1.5.5 Adapting the Proposed GCPSF for ITS and Manufacturing

The CPS on the ITS and manufacturing domains were verified through experimentation by adapting the proposed CPS framework for specific purposes. The validation and verification process provides insight into how the proposed framework was implemented in different domains.

1.6 Thesis Outline

The research is separated into four research objectives. Each research objective corresponds to a research contribution. Table 2 contains the list of research objectives and contributions along with the corresponding chapters. The last two columns describe the actions taken to achieve the research objectives in different research domains.

Table 2 Outline of thesis using research objectives and the corresponding contributions and chapters where the approach is described.

Research Objective	Research Contribution	Chapter	CPS1 (ITS)	CPS2 (Manufacturing)
Development of a GCPSF	Proposed a four-layer generic CPS framework (GCPSF) suitable for CPS designers and academics.	3		
Development of a robust, reliable, and safe near-real-time physical and cyber subsystems.	Proposed using RTOS concepts for seamlessly integrating the physical systems with cyber system.	3, 5, and 6	Developed a near-real-time scaled vehicle with a limited computing power microcontroller	Implemented using off-the-shelf components with the latency management proposed in framework
Development of a DT	Proposed building a DT framework for integration of various models (discrete and continuous) and data.	3, 5, and 6	Derived a DT model of a steering system for performance prediction	Collected data from physical system; latency, however, is the significant issue
Verification and validation of a GCPSF	Customized a verification method based on model-based design and Brace framework. This customised method was used to verify GCPS requirements.	4 and 6	Experimented with the ITS CPS developed using the proposed framework	Experimented with the manufacturing CPS developed using the proposed framework

The following sections will describe the chapters of this thesis and the contributions to the research objectives.

Chapter 2 reviews the literature related to CPS and DT on the background, architectures, challenges, and applications. Since a CPS is a generic concept for different domains, a literature review for various functions was carried out to ensure each component was applicable. Additional attention was spent on vehicular steering systems and DT modelling methodology.

In this chapter 3, the development process of the generic CPS framework was described in detail and each architectural layer from sensors to the selection of cloud servers was discussed. Then, a step-by-step application of the GCPSF on both ITS and manufacturing domain was described.

The verification process of the CPS in both the ITS and manufacturing domains was described in chapter 4. The function of each component belonging to each layer in the framework was verified. The verification proves that the framework is a generic concept for developing a CPS from the ground up.

The results in chapter 5 displayed the utilisation of the microcontroller's central processing unit (CPU). The processing time revealed that even with limited computing power, the vehicle operated in near-real-time. The simulation of the DT model accompanied by open-loop experiments was carried out on the scaled vehicle, verifying the prediction capabilities. The time taken for data to be sent to the cloud for both the ITS and manufacturing domains were measured and compared. The difference between the two was significant, which indicated the importance of developing a physical system to function in real-time.

The discussion chapter 6 covered the key challenges and solutions to developing a near-real-time physical system. A methodology of modelling a steering system as a DT was discussed and how additionally integrating other models is beneficial. Lastly, the validation of the GCPSF was carried out using a self-built ITS and commercial manufacturing CPS.

The main contribution of this study is the methodology of building a CPS using the proposed framework as described in chapter 7. The physical and communication systems were built with a limited computing power microcontroller and Wi-Fi module. The cloud server with full customisation was installed on an off-site laptop for simulating a server outside of the network. Moreover, DT modelling of a subsystem can be integrated with other models to evolve the DT for other applications.

Chapter 2 Literature Review

A review of the history, challenges, architectures, and applications was carried out. These reviews outlined the gaps and requirements of developing a generic CPS framework.

Section 2.1 describes the background, challenges, and gaps in the research of a GCPSF. The existing CPS architecture is reviewed in Section 2.2 to develop the GCPSF. Section 2.3 outlines the challenges in the physical system of the CPS and the reasons for not implementing an RTOS directly. The issues of the cyber system in the CPS and the implementation of a DT in the GCPSF are discussed in Section 2.4. Section 2.5 presents the challenges of verifying and validating the GCPSF.

2.1 Cyber-Physical System

A detailed background review of CPS showed that this research topic is applicable in various domains and fields. As such, researchers in different domains have defined CPS for specific applications. In the current literature, no GCPSF exists that applies to various domains and applications.

As widely known, the emergence of CPS is the result of a convergence of technologies from multiple domains such as computer science, embedded systems, distributed networks, cloud computing, robotics, and automation. Figure 5 displays a timeline of CPS mainly from the networked control systems perspective. The evolution of control systems has enabled various tasks to be performed autonomously. Moreover, these control systems have been extended into the development of communication systems. Advanced Research Projects Agency Network (ARPANET) pioneered using the Transmission Control Protocol and Internet Protocol (TCP/IP) protocol suite and packet switching with distributed control. Subsequently, ARPANET evolved to be known today as the internet.

Subsequently, in 1997, the IEEE 802.11 Wi-Fi Standard enabled systems to communicate wirelessly over networks. Therefore, it can be argued that control systems in the physical world and communication systems were the main technologies that laid the foundations for CPS [12]. Then, various architectures were developed in the following years, such as 5C architecture [13], RAMI 4.0 [14], IIRA [15], and DT [16], further enhanced CPS research.

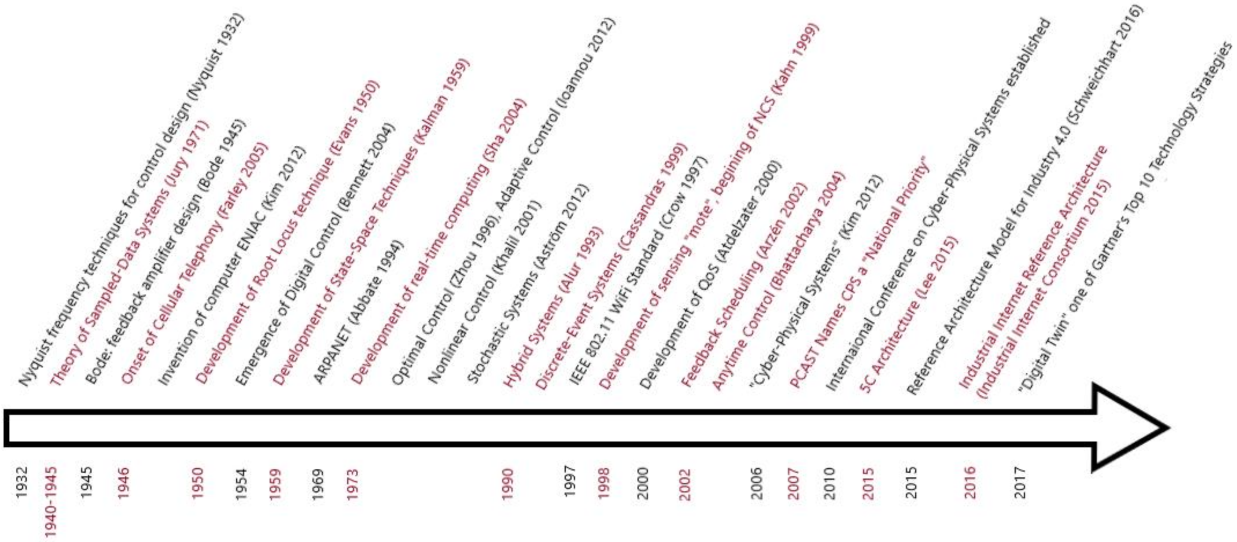


Figure 5 History of CPS adopted from [12].

The term CPS was coined by Dr. Helen Gill as ‘the integration of computation with physical processes using feedback loops across networks that affect both physical and cyber systems’ [2]. The medical, aerospace, automotive, energy, chemical, materials, civil, manufacturing, and rail domains have benefited from CPS [17]. Table 3 illustrates the applications of CPS in the major research domains.

Table 3 Sectors that benefit from CPS and their corresponding applications.

Domain	CPS Application	Reference
Medical	Life critical, context-aware, networked systems of medical devices	[18-22]
Aerospace	Safe and robust real-time control over vehicles	[23-27]
Automotive	Safe and robust real-time control over vehicles	[28-32]
Energy (smart grid)	Safe real-time grid wide voltage control	[33-37]
Civil	Structural health monitoring and control	[38-42]
Manufacturing	Improved production efficiency and effectiveness	[13, 43-46]
Rail	Track vulnerability assessment for security and vehicle control	[47-51]

Because CPS design concepts are non-domain-specific to developing systems for various applications, there are numerous challenges to overcome. The main challenge of developing a CPS in any domain is the lack of a generic framework that accounts for the varying requirements of different fields of research. As of date, there are numerous domain-specific frameworks to overcome the specific challenges of developing a CPS.

The existing frameworks in literature cover aspects of design [52-55], modelling [52, 56-61], securing [62-66], and benchmarking [67] the CPS. In designing a CPS, [52] proposed a generic framework for the design, modelling, and simulation of the CPS; the framework is separated into eight features in Table 4.

Table 4 Requirements of a generic CPS framework.

Requirements	Description
Heterogeneous applications support [52]	A need for simultaneous heterogeneous application logics in non-homogeneous CPS applications.
Various physical modelling environments [52]	A need for fusing domain-specific physical models.
Scalability support [52]	A need for scaling development and simulations of numerous sensors and actuators.
Mobility support [52]	A need for modelling systems using relevant properties.
Integration of existing simulation tools [52]	A need for linking existing simulation tools from different domains.
Integration of proprietary solutions and open standards support [52]	A need for incorporating of protocols, infrastructure, and existing software for a generic framework.
Software reuse [52]	A need for exploitation of code generation techniques, configurable components, and linking libraries.
Usability [52]	A need for representing models and simulations using graphics for simplified development of new applications.

These features describe the requirements for designing a CPS from the ground up. Moreover, [54] introduced a framework with a suite of tools to account for undesirable cross-layer interactions by eliminating integration-time. The elimination of integration-time errors ensures that the latency across the entire CPS is accounted for, resulting in a CPS capable of real-time networked control. The integration of networks and physical systems has also affected the mobile industry. A novel mobile CPS for crowdsensing, Vita, implements the benefits of computation in social, service, cloud, and several open-source techniques across mobile devices and cloud platforms for application developers and users [53]. A similar CPS framework exists that uses physical systems with low-powered microcontrollers to control mobile agents. The CPS framework is based on rewriting the logic, and a Reflective Russian Dolls model [68, 69] of coordination for managing agent activities was developed by [55]. The framework focuses on the abstraction of agent behavior, classification of agent skills, formal design space to represent design decisions, and the study of adaptability versus predictability. These create a framework for designing domain-specific CPS, but there is no step-by-step process for developing a generic CPS from physical systems up to cyber systems.

Besides designing, frameworks on modelling CPS describes various knowledge models and techniques to integrate continuous-time and discrete-time models. An application framework for a networked CPS was developed based on a partially ordered knowledge-sharing model [56]. Moreover, a general framework for quantitative modelling was developed using a multi-agent model that bridges the cyber and physical networks and facilitates the accurate study of their interdependencies [57]. [60] introduced an open-source framework Ptolemy II [61] as a tool for heterogeneous, concurrent modelling and design that allows a combination of discrete and continuous models. In addition, the energy system’s open-source co-simulation framework for formal model description and couplings are Simantics [58] and Mosaik [59]. These frameworks are specific to the field of study and do not integrate models representing different aspects of the CPS.

There are also frameworks for the security aspects of developing a CPS. [62] proposed a generic framework on top of unreliable networks for safety-critical systems, namely the network-aware supervisory system (NASS). The NASS provides an abstraction of a reliable network to designers and the transformation of distributed device control vectors for security. Additionally, a generic CPS model was outlined by [63] with an attack surface suitable for security analysis. A framework for securing smart grid CPS by impact analysis on cyber-attacks was presented in [64]. An adaptive multi-tiered framework for CPS designers was proposed and implemented on a motor for processing anomalous data and taking information security into account [65]. Moreover, [66] developed a logical framework using a Simplex reference model to assist developers with CPS architectures, which limit fault-propagation. The focus of Simplex is the safety of the CPS and the consideration of real-time quality of service. Lastly, there is the Failure Analysis and Reliability Estimation (FARE), a framework for benchmarking the reliability of the CPS [67]. These frameworks apply to specific parts of the development of a CPS. However, there is no step-by-step generic framework for designers and system integrators to develop a CPS from the ground up or on top of existing systems.

Developing a CPS framework that is applicable in various fields is a challenging process. Multiple aspects from different fields of study must be included. A list of challenges is listed in Table 5.

.

Table 5 Research challenges for developing a generic CPS framework.

Challenge	Description
Real-time system abstraction *[4]*	It is a challenge to abstract the salient features of real-time systems using distributed real-time computing and communication mechanisms.
Robustness, safety, and security	CPS requires robustness towards interactions with the environment, errors in physical devices, and security attacks.
Hybrid system modelling and control	CPS demands the development of a hybrid model that combines various models (such as physics, mathematical, network, and data) for control.
Control over networks	Networks introduce latency issues that require network models to model time-varying delays and transmission failures. Control law design and real-time computation in CPS is needed to bridge the gap between continuous and discrete systems.
Verification and validation	Hardware, software, operating systems, and middleware require verification and testing to guarantee the overall CPS requirements. Today's systems typically use overdesign, but new methodologies and tools are needed to incorporate verification and validation at the design stage of the CPS.
Architecture	The CPS architecture requires collecting massive amounts of data from physical systems and communicated across network layers.

Each system within the CPS brings a set of specific challenges. The challenges are separated into three sub-systems: physical, communication, and cyber.

2.1.1.1 Physical system challenges

Every component of the CPS must be thoroughly thought of to ensure the functionality from the lowest layer to the highest layer. On the physical layers, the abstraction of time is challenging [4, 5, 70].

The control systems in the physical layers require performance as close to real-time as possible. However, a CPS in aerospace transportation has dimensional restrictions, limited computing power, restricted communication bandwidth, and large latencies [71]. The real-time performance is crucial due to latency introduced from the other layers. In addition, latency management in physical systems is more straightforward compared to networking systems. A standard solution to develop physical systems to operate in real-time is to use an RTOS. However, the timing behavior of RTOSs is coarse and unreliable in complex systems. High-level programming typically does not take into account the performance of the system in real-time [5]. Thus, robust scheduling is one of the challenges of developing a CPS [70]. However, the autonomous vehicle in this work takes concepts from the RTOS to ensure its functionalities run as close to real-time as possible.

2.1.1.2 Communication system challenges

After the construction of the physical system in the CPS, the next step is to bridge the communication between the physical and cyber systems. The fusion of the systems requires context-adaptive synchronisation methods for efficient control over networks [70].

There are various communication protocols for microcontrollers with their sensors and actuators alone. I2C, SPI, CAN, UART, digital, and analog are the standard communication protocols for microcontrollers to obtain data from sensors and other microcontrollers. For production systems, communication to legacy machinery is challenging [70]. Moreover, these data are insecure because of the real-time performance requirement. Thus, the wires of these components must be kept within a secure location to prevent tampering.

The next stage of communication occurs in a wireless network that bridges the gap between physical and cyber systems. For applications in close quarters, Bluetooth, radiofrequency, and infrared are suitable choices. However, for practical validation, long-range communication across the internet is essential. Standard Internet of things (IoT) data protocols include HyperText Transfer Protocol (HTTP), Constrained Application Protocol (CoAP), Data Distribution Service (DDS), Lightweight Machine to Machine (LWM2M), and Message Queuing Telemetry Transport (MQTT). Choosing a suitable protocol with security features is essential to secure data transmission across networks [4, 72].

One of the significant challenges of developing a CPS in the communication system is security. Catastrophic failure may occur if an attack is carried out on a safety-critical system such as power generators. Manipulating the sensor readings to increase the generation of power will overload the generators and cause destruction. Therefore, a literature review is necessary to understand the security issues when building a CPS.

Before security features can be added, understanding what and how threats are imposed is preliminary [3, 73]. A CPS has multitudes of components connected both physically and wirelessly. Physical components comprise either one of the most straightforward or most laborious parts of securing the CPS. The former occurs when the plant of operations is not heavily guarded. The latter poses a challenge to the attacker as physical clearance is kept within a tight circle. Wireless communication provides the CPS with a vast range of possible services, but the security of safeguarding it is a challenging task. The CPS can be viewed in information technology (IT) form only when discussing the cyber portion. However, the inclusion of safety-critical physical systems distinguishes CPS from traditional IT security.

Traditional IT uses software patching and frequent system updates to improve security features. However, a CPS is not suited for this methodology since the CPS typically runs on RTOSs or plants that are very costly to shut down to update their software. Even if the software update could run parallel with the current setup, issues arise when a significant variable is accidentally modified during the update that causes the system to detect a fault. Moreover, real-time systems operate in austere environments that restrict any delay for safety purposes. Legacy systems are a large part of the automation industry that has little to no communication devices and have close to zero security features. This is one of the most prominent challenges of introducing CPS to established industrial processes. The level of security of communication protocols across systems must be equal [72]. These control systems have a simple network topology, and a network intrusion detection system can be an effortless solution [3]. This thesis implements simple security measures that can be further improved using literature methods.

An essential piece in CPS and DT is the communication of data across systems. There are offline and online communication protocols commonly used, ranging from small-scale projects to industrial automation. Data is transmitted using a wired network. In the past, messages transmitted in the lines were binary, and Morse code gave the meaning of the bits. Nowadays, information transfer across wired networks varies depending on the size and distance. Standard short-range networking uses Ethernet in the manufacturing sector. However, within Ethernet communications, numerous protocols are available. These protocols link to specific device manufacturers. The standard Ethernet protocols include Modbus, Profinet, and others. These devices with different protocols create a need for an integrating component to decipher each protocol into information.

2.1.1.3 Cyber System Challenges

Upon leading the CPS to the internet, securing it becomes one of the most significant challenges. Numerous IoT service providers offer servers and services that connect multiple physical systems into a single application. Nevertheless, the validation process of the built CPS requires full control and customisation ability to ensure the cyber system functions exactly as planned.

2.2 Existing Architectures of Cyber-Physical Systems

One of the objectives of this work is to develop a functional CPS framework, which requires a detailed review of the available architectures. The framework development included a review of numerous popular CPS architectures as well as the available DT architectures. These architectures in Table 6 served as a guideline by integrating the advantages of each to develop the proposed GCPSF.

Table 6 List of popular CPS and DT architectures implemented in the development of the proposed GCPSF.

CPS Architecture	Year	Reference	Essential Implementation in Framework
3C	2011	[74]	Robust physical system to account for environmental factors and attend to computation, communication, and control
CPS architecture	2013	[75]	Data management module to bridge the gap between the physical and cyber world
5C	2015	[13]	Work-flow manner detailing the construction process for a CPS from data collection to analytics at the cyber level
RAMI 4.0	2016	[14]	Maturity levels of the architecture described the requirements of improving the generic framework
Seven-layer architecture	2017	[76]	Communication architecture to signify the communication in CPS and which protocols are suited for generic purposes
Six-layer DT with aggregation	2019	[77]	Implementation of a DT in the framework, which is beneficial because the DT architecture showed similarities with CPS architectures that are interchangeable
DT-driven CPS	2019	[78]	Implementation of a DT in a manufacturing CPS, showing the importance of implementing a DT in the cyber system of the framework

One of the earliest CPS architectures was developed in 2011 [74]. The 3C architecture describes the physical platforms that support CPS for computation, communication, precise control, remote collaboration, and autonomy. Figure 6 defines the three Cs of developing a CPS as computation, control, and communication. This architecture illustrates the requirements of developing a CPS and acts as a baseline for the GCPSF. Strict power-constrained systems typically have unreliable communication and limited computing power, signifying the research objective of developing a near-real-time physical system.

The uncertainties of environmental factors cannot be modelled deterministically. Therefore, a CPS requires a robust control system that adjusts the behavior of the model based on empirical observations in different environmental conditions.

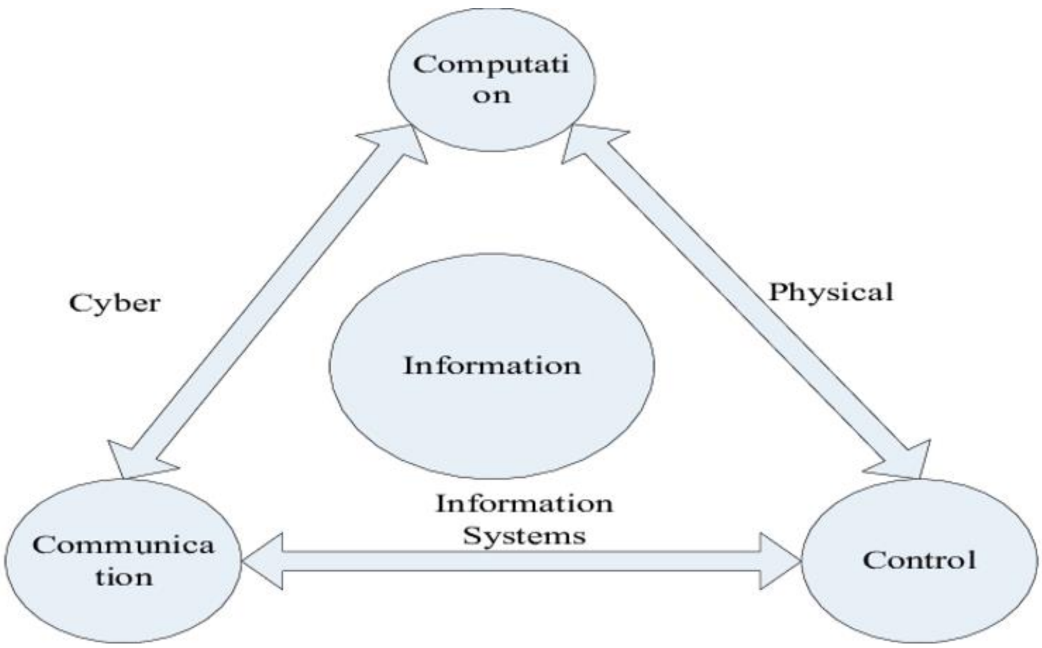


Figure 6 The 3C Architecture from [74] was one of the earliest CPS architectures.

The proposed architecture consists of six modules that support CPS development in Figure 7 [75]. The sensing module supports multiple networks for data collection for real-time control. The data management module consists of computational devices and storage for normalisation, noise reduction, and data storage that bridges the physical and cyber worlds using Next Generation Internet. Next Generation Internet implements new research, Quality of Service (QoS) routing for communications. Service aware modules perform the decision-making, task analysis, and task scheduling using the data from the sensing module. The application module provides services and interacts with the Next

Generation Internet with authentication access security. Lastly, sensors and actuators obtain commands from the application module and carry out specific tasks.

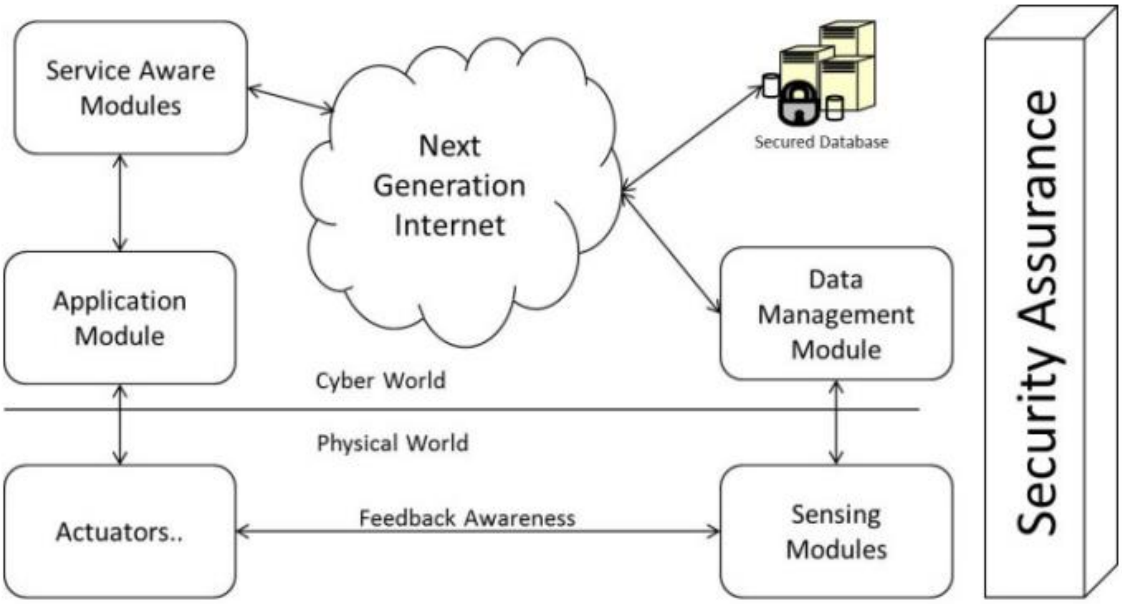


Figure 7 Architecture proposed by SH Ahmed, G Kim, and D Kim [75] to support CPS development.

5C architecture (Figure 8) is a step-by-step guide for developing and deploying CPS for manufacturing applications [13]. The pyramid shape represents the number of components in each layer, which reduces upwards. GCPSF proposed in this thesis implements all the functionality in addition to latency management, which was missing in 5C architecture. 5C functions in advanced connectivity that ensures real-time data acquisition from physical and cyber systems for feedback. The architecture described intelligent data management, analytics, and computational capability that constructs the cyber system. The requirements stated are not specific enough for implementation purposes in general. Therefore, 5C architecture 'clearly defines, through a sequential workflow manner, how to construct a CPS from initial data acquisition to analytics to final value creation' [13].

At the smart connection level, a seamless and tether-free method of managing data acquisition and transferring data to a central server is required. However, this vague statement does not show the possible solutions to acquiring real-time data from the sensors and actuators connected to the local controllers.

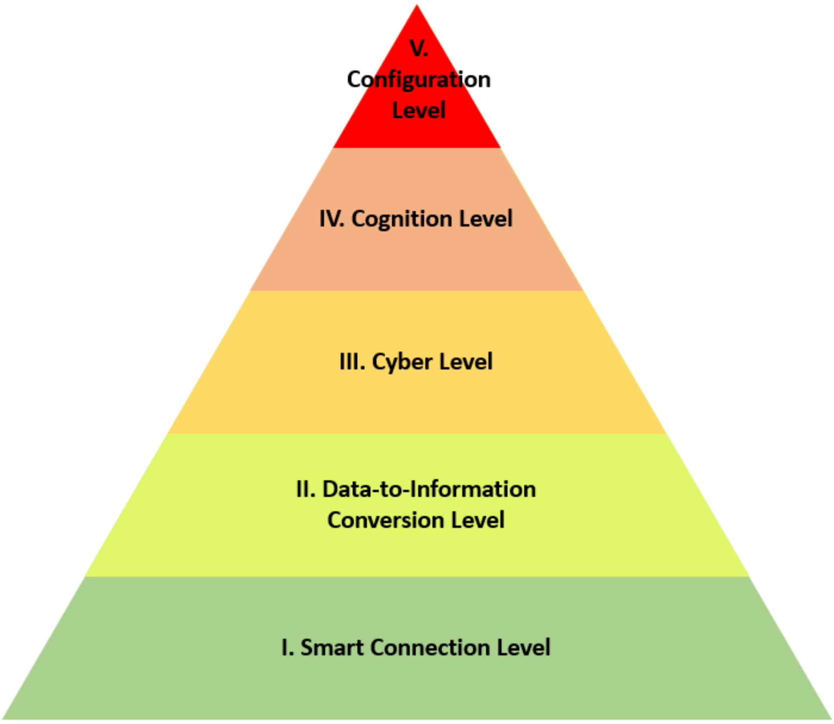


Figure 8 The 5C Architecture in [13] is a more detailed CPS architecture.

In Figure 9, RAMI 4.0 was proposed as a three dimensional architecture for developing a CPS [14] and can be used to illustrate the possible maturity levels of development. The maturity levels (Table 7) cover comprehensive monitoring all the way towards the cooperation of globally networked systems and provide indications of the possibilities of the CPS at that level.

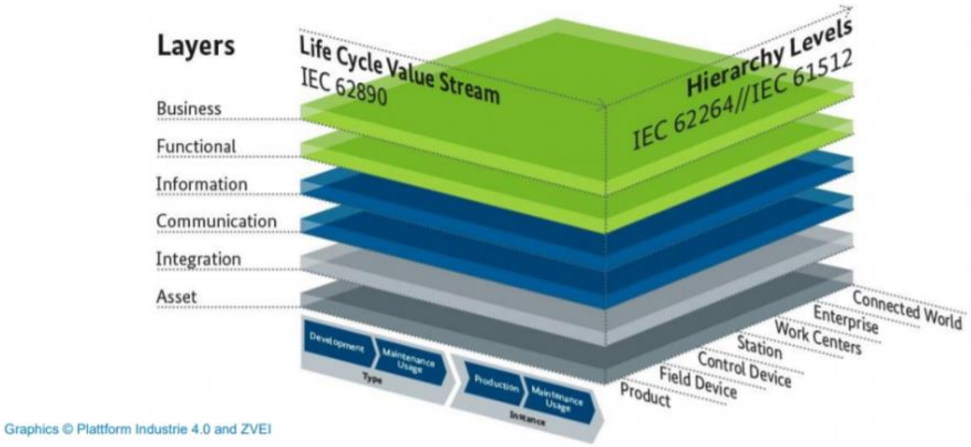


Figure 9 Reference Architectural Model Industry 4.0 (RAMI 4.0; 67) [14].

Table 7 Description of the maturity levels of CPS.

Maturity Level	Description
1 – Monitoring	Obtaining, processing, and storing physical data to analyse the behavior of the actual system [79].
2 – Communication and Analysis	Analysing historical and real-time data of partially interlinked systems to provide information to other systems [79].
3 – Interpretation and Services	Interpreting and enriching recorded data with external data and services to provide recommendations for conforming system behavior [79].
4 – Adaption and Optimisation	Machine learning and adapting its behavior and configurations according to the combination of its own and external partly disordered data [79].
5 – Cooperation	Intercommunication between globally linked systems that negotiate their behavior [79].

The architecture in Figure 10 identifies the interactions in a complex CPS. The interaction begins with physical systems communicating with one another using M2M and V2V. V2I and M2P describe communication across physical and cyber systems. On top of these, B2B, B2C, and P2P represent the interaction between business entities and people [76].

The architecture describes the method to choreograph the interactions between systems for outcomes. Assets to Automation (A2A) describes the progression of connected assets to produce and process data to analyse and automate simple tasks for complex workflows.

Figure 10 lists the protocols needed for IT and Operational Technology (OT) interactions. OT standards for CPS must deliver synchronisation and determinism to ensure near-real-time performance. IT wide area network standards cater to the needs of both ends of the spectrum, including high-power/high-bandwidth and low-power/low-bandwidth connections.

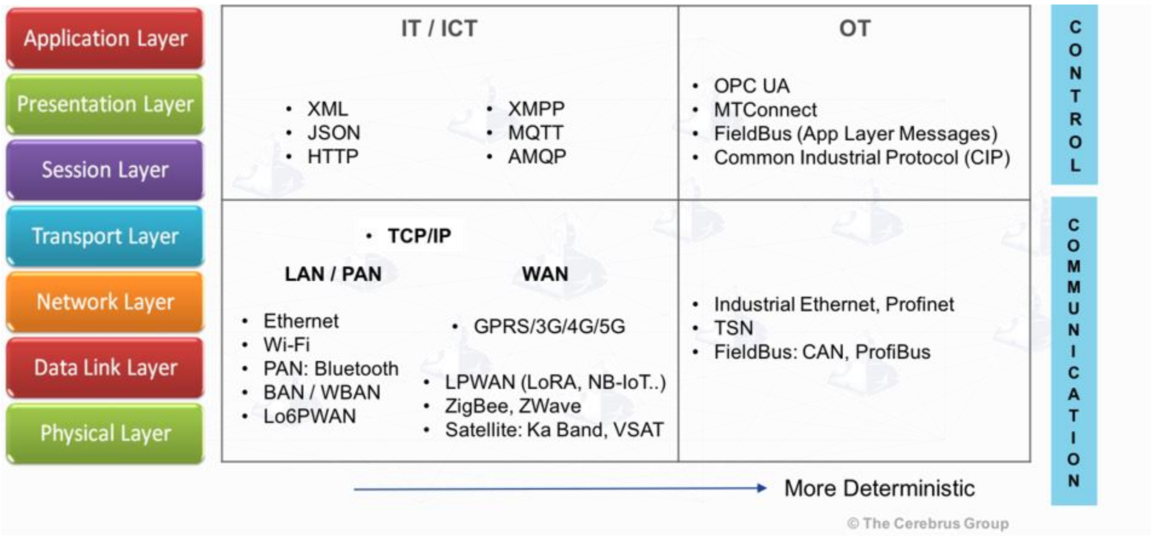


Figure 10 The seven-layer CPS architecture from [76] describes the communication systems in a CPS.

The six-layer architecture for a DT with aggregation (Figure 11) describes the connections between physical twins and DT and their corresponding elements [77]. In Layer 3, local data repositories are connected in a hierarchy to implement Object Linking and Embedding for Process Control Unified Architecture (OPC UA). The IoT gateway (Layer 4) controls the flow of information between the DT. The IoT gateway also has the capability of configuring the interconnections between the DT on Layer 5. However, the research focused on communication and cyber systems. Therefore, implementing a DT in the CPS framework would be beneficial.

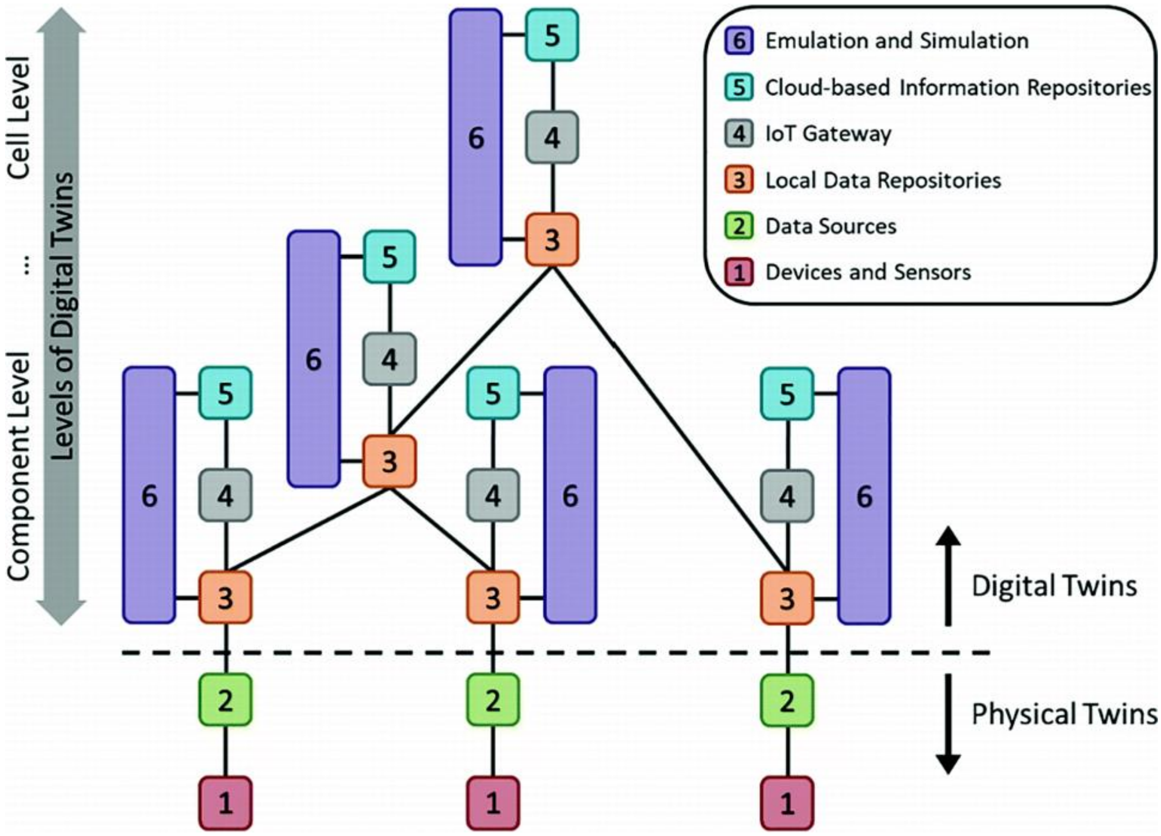


Figure 11 The six-layer architecture for a DT with aggregation in [77] shows similarities with the CPS schematic.

The architecture in Figure 12 [78] describes a DT-driven manufacturing cognitive loop that translates the individualized demands into control parameters. The architecture aids the development stages from planning to manufacturing. For quality control, the DT is interfaced with the inspection database for individual parts.

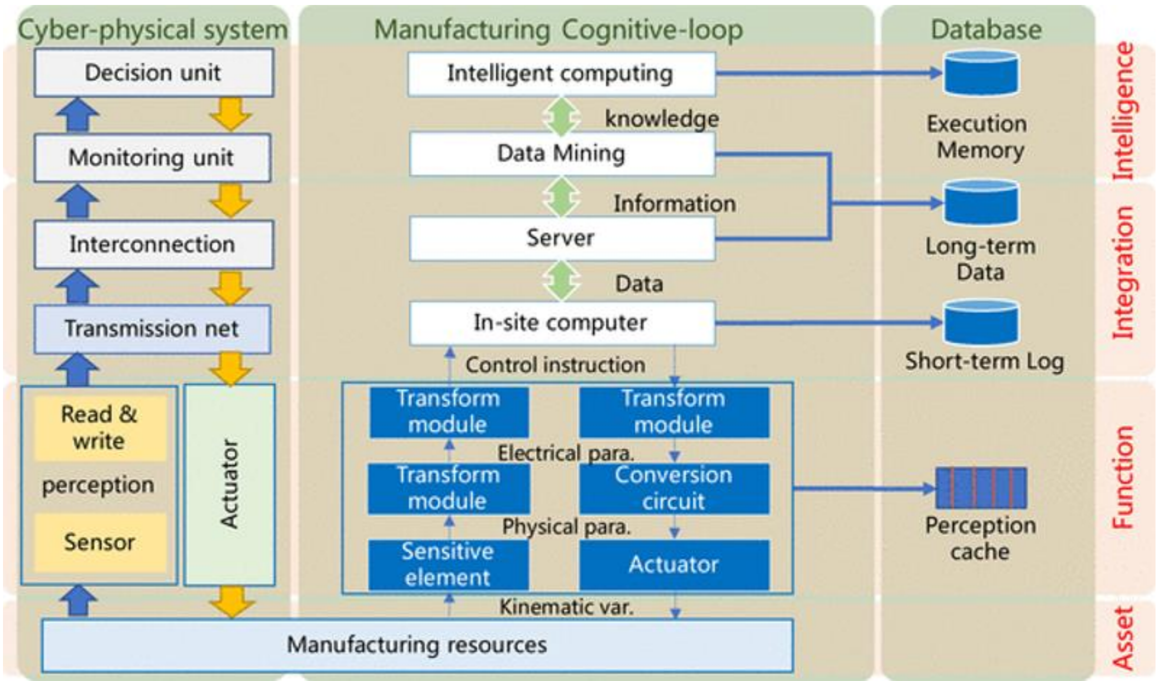


Figure 12 The DT-driven manufacturing CPS in [78] illustrates the latest integration of CPS and DT in a manufacturing system.

2.3 Issues in the Physical System When Implementing A Real-Time Operating System Directly into the Cyber-Physical System

On time-sensitive physical systems, real-time response is a requirement. These physical systems were the benchmark to satisfy the requirements of being a GCPSF. The real-time interactions with the physical system's environment, accompanied by communicating information to a cloud server, raised the requirements of developing an effective CPS. The microcontroller had to function in real-time as well as transmit essential data across networks.

RTOSs had existed since the 1980s when Andre van Tilborg launched a Real-Time System Initiative by the United States Office of Naval Research [78]. The IEEE Real-Time Systems Symposium is one of the main forums for publishing key results on this scheduling theory [80]. Generally, the RTOS is implemented on embedded applications that require running multiple peripherals and middleware in parallel. Traditional methods use custom states and logic, whereas an RTOS operates by choosing which task will be executed next by a priority list [81].

An RTOS provides a fast and guaranteed real-time performance for simple systems. This enables critical processes to be done in a timely manner for system stability and security. Therefore, an RTOS is essential for a CPS because both physical and cyber systems are required to function in parallel. In addition, the embedded system is programmed using a combination of round-robin and pre-emptive priority scheduling [82]. This programming method ensures the processes, from capturing sensor data to filtering, processing, and sending them via MQTT, are completed in a timely manner.

However, complex systems such as CPS give indeterministic behavior [5]. Therefore, there is a need for latency management in the physical system of the GCPSF.

2.4 Uprising of Digital Twin Research-Enhanced Cyber-Physical Framework

The GCPSF requires a cyber system that can accommodate models and data from various fields of research. DT became popular during the duration of the research in 2017 [16] and is applicable on top of the existing GCPSF.

The DT concept is still in its infancy stage. Both academics and industries have their definition of DT. Literature suggests that the DT architecture focuses on the cyber system of the CPS. When the CPS framework is implemented together with the DT, the two complement one another.

In the NASA Apollo 13 mission, two identical spacecraft were built. One went on a journey to the moon while the other stayed on Earth as a physical twin. An exposed wire in the oxygen tank caused a fire, and ground engineers simulated a solution on the physical twin. The crew onboard implemented the solution and managed to return in one piece. The physical twin proved to be an asset to safety-critical systems such as spacecraft. However, the cost of building a redundancy is not encouraging. Thus, DT is a vital component in both newly developed and old systems for real-time improvement and prediction [83, 84].

Figure 13 summarises the timeline of DT. Even though DT was first introduced in 2003, the interest of academics and industry took root in 2017. Therefore, research on DT is a new topic with great potential.

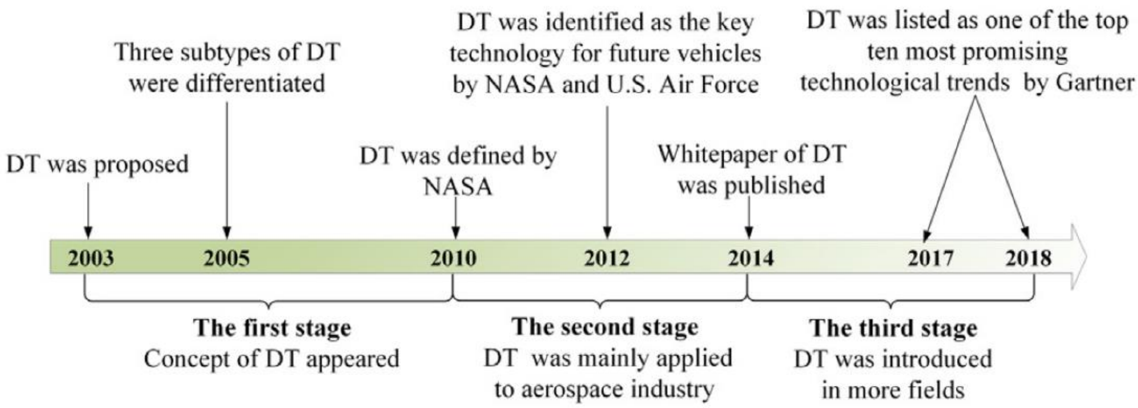


Figure 13 The history of DT adopted from [16] denotes that DT prompted interest from researchers in 2017. When CPS research was evolving, the seamless integration needed by DT was fulfilled.

Since DT is considered a new topic, various researchers have examined them. [83] reviewed the proposed elaboration and work but concluded no same definition of DT. However, [85] identifies the core elements of a DT: model, simulation, and IoT.

A published book divides DT into four categories: Product Health Management (PHM), Product Life Management (PLM), design, and production [16]. Table 8 to Table 11 lists the definitions in each category sorted by the year of publication.

Table 8 Numerous definitions of DT exist in the design community.

Year	Title of Paper	Definition
2016	A simulation-based architecture for smart cyber-physical systems [86]	'The simulation of the physical object itself to predict future states of the system'.
2016	Architecture to geometry—Integrating system models with mechanical design [87]	'A unified system model that can coordinate architecture, mechanical, electrical, software, verification, and other discipline-specific models across the system lifecycle, federating models in multiple vendor tools and configuration-controlled repositories'.
2016	From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems [88]	'From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems'.
2017	A digital twin-based approach for designing and multi-objective optimization of hollow glass production line [89]	'Realistic product and production process models linking enormous amounts of data to fast simulation and allowing the early and efficient assessment of the consequences, performance, and quality of the design decisions on products and production lines'.
2017	C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems [90]	'The cyber layer of CPS, which evolves independently and keeps close integration with the physical layer'.
2017	Innovations in digital modelling for next generation manufacturing system design [91]	'Consisting of a unique instance of the universal digital master model of an asset, its individual digital shadow, and an intelligent linkage (algorithm, simulation model, correlation, etc.) of the two elements above'.

Table 9 DT in production also show different definitions of DT and the increasing need to standardize the definition.

Year	Title of Paper	Definition
2017	A review of the roles of digital twin in CPS-based production systems [83]	‘A virtual and computerized counterpart of a physical system that can exploit a real-time synchronization of the sensed data coming from the field and is deeply linked with Industry 4.0’.
2017	A microservice-based middleware for the digital factory [92]	‘A digital avatar encompassing CPS data and intelligence, representing the structure, semantics, and behavior of the associated CPS and providing services to mesh the virtual and physical worlds’.
2017	Building blocks for a digital twin of additive manufacturing [93]	‘A rigorous validation for additive manufacturing processes, predicting the most important variables that affect the metallurgical structure and properties of the components, and replacing expensive, time-consuming physical experiments with rapid, inexpensive numerical experiments’.
2017	Digital twin as enabler for an innovative digital shopfloor management system in the ESB Logistics Learning Factory at Reutlingen-University [94]	‘A digital copy of a real factory, machine, worker, etc., which is created and can be independently expanded and automatically updated as well as being globally available in real time’.
2017	M2DDM—a maturity model for data-driven manufacturing [95]	‘A digital representation that contains all the states and functions of a physical asset and has the possibility to collaborate with other digital twins to achieve a holistic intelligence that allows for decentralized self-control’.
2017	Toward a digital twin for real-time geometry assurance in individualized production [96]	“A digital copy of a product or a production system going across the design, preproduction, and production phases and performing real-time optimization’.

2018	Digital twin-based smart production management and control framework for the complex product assembly shop-floor [97]	'A dynamic model in the virtual world that is fully consistent with its corresponding physical entity in the real world and can simulate its physical counterpart's characteristics, behavior, life, and performance in a timely fashion'.

Table 10 Definitions of DT in PHM

Year	Title of Paper	Definition
2012	Challenges with structural life forecasting using realistic mission profiles [98]	'An ultrarealistic model of an as-built and maintained aircraft that is explicitly tied to the materials and manufacturing specifications, controls, and processes used to build and maintain a specific airframe'.
2012	The airframe digital twin: Some challenges to realization [99]	'An ultrarealistic model of an as-built and maintained aircraft that is explicitly tied to the materials and manufacturing specifications, controls, and processes used to build and maintain a specific airframe'.
2013	Multi-physics response of structural composites and framework for modeling using material geometry [100]	'Structural model which will include quantitative data of material level characteristics with high sensitivity'.
2013	Multiphysics stimulated simulation digital twin methods for fleet management [101]	'Integrating ultrahigh fidelity simulation with an on-board health management system, maintenance history, and historical vehicle and fleet data to mirror the life of a specific flying physical twin to enable significant gains in safety and reliability'.

2013	Recent advances and trends in predictive manufacturing systems in big data environments [102]	‘A digital model of the real machine that operates in the cloud platform and simulates the health condition with an integrated knowledge from both data-driven analytical algorithms as well as other available physical knowledge’.
2014	On the effects of modeling as-manufactured geometry: Toward digital twin [103]	‘A paradigm that could potentially supplant current structural life maintenance and prediction practices, was conceived to maintain the highest level of personalization in fleet management, thereby circumventing ambiguities like the one detailed herein’.
2015	Isogeometric fatigue damage prediction in large-scale composite structures driven by dynamic sensor data [104]	‘High-fidelity structural model that incorporates fatigue damage and presents a fairly complete digital counterpart of the actual structural system of interest’.
2017	A dynamic Bayesian network approach for digital twin [105]	‘A digital model that flies virtually through the same load history as the actual aircraft wing, integrates various uncertainty sources over the entire life of aircraft wing and heterogeneous information, reduces the uncertainty in model parameters, tracks the time-dependent system states using measurement data, and predicts the evolution of damage states if no data is available’.
2018	Digital twin in the analysis of a big data [106]	‘A digital replica of a real physical installation, which can check the consistency for monitoring data, perform data mining to detect existing and forecast upcoming problems, and use an AI knowledge engine to support effective business decisions.’

2018	The role of data fusion in predictive maintenance using digital twin [107]	‘A living model that continually adapts to changes in the environment or operation using real-time sensory data and can forecast the future of the corresponding physical assets for predictive maintenance’.

Table 11 Research on DT began with definitions of DT in PLM.

Year	Title of Paper	Definition
2012	Challenges with structural life forecasting using realistic mission profiles [98]	‘Ultra-realistic, cradle-to-grave computer model of an aircraft structure that is used to assess the aircraft’s ability to meet mission requirements’.
2012	Modeling, simulation, information technology and processing roadmap [108]	‘An integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin. It is ultra-realistic and may consider one or more important and interdependent vehicle systems’.
2012	The airframe digital twin: some challenges to realization [99]	‘A cradle-to-grave model of an aircraft structure’s ability to meet mission requirements, including sub-models of the electronics, the flight controls, the propulsion system, and other subsystems’.
2013	Multiphysics stimulated simulation digital twin methods for fleet management [101]	‘Ultra-high-fidelity physical models of the materials and structures that control the life of a vehicle’.
2015	Product avatar as a digital counterpart of a physical individual product: Literature review and implications in an aircraft [109]	‘A product equivalent digital counterpart that exists along the product lifecycle from conception and design to usage and servicing, knows the product past, current and possible future states, and facilitates the development of product-related intelligent services’.

2015	About the importance of autonomy and digital twins for the future of manufacturing [110]	'Very realistic models of the processes' current state and its behavior in interaction with the environment in the real world'.
2015	Computationally efficient analysis of SMA sensory particles embedded in complex aerostructures using a substructure approach [111]	'Ultra-realistic multi-physical computational models associated with each unique aircraft and combined with known flight histories'.
2016	Semantic data management for the development and continuous reconfiguration of smart products and systems [112]	'Having a high semantic content and considering both virtual product models as well as feedback data from the physical product along its whole lifecycle'.
2016	Digital twin data modeling with automation and a communication methodology for data exchange [113]	'A set of models from different stages of product lifecycle, such as the system models, functional models, 3D geometric models, multiphysics models, manufacturing models, and usage models, which are kept interacting with each other'.
2016	Industrial IoT lifecycle via digital twins [114]	'Digital representation of a real-world object with focus on the object itself'.
2016	The air force digital thread/digital twin-life cycle integration and use of computational and experimental knowledge [115]	'An integrated multi-physics, multi-scale, probabilistic simulation of an as-built system, enabled by Digital Thread, that uses the best available models, sensor information, and input data to mirror and predict activities/performance over the life of its corresponding physical twin'.
2017	Shaping the digital twin for design and production engineering [116]	'A bidirectional relation between a physical artifact and the set of its virtual models, enabling the efficient execution of product design, manufacturing, servicing, and various other activities throughout the product lifecycle'.

2018	Digital twin-driven product design, manufacturing and service with big data [117]	‘A digital twin is an integrated multi-physics, multi-scale, probabilistic simulation of a complex product and uses the best available physical models, sensor updates, etc., to mirror the life of its corresponding twin’.
2018	Digital twin in industry: State-of-the-art [118]	‘Seamless integration between the cyber and physical spaces’.
2018	Digital twin—proof of concept [119]	‘A collection of all digital artifacts that accumulate during product development linked with all data that is generated during product use’.
2019	Synchronizing physical and digital factory: Benefits and technical challenges [120]	‘A representation of characteristics and behavior of a factory according to various levels of detail and the scope it addresses’.
2019	The digital twin of discrete dynamic systems: Initial approaches and future challenges [121]	‘A digital twin is an avatar of a real physical system which exists in the computer. While a computer model of a physical system attempts to closely match the behavior of a physical system, the digital twin also tracks the temporal evolution of the physical system’.

Industry aims for solutions rather than concepts. The roles of DT for Industry 4.0 era were:

1 To forecast and optimize the behavior of the production system at each life cycle phase in real-time.

2 To exploit DT for quicker virtual commissioning of its assembly lines.

3 To improve the line performance evaluation and optimisation of processing metal sheets [83].

In 2018, various industry players built products and services to market as DT. Each supplier defined DT with the expertise at hand, as shown in Table 12 [122].

Table 12 Definitions of DT by industry leaders show the significance of the topic.

Company Name	DT Definition
Altair [123]	'Integration platform blends physics- and data-driven twins to support optimization throughout the products lifecycle'.
ANSYS [124]	'Combined outstanding simulation capabilities with powerful data analysis capabilities, it is to help enterprises gain strategic insights'.
Dassault [125]	'Through the 3D experience platform, designers and customers can interact with the product during product design or manufacturing process to understand how the product works'.
General Electric [126]	'Through the integration of physical machinery and analytical techniques, the machines are tested, debugged and optimized in a virtual environment'.
IBM [127]	'Occurring in asset-intensive industries that will change operating models in a disruptive way, requiring an integrated physical plus digital view of assets, equipment, facilities and processes.'
Oracle [128]	'Through the virtual models of devices and products, the actual complexities of physical entities are simulated, and insights are projected into applications'.
Parametric Technology Corporation [129]	'The PLM process is extended into the next design cycle to create a closed-loop product design process and help achieve predictive maintenance of the product'.
Siemens [130]	'Based on the consistent data model across all aspects of the product life cycle, some of the actual operations are accurately and veritably simulated'.
Schneider [131]	'A complete 360° digital representation of a physical asset, i.e., a pump, motor, turbine, or an entire plant'.
Systems, Applications and Products (SAP) [132]	'Through building digitized models, product development and innovation are promoted based on real-time data acquisition and analysis'.

Referring to the definitions from Table 8-12, DT does not have a clear, specific, and unique definition in the research community [83, 133]. Nevertheless, the core elements are models, simulation, and IoT [85].

The DT models are separated into five categories in Table 13, each describing different aspects of the physical system for a specified purpose. At the beginning of building a DT, the geometric model must be developed firsthand. Then, fusing additional models on top of the geometric model extends the functionalities of the DT.

In the end, these definitions describe how DT can evolve through time. These show the possibility of consolidating DT into more exceptional services. The gap in the research is the methodology to achieve this end. Therefore, the work in CPS is essential to realize the possibilities of DT.

DT models are separated into five categories in Table 13 [134]—each describing different aspects of the physical system for a specified purpose [135]. At the beginning of building a DT, the geometric model must be developed firsthand. Then, fusing additional models on top of the geometric model extends the functionalities of the DT.

Table 13 The types of DT models that can be integrated

DT Model	Definition	Reference
Geometric	Shape, size, and positions of the physical entity.	[107, 113, 119, 136, 137]
Physical	Function, capacity, and applicable force of the physical entity.	[107, 113, 119, 136, 137]
Behavior	Responsive mechanism under certain factors of the physical entity.	[107, 113, 119, 136-138]
Rule	Associations, constraints, and deduction of the physical entity.	[107, 113, 119, 136-138]
Process	Underlying process of the entity is involved as a part of a CPS.	[97, 107, 119, 137, 139, 140]

The steering system consists of links and joints to manoeuvre the vehicle. Therefore, kinematic modelling methods of robot manipulators were implemented for the DT model. The steering system for an actual vehicle varies in design. Numerous designs aim at achieving Ackermann geometry. However, for this work, a scaled vehicle was used for the validation experiments instead of a full-sized vehicle. With a thorough understanding of the steering system, a DT model of the scaled version was modelled.

The equivalent of an ideal steering system is the Ackermann geometry from Figure 14. The steering angles of both wheels intersect at a single point on the rear wheel axis. Therefore, the vehicle would have a predictable turning radius.

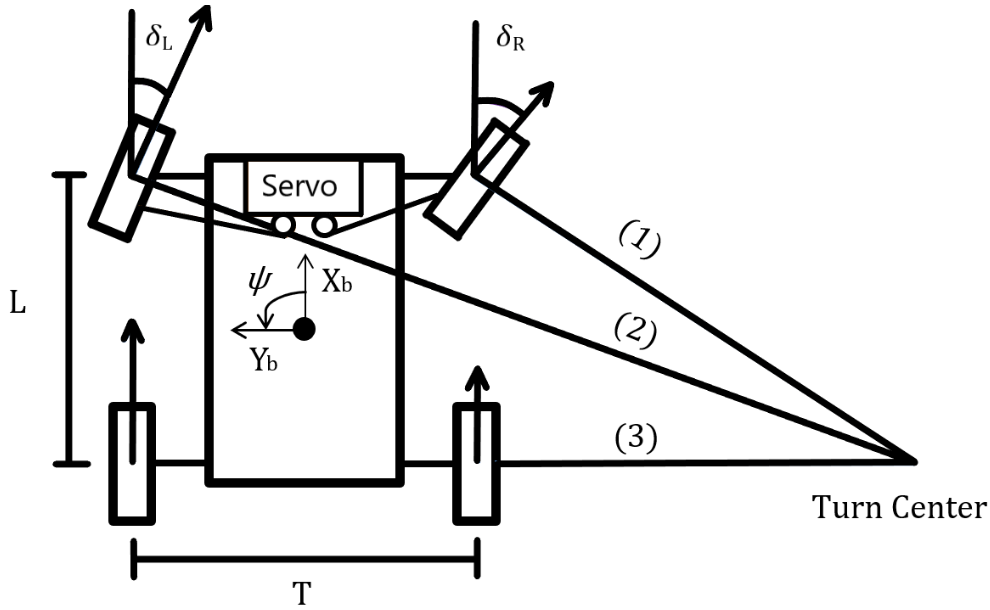


Figure 14 Ackermann geometry for a vehicle with an ISO 8855 vehicle coordinate system.

Developing a DT model started with a geometrical model. Then, the model was further developed into a behavior model. The steering system of the vehicle consists of links and joints to manipulate the steering angle.

The link and joints of the steering system were modelled using Denavit-Hartenberg (D-H). D-H was particularly selected due to a review of other robot manipulator modelling techniques.

Table 14 shows numerous modelling methodologies that were compared according to number of parameters and complexity of method [141]. The complexity of method refers to “the measure of uncertainty in achieving the functional requirements of a system within their specific design range” [142, 143]. Therefore, the simplest method was chosen for the DT of the steering system because the lower parameter and lesser complexity reduced the computation time needed.

Table 14 Comparison of Kinematic modelling methods for robot manipulators.

Method	Sheth and Ucker [144]	Khalil and Kleinfinger [145]	Unified [146]	Quatenion Theory Based [147]	Lie Algebra Based [148]	Screw Based [149]	D-H [150]
Number of Parameters	7	4 or 6	5	8		7	4
Complexity of Method	More	Less	Less	More	More	More	Less

The key findings of the literature review for CPS included significant challenges, architectures, and gaps in the research. A prototype CPS was planned to use the architectures and methodologies of different components. The definition and challenges of DT were discussed to enhance the CPS framework. The comparison and correlation of CPS and DT showed that CPS is akin to a scientific category, while DT is in the engineering category. Research on CPS focuses on theoretical concepts, architecture, and security. Research on DT focuses on the modelling and data of a specified physical system.

2.5 Verification and Validation of the GCPSF

The proposed GCPSF needed to be verified and validated. Verification and validation in CPS are especially tedious and challenging [10]. CPS developers require an easy method to specify desired correctness properties of the application, incorporation of physical world properties into runtime verification processes, and a usable tool suite for the development process of CPS to designers from different fields of study [9]. The current state of CPS verification and validation practices remains an ad hoc trial and error process [10].

Brace Architecture (Figure 15) is an online monitoring framework that replaces the trial-and-error method as the standard CPS practice. The framework expresses the requirement of the Brace Architecture to be intuitive and be accessible to CPS practitioners who are not logicians or do not have an in-depth understanding of temporal logic. The connection to real-time simulation results in a practical tool for software-in-the-loop testing. Additionally, the framework also aims to address uncertainties by automating a process to identify and solve a chance-constrained problem [9]. Besides Brace Architecture, [8] describes the implementation of ModelPlex for building and verifying high-assurance controllers for safety-critical computerized systems that interact physically with their environment. [11] described the verification and validation of a CPS carried out by creating a test environment by manipulating the adjustable parameters to be as simple as possible. The computational system can be isolated from physical systems via hardware-in-the-loop testing. Embedded systems, such as FPGAs, can simulate feedback from physical systems. The execution time and latency are measurements for the refinement of the previous CPS. Therefore, the verification and validation of the GCPSF were performed by customizing Brace Architecture [9] and model-based design techniques in [11].

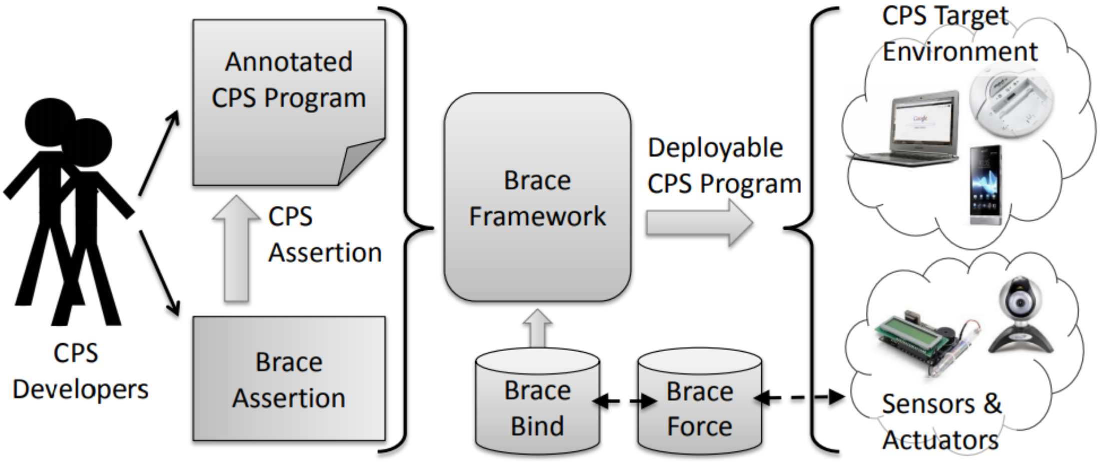


Figure 15 The research challenges in practical runtime verification as described by Brace framework architecture adopted from [9].

2.6 Chapter Conclusion

This chapter discussed the background, applications, challenges, and architecture of CPS. Upon reviewing the literature, the gaps in the study of CPS were identified and converted to problem statements and research objectives in Table 15.

Table 15 Problem statements derived using the gaps in the literature reviews.

Gap in the Literature Review	Reference	Problem Statement	Research Objective (Chapter)
Research literatures contain various architectures, but no generic framework for CPS was found.	[4]	Need for GCPSF	Develop a GCPSF (3)
A robust, reliable, and safe near-real-time physical and cyber subsystems are required for GCPSF.	[4, 5, 71, 74, 151, 152]	Latency issues affecting the CPS performance	Develop a near-real-time physical system (3)
A cyber system that is capable of integrating various cross-discipline models and data is required for the GCPSF.	[4, 74, 151, 152]	Need for a cross-discipline cyber system for the GCPSF	Develop a DT (3)
Requirements for the designing and development of CPS vary widely. Hence, it is tedious and challenging to verify and validate the CPS, but no standard method also exists; most researchers use experimentation to verify and validate.	[4, 151]	Need to verify and validate the GCPSF	Verify and validate the GCPSF (4)

Chapter 3 Development of a Generic CPS Framework

The main objective of this work was to develop a generic cyber-physical system framework (GCPSF). The implementation of the proposed framework needed to be domain-neutral and application-agnostic. The proposed framework provides the fundamentals of developing a CPS from the ground up or on an existing system. The development of the core components is described in the following sections, from design to implementation. Subsequently, the detailed development description of the proposed framework is explained, as well as the methodologies of building a CPS. The GCPSF was used on an ITS for autonomous trajectory planning and control for validation. A second CPS prototype in the manufacturing domain was developed using the framework and COTS components as a second validation.

In this chapter, the development of the proposed framework and implementation of the CPS framework on two applications is discussed. Section 3.1 discusses the development of the GCPSF and its implementation methods. Section 3.2 describes the development of a CPS in the ITS domain using the proposed framework. The section was designed to achieve the objectives of developing a near-real-time physical system and DT model of the CPS. Lastly, Section 3.3 describes the development of a manufacturing domain CPS using the proposed framework and COTS components.

3.1 Developing a Four-Layer Generic Cyber-Physical System Framework

The GCPSF was developed based on the architectures from the literature described in Section 2.2. The GCPSF is displayed as a four layered pyramid in Figure 16. In addition to describing the functional layers, the pyramid works as an illustration to visually convey the reducing number of components going up the layers. The layers adapted from previous architectures described the core functions required in the GCPSF. The components within the pyramid represent specific hardware or software required to perform the functions.

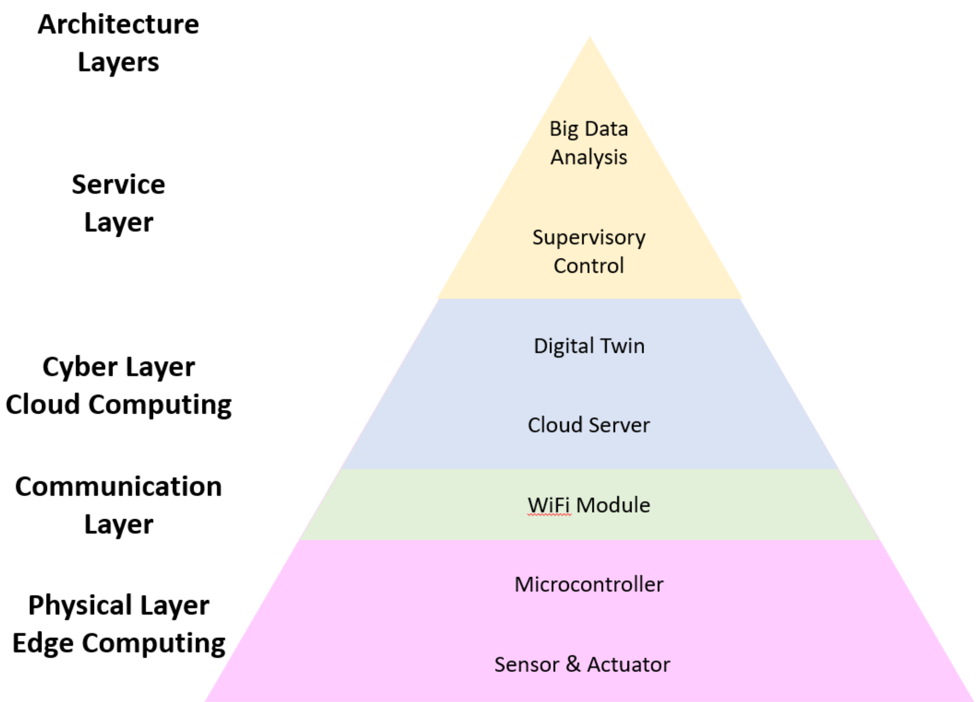


Figure 16 The proposed four-layer CPS framework illustrates how to develop a CPS in any domain or application.

The following subsections explain the functionalities required from each layer and the corresponding components. The proposed four-layer CPS framework covered all aspects stated in the architectures.

3.1.1 Physical Layer

The CPS interacts with the environment in the physical layer, where edge computing is the primary controller. Different fields of study have distinct requirements in the environment where the system operates. For example, medical systems must be sanitary while manufacturing systems must be safe, and safety systems must be responsive. One of the highest requirements for physical systems in CPS is that they have to operate in real-time with networked control. That specific requirement was attempted in this work on a limited computing power microcontroller. Nevertheless, specific industries prefer to have a CPS implemented on top of existing systems. With the limitation of developing the physical system, the other components remain the same, but reaction time decreases.

Every autonomous physical system requires sensors and actuators to interact with its environment. The purpose of these components remains the same in all domains and applications. However, quick response systems, such as safety systems, require higher sampling frequency when compared to measuring water levels in a tank. These details are handled by the microcontroller that controls the sensors and actuators.

A computer of different sizes controls every autonomous system. With the advancement of microcontroller technology, edge computing is the current trend. This is because there is a trend to move computation to the physical system instead of cloud computing. Cloud computing is sending the data from a physical system to a remote computer to analyse the data. Whereas edge computing analyses the data at the physical system. The benefits of edge computing are having faster response time and reducing the dependency on the communication between the cyber system for computation. However, demanding computation like artificial intelligence and machine learning algorithms will need cloud computing. Different fields of study are also limited in the customisation of the microcontrollers due to specific constraints.

Microcontrollers on autonomous vehicles have less than a second to decide between emergency braking or evasive manoeuvring and actuating the correct operations. Therefore, a robust CPS requires the microcontroller to operate the physical system as close to real-time as possible.

Industrial systems typically require safe, reliable, and temperature resistant controllers, limiting the choices of controllers available for the specified function. However, the framework requires the microcontroller to operate sensors and actuators and communicate with a Wi-Fi module or gateway. The CPS designer must decide the other specification of the microcontroller.

3.1.2 Communication Layer

The first communication in a CPS is the transmission of data between the microcontroller and the communication device. Numerous devices from different manufacturers exist that implement distinctive communication protocols. The integration process of communicating data from the microcontroller to a gateway depends on the field of study. Scaled robotic experiments would implement serial communication using I2C or SPI, whereas a manufacturing system would implement Modbus/IP for its communication protocol.

The second communication in this layer is the connection between physical and cyber systems. The data collected from sensors and actuators are sent to the cyber system for processing. The Wi-Fi module or gateway is the main component for the collection and wireless transmission of data. Like the previous communication with microcontrollers, there exists a variety of internet protocols. For the CPS framework to be generic, a small electronic footprint and safe and fast internet protocol is needed. Thus, MQTT was the internet protocol chosen in the communication layer.

3.1.3 Cyber Layer

Cyber systems are the core controlling and data storing components in the CPS. The data obtained from the physical systems are stored in a cloud server for backup and trend detection. Numerous companies provide cloud servers as a service. However, each service provider has advantages and drawbacks. A practical solution for full CPS control is to implement a server controlled by the designer. All the data traffic to and from each system is recorded and monitored by the designer. The proposed server implemented in the CPS framework was the Eclipse Mosquitto MQTT broker. The MQTT broker was running on Windows or a portable Raspberry Pi for different applications. The Mosquitto MQTT broker provides customisable security and rapid communication.

The central computational core of the CPS is the DT. A DT model of the physical system starts as a geometrical model describing the size and features of the physical twin. Additional models can then be developed and integrated on top to improve the functionalities of the DT model. Behavior models using physics modelling methods can predict and control the behavior of the physical system.

3.1.4 Service Layer

Lastly, the service layer is based on the service-oriented architecture (SOA) concepts [153, 154] that aggregates simple service bundles to provide complex applications such as the waypoint-based navigation service. SOA concepts provide the framework the capability of supervising, monitoring, and controlling multiple DT models to analyse large amounts of data collected from physical systems and to provide decision support. The supervisory control oversees all the actions and integration of DT in the CPS. The data collected from numerous sensors and actuators enable big data analysis to determine trends in the physical systems for prediction control and maintenance. MATLAB was used as a platform in this service layer because of its artificial intelligence toolbox for big data analysis and support for DT models.

3.2 Developing a Cyber-Physical System in the Intelligent Transportation System Domain Using the Four-Layer GCPSF

In order to validate the proposed GCPSF, two systems from different domains were chosen. One of the domains was the ITS. ITS is a suitable domain to develop a CPS with trajectory planning and control capabilities. Moreover, a scaled ITS was preferred to illustrate the requirements of robust, reliable, and safe real-time physical systems for the GCPSF. Vehicles typically travel at high speeds to reduce transportation time, which requires the computation time of the microcontrollers to be as close to real-time as possible. In the ITS cyber system, DT modelling of the steering system was developed. The DT modelling showed the possibilities of integrating various models in a single software for various applications. The following sections are separated into a step-by-step explanation of implementing the proposed framework in the ITS domain shown in Figure 17.

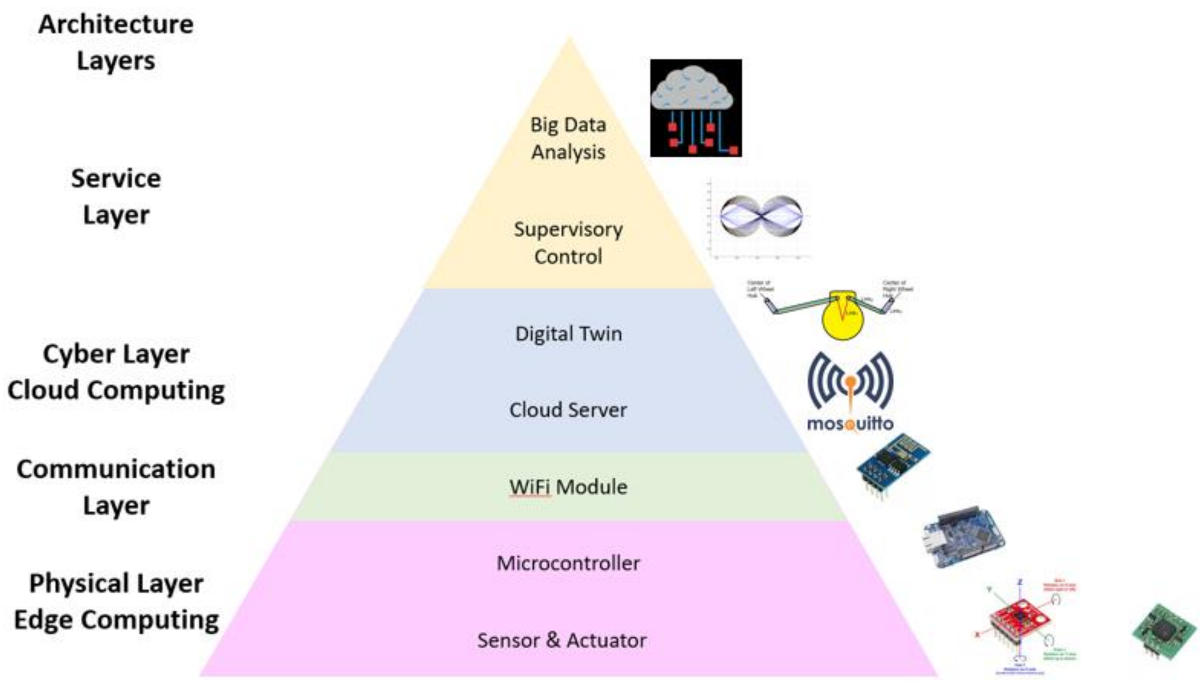


Figure 17 Implementation of the proposed four-layer CPS framework on an ITS. Each image represents a specific component that was chosen in that layer.

3.2.1 Physical Layer

Figure 18 shows the scaled vehicle used in the developed ITS. The vehicle has an accelerometer, magnetometer, gyroscope, and hall effect sensors, as well as a DC servo for steering and two DC motors for drive. The vehicle is battery powered with a limited computing power microcontroller (FRDM-K64F) that operates the vehicle and sends data across wireless connections. The integration and programming of the physical system were done to achieve near-real-time to manage the latency in networking.

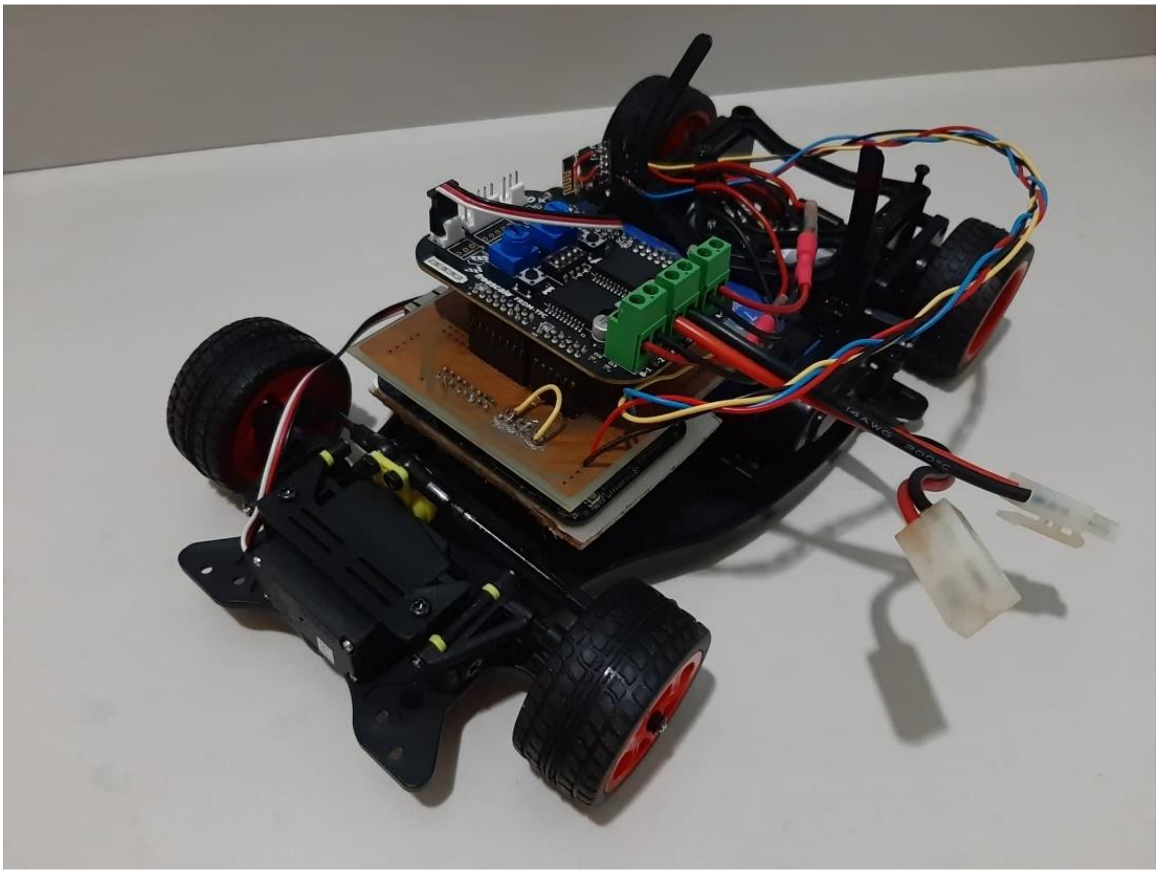


Figure 18 Scaled vehicle used in the CPS for the ITS application.

The vehicle interacts with its environment using the sensors and actuators equipped. As discussed earlier, these vehicles require close to real-time performance to be safe and reliable with networked control. The limited computing power microcontroller was programmed using Mbed, an open-source IoT operating system. Since the direct implementation of an RTOS was not a deterministic method, the program was self-written with assistance from libraries provided by Mbed. The concepts implemented in the program were memory allocation, interrupt service routines, and scheduling function files. The required functions were compiled in a round-robin loop, and the worst-case-scenario time was measured. The written program allows the microcontroller to read, filter, fuse, packet, and send the data to an external Wi-Fi module and SD card within 25 ms in addition to operating as a vehicle. The program achieved the second research objective of developing a near-real-time physical system.

3.2.2 Communication Layer

The first communication in a CPS is the transmission of data between the microcontroller and the communication device. The microcontroller communicates with an ESP8266 Wi-Fi module using serial communication (UART). The second communication in this layer is the bridge between the physical and cyber systems. The ESP8266 Wi-Fi module is also a limited computing power device. Therefore, a review of existing internet protocols was carried out, and MQTT was chosen. MQTT is a lightweight internet protocol that is fast and secure, and suitable for the GCPSF and ITS application.

3.2.3 Cyber Layer

Cyber systems are the core controlling and data storing components in the CPS. The data sent from the communication layer were stored in the Eclipse Mosquitto MQTT broker. Even though there are numerous internet protocol, the primary internet protocol for the GCPSF is MQTT to match the cloud server (MQTT broker). Moreover, designers will have full access to all the data traffic in the MQTT broker and can implement security features from existing CPS security frameworks.

Once the storage for the cyber system was set up, the development of the DT model of the steering system followed. The steering system of any vehicle acts as the primary source of manoeuvring. The trajectory control relies heavily on the steering system performance. The DT model of the steering system starts as a geometrical model describing the links and joints of the system using kinematics. The steering system's links and joints were attached with reference frames, as shown in Figure 19 to Figure 23. By setting the end of link three as a stationary point, inverse kinematics was done to derive individual link angles. The link angles correspond to the relationship between steering input and vehicle steering angle that determines the performance of the steering system. D-H was chosen because it is one of the simple kinematic modelling methods for robot manipulators. After developing a kinematic model of the steering system, a behavior model was built. The model uses inverse kinematics to predict the movements of the links and joints and the corresponding vehicle steering angle.

Figure 19 Left wheel with steering system showing the links of the steering system. The colors red, green, and blue correspond to link 1, 2, and 3 of the steering system for simulation.

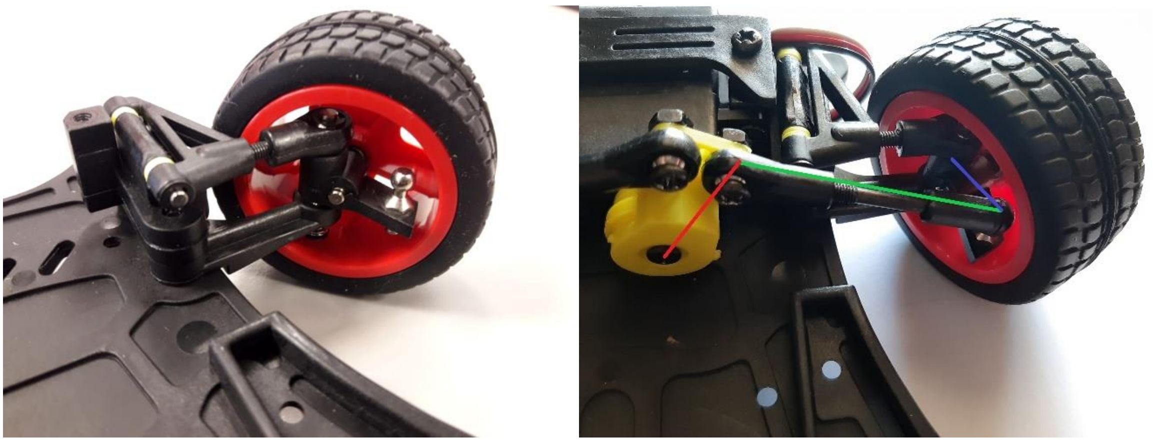


Figure 20 Right wheel with steering system showing the links of the steering system. The colors red, green, and blue correspond to link 1, 2, and 3 of the steering system for simulation.

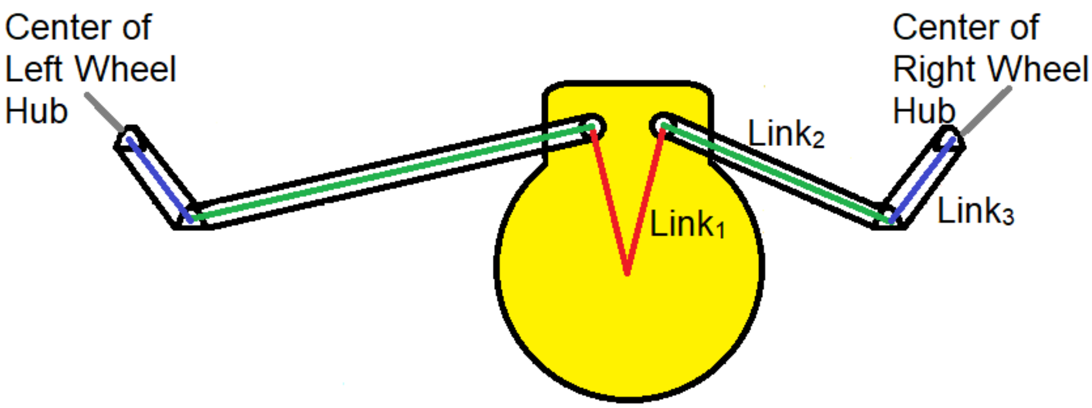


Figure 21 Steering system with D-H reference frames to illustrate the links.

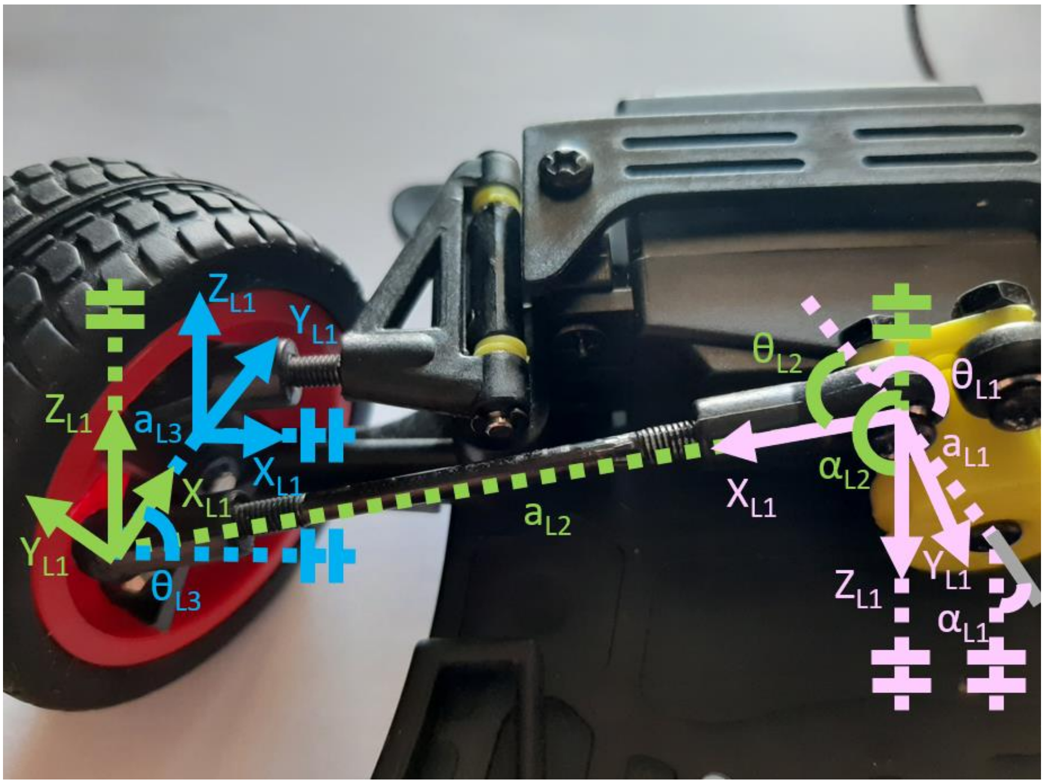


Figure 22 D-H reference frames on the left steering system illustrate how the axes were drawn.

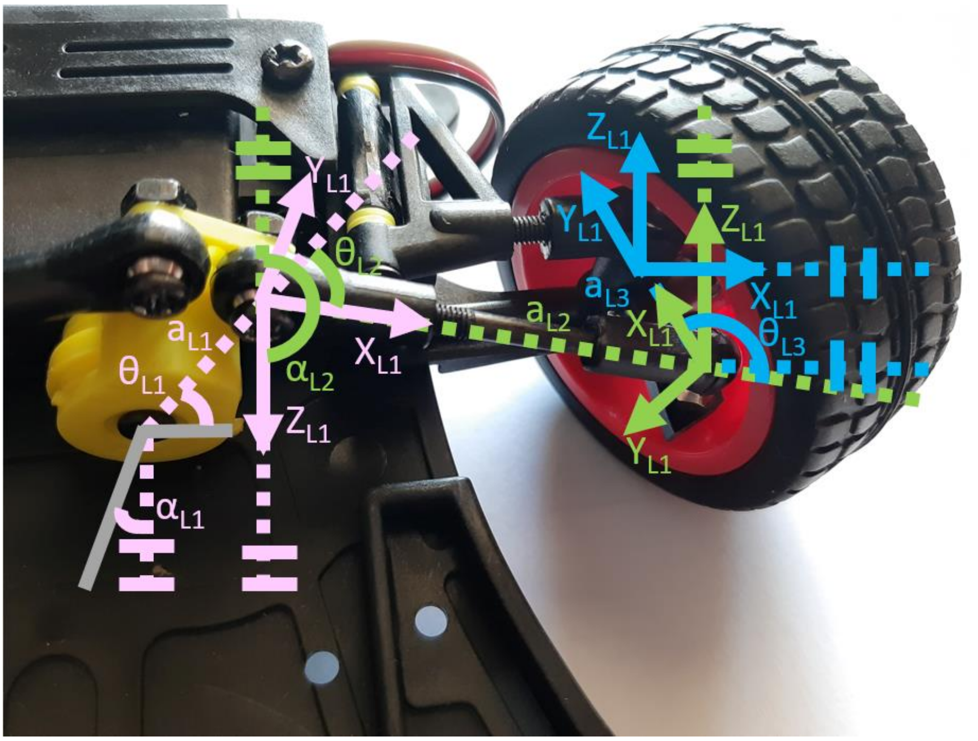


Figure 23 D-H reference frames on the right steering system to illustrate how the axes were drawn.

A D-H table (Table 16) is filled with the variables in Figure 22 and Figure 23. Then, the matrices are derived from the table. a_{L1}, a_{L2}, and a_{L3} are measured link lengths. Multiplying the matrices results in R_{L03} with five variables, α_{L1}, α_{L2}, θ_{L1}, θ_{L2}, and θ_{L3}. However, $\alpha_{L1} + \alpha_{L2} = 90^o$, and θ_{L1} is an independent variable equivalent to the servo angle that controls the steering of the vehicle ($Link_1$). Moreover, the final position at the end of $Link_3$ is the center of the wheel hub measured from the start of $Link_1$. By using inverse kinematic, comparing the X and Y position in R_{L03} with the final position, θ_{L2} and θ_{L3} were solved, giving the steering angle of the vehicle relative to the steering input θ_{L1}.

Table 16 D-H Notation for Steering System

Link	a	α	d	θ
1	a_{L1}	α_{L1}	0	θ_{L1}
2	a_{L2}	α_{L2}	0	θ_{L2}
3	a_{L3}	0	0	θ_{L3}

$$R_{L01} = \begin{bmatrix} cos\ (\theta_{L1}) & -\cos\ (\alpha_{L1})\ sin\ (\theta_{L1}) & \sin\ (\alpha_{L1})\ sin\ (\theta_{L1}) & a_{L1}\ cos\ (\theta_{L1}) \\ sin\ (\theta_{L1}) & \cos\ (\alpha_{L1})\ cos\ (\theta_{L1}) & -\sin\ (\alpha_{L1})\ cos\ (\theta_{L1}) & a_{L1}\ sin\ (\theta_{L1}) \\ 0 & \sin\ (\alpha_{L1}) & \cos\ (\alpha_{L1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{L12} = \begin{bmatrix} cos\ (\theta_{L2}) & -\cos\ (\alpha_{L2})\ sin\ (\theta_{L2}) & \sin\ (\alpha_{L2})\ sin\ (\theta_{L2}) & a_{L2}\ cos\ (\theta_{L2}) \\ sin\ (\theta_{L2}) & \cos\ (\alpha_{L2})\ cos\ (\theta_{L2}) & -\sin\ (\alpha_{L2})\ cos\ (\theta_{L2}) & a_{L2}\ sin\ (\theta_{L2}) \\ 0 & \sin\ (\alpha_{L2}) & \cos\ (\alpha_{L2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{L23} = \begin{bmatrix} cos\ (\theta_{L3}) & -sin\ (\theta_{L3}) & 0 & a_{L3}\ cos\ (\theta_{L3}) \\ sin\ (\theta_{L3}) & cos\ (\theta_{L3}) & 0 & a_{L3}\ sin\ (\theta_{L3}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{L03} = R_{L01} * R_{L12} * R_{L23} = \begin{bmatrix} . & . & . & X \\ . & . & . & Y \\ . & . & . & Z \\ . & . & . & 1 \end{bmatrix}$$

3.2.4 Service Layer

The service layer of the ITS is the application of a Dubins path [155] for trajectory planning. Dubins path is the simplest path that captures the basic motions of a moving vehicle: rectilinear and curvilinear motions, which can completely describe a general motion of a rigid body. Additionally, the Dubins path can be extended for three-dimensional trajectories for unmanned aerial vehicles. However, this trajectory requires the vehicle to manoeuvre at a constant speed throughout its path. The second requirement is the exact locations of the waypoints at which the vehicle changes direction from a straight line to curvature or vice versa. Each Dubins path generated numerous paths with different radii in Figure 24 and waypoints in Figure 25. Figure 24 shows each path requiring a specific turning radius performance from the vehicle. In Figure 25, these points indicate at which the vehicle had to change its direction. Therefore, to be practical, the steering limitations of the vehicle had to be determined using the kinematic and DT models for the Dubins path.

From these limits, an optimum and feasible path was selected by the trajectory planner for the desired manoeuvre.

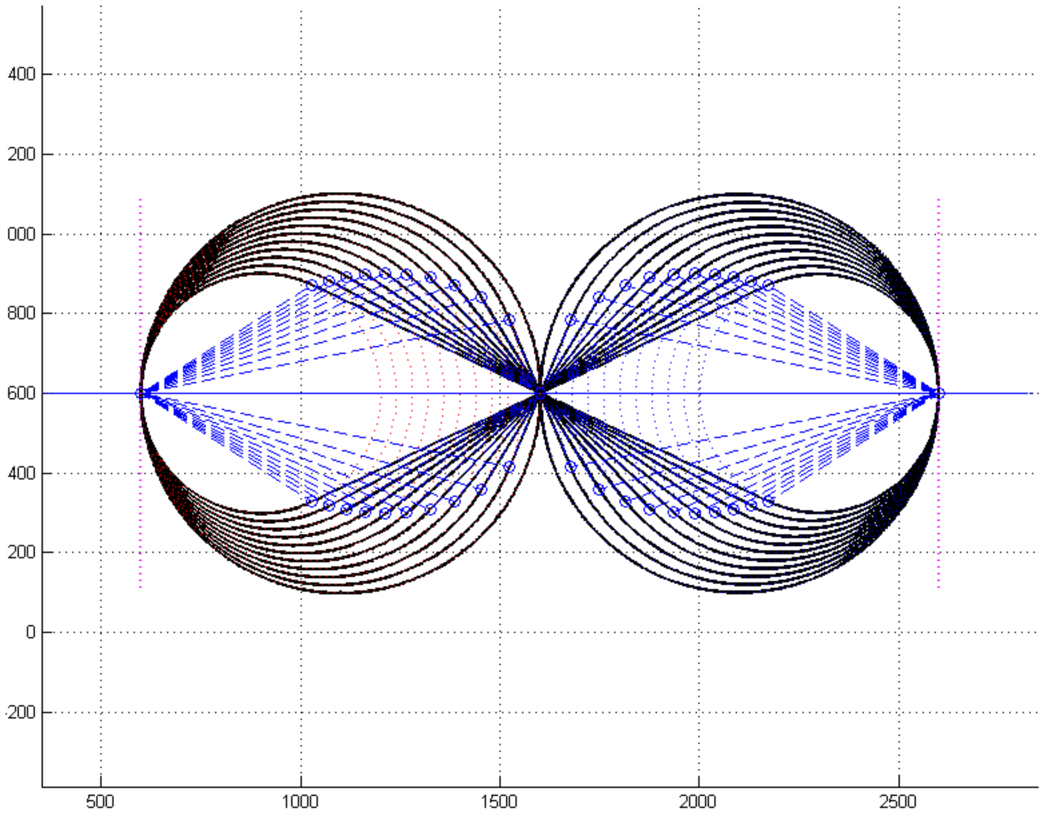


Figure 24 Various Dubins paths for different turning performances.

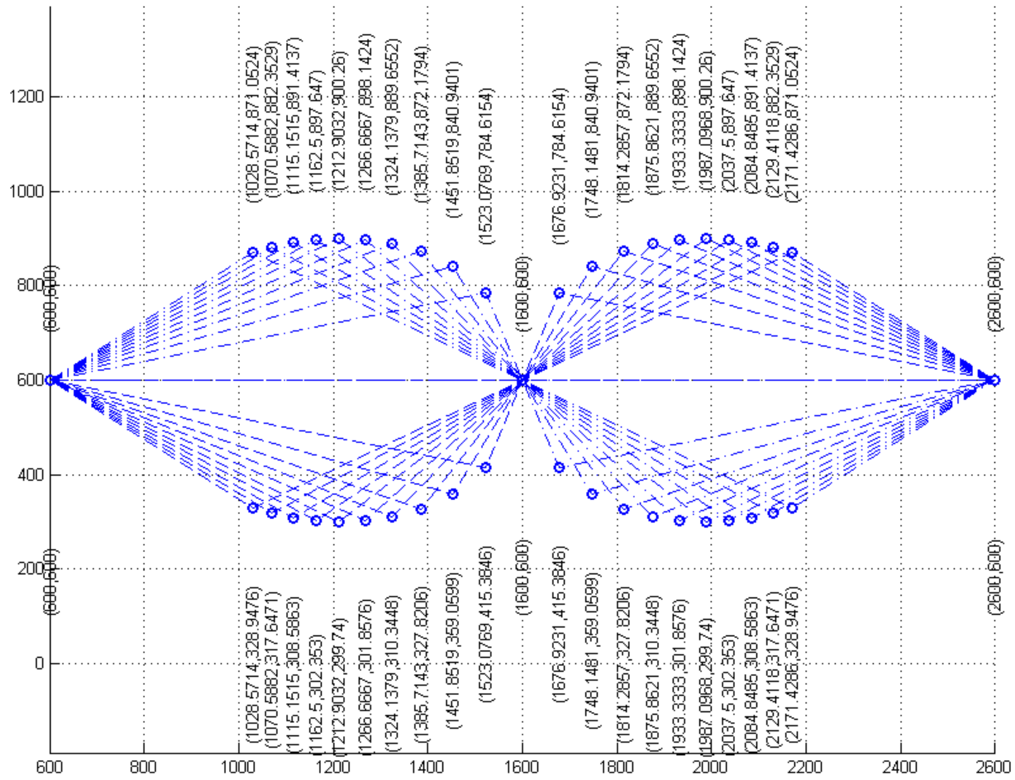


Figure 25 Various waypoints for the generated Dubins paths.

Figure 26 represented the integration of Dubins path planning service [155], the vehicle behavior model [156], and the DT steering model in a singular Simulink block diagram. The user inputs the position and orientation of the vehicle at the start and end position were used to generate the shortest distance Dubins path. The vehicle dynamics model tracked the path using steering angles from the DT steering model and constant speed (40% top speed or 1.3 m/s). Then, Simulink acted as the service layer of the GCPSF and sent the planned trajectory to the physical vehicle.

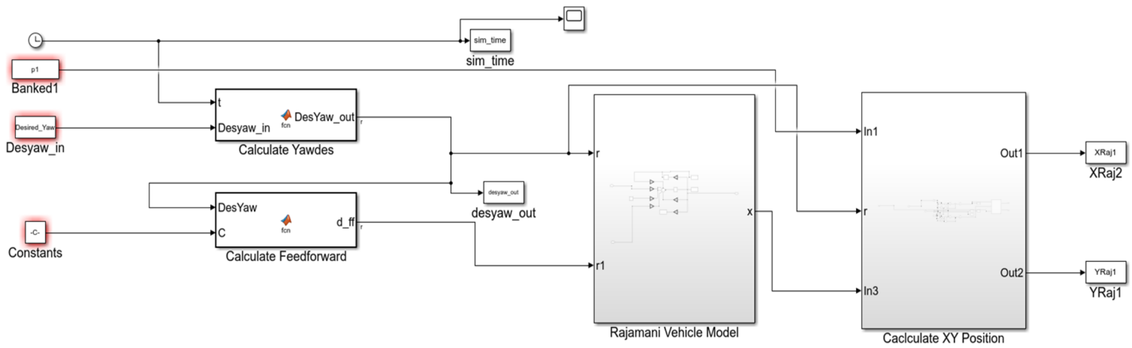


Figure 26 MATLAB Simulink block diagram of the cyber system.

3.3 Developing a Cyber-Physical System in Manufacturing Domain Using the Four-Layer GCPSF

The CPS functional framework aimed to contribute a generic concept without being domain or application specific. A manufacturing CPS was built using off-the-shelf-solutions based on the framework in Figure 27. The manufacturing CPS acts as an industrial benchmark.

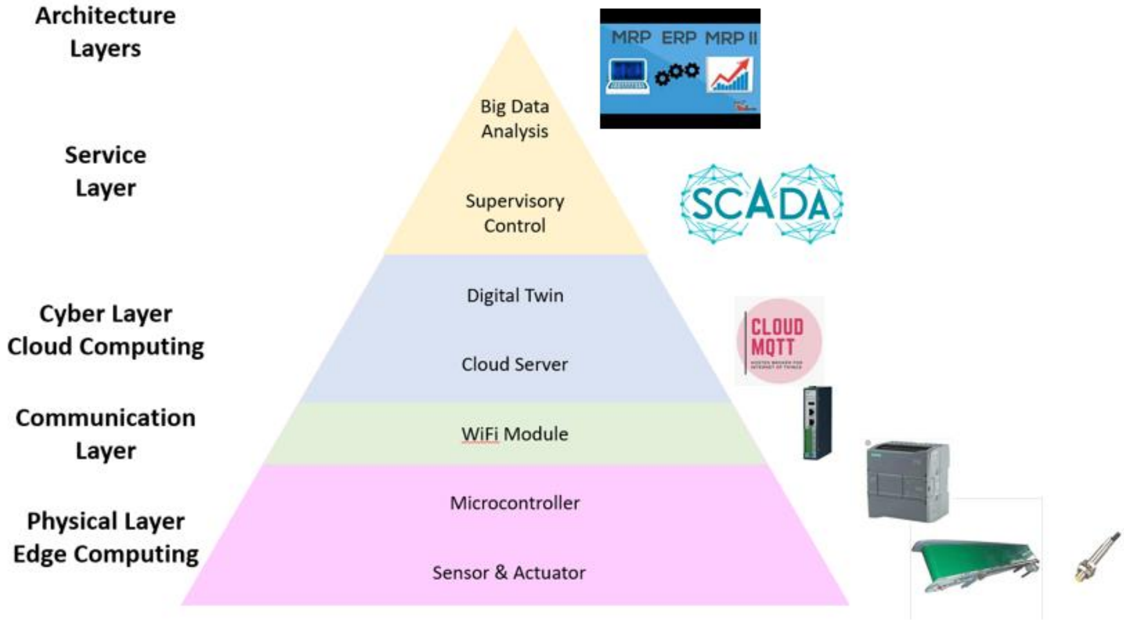


Figure 27 Implementation of the proposed four-layer CPS framework on a manufacturing system. Each image represents a specific component that was chosen in that layer.

3.3.1 Physical Layer

The physical setup is captured in Figure 28. Two magnetic proximity sensors were mounted to determine the workpiece had reached the intended positions. An electrical conveyor belt moves the workpiece to the second sensor, and then a pneumatic linear actuator ejects the workpiece. The universal industrial controller, a programmable logic controller (PLC), controls the entire system.

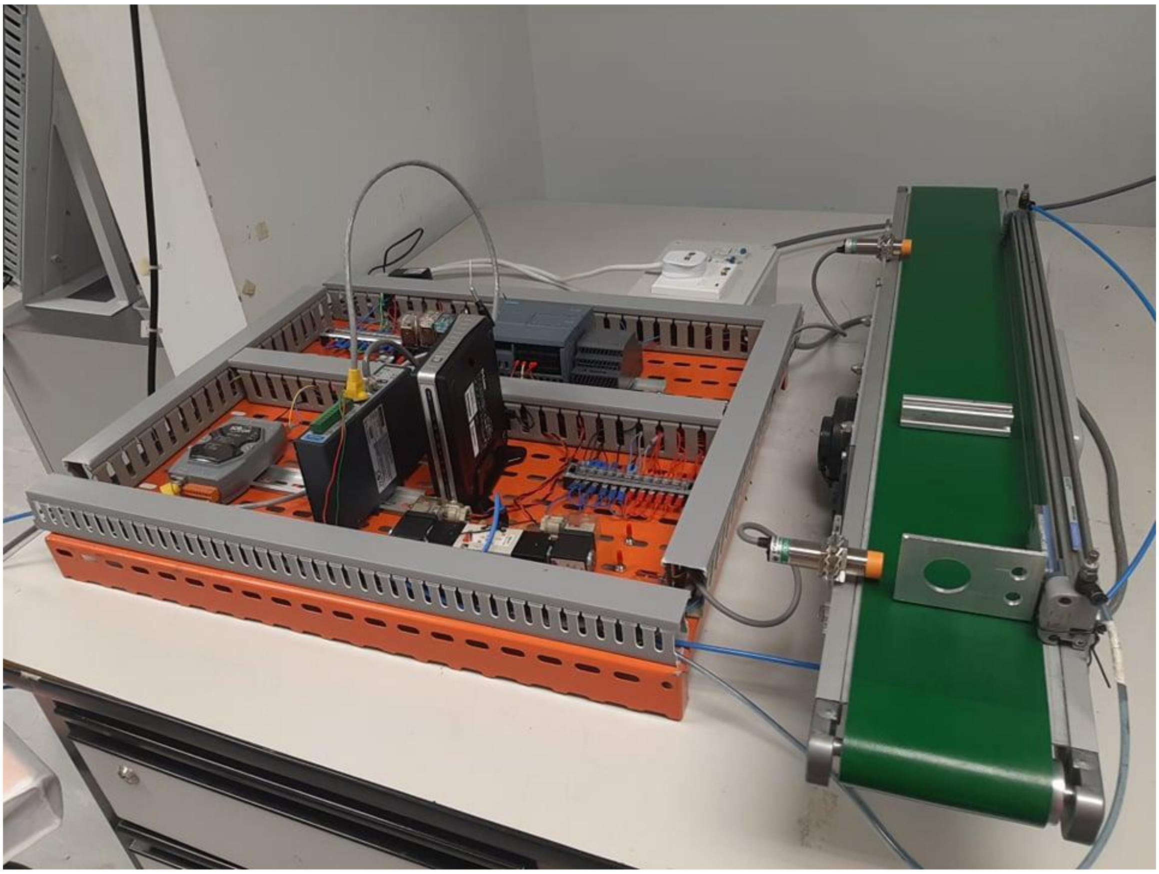


Figure 28 Manufacturing CPS built using COTS components according to the proposed CPS framework.

3.3.2 Communication Layer

Industrial components come from numerous suppliers, each using different protocols for communication. The standard industrial ethernet communication protocols are EtherNet/IP, Profinet, EtherCAT, Powerlink, and Modbus. For the industrial experiments, modules that communicate using Profinet and Modbus were chosen. Siemens S7-1200 uses Profinet, and the ICP CON ET-7051 [157] uses Modbus. These offline modules do not possess direct internet access. Therefore, a commercial gateway module from Advantech, ECU1251, introduced internet access to the manufacturing system.

ECU1251 can handle a variety of standard ethernet protocols, but the disadvantage is the limitation of ethernet ports. An additional Wi-Fi router adds ethernet ports and wireless communication between the modules. Once the networking between manufacturing modules is completed, the gateway compiles the data from the networked modules. The packaged data are sent to a cloud server and a webpage. All the setups required for the system to function are arranged in the appendix.

3.3.3 Cyber Layer

The ECU1251 came with the automated generation of a webpage for monitoring. A laptop with a wireless connection to the router was able to monitor the ECU1251 Figure 29.

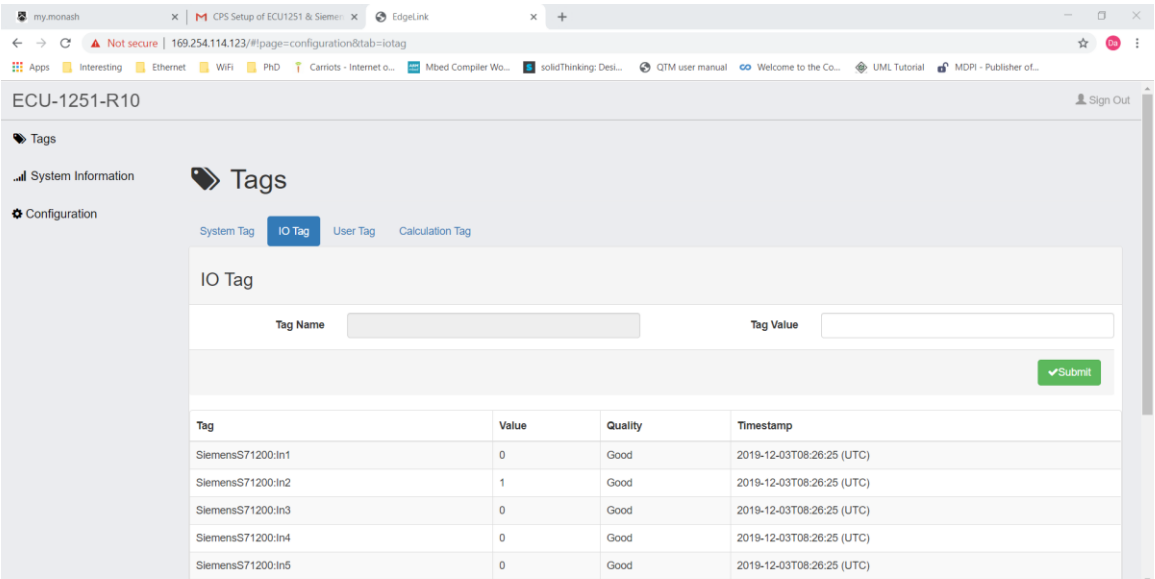


Figure 29 The Input Output (IO) Tags values associating with the inputs from the sensors and actuators.

There are a variety of commercial cloud servers available. Azure is one of the costliest, and it provides services from cloud computing to artificial intelligence. However, these services were not needed to show the functionality of the framework. Due to the specific software required to program the gateway, the packaging of the data could be customized. Instead of a higher security cloud platform such as Altair SmartCore, a simple CloudMQTT was chosen. The headers and data packet formats for CloudMQTT were much simpler and possible to program using the software.

3.4 Chapter Conclusion

This chapter discussed the steps of developing a CPS using the proposed GCPSF. Implementing of the GCPSF on an ITS and manufacturing system illustrated the practicality of the framework on different domains. In the ITS domain, the development of a near-real-time physical system intended to resolve the latency issues of CPS. The implementation of DT also signified the method of modelling a CPS. Lastly, developing the manufacturing system using COTS components and the framework showed the simplicity of implementing the GCPSF.

Chapter 4 Experimental Verification

IEEE-STD-610 defined validation as “an activity that ensures that an end product stakeholder’s true needs and expectations are met” and verification as “a test of a system to prove that it meets all its specified requirements at a particular stage of its development.” [158]. Therefore, the validation of the proposed framework is essential to show it is pragmatic for developing a CPS in different domains. The implementation of the GCPSF on an ITS and a manufacturing system aims to validate its practicality on different domain systems. ITSs typically require fast response times (in the order of 25ms), while manufacturing systems require security. This is because a vehicle travelling at 1 m/s with a 25ms program cycle time would have travelled 25mm away from the intended waypoint before applying steering commands. This error will increment at every waypoint if there is no closed-loop control. For the manufacturing CPS, the PLCs with internet access require security to ensure access to manipulate the systems is restricted. These requirements were met using the framework while aiding the development of the CPS.

Before validation, a verification process on the components within the CPS is crucial to ensure its functionality[8, 9, 35, 159, 160]. There are several verification tools such as ModelPlex [8], Simulink Design Verifier, and UPPAAL-SMC [159], and the common trial-and-error approach [9, 35]. However, these tools are meant for specific domains and software, which were unapplicable for the GCPSF. As of date, there is no benchmark methodology for verifying a general CPS framework except for the trial-and-error approach as explained in [35]. A quick look at the definition of verification - “a test of a system to prove that it meets all its specified requirements at a particular stage of its development.” – shows the reason for the absence of a standard method. The requirements for developing CPS vary a lot, and hence, no standard validating methods were found. However, for the purpose of this research, a verification methodology motivated by the Brace framework [9] and model-based design techniques [11] was customized for the GCPSF. As shown in Figure 30, this method was developed to verify the GCPSF. The developed CPS was verified by testing each layer meets the functional requirements before integrating the next layer. In the physical layer, each component and subsystem was tested to ensure the functional requirements were met. Then, the cyber layer was verified using the DT model and the experiments. Lastly, the requirements of implementing the GCPSF on both ITS and manufacturing domain was verified. After verifying every component in the CPS, the GCPSF was concluded to be a valid generic CPS framework.

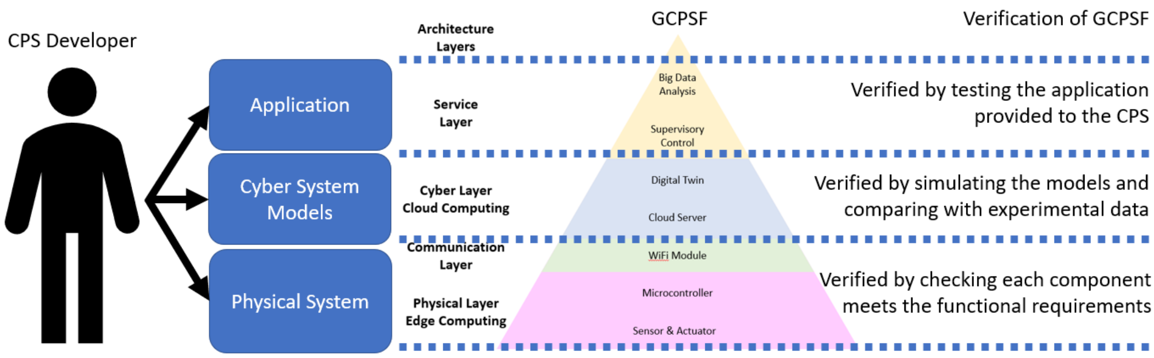


Figure 30 Customized verification method developed in this work based on Brace framework [9] and Model-based Design Techniques [11].

In this chapter, the verification of each component within the ITS and manufacturing system CPS will be discussed followed by the validation of the proposed framework. Section 4.1 discusses the process of verifying the ITS components, while section 4.2 covers the manufacturing components. Lastly, section 4.3 describes the validation of the proposed framework using the two developed CPS.

4.1 Verifying the ITS Components

After developing the CPS according to the framework, the various implemented components must be verified to ensure its functionality. The components from each layer in the GCPSF have specific requirements depending on the domain and application. Therefore, it is crucial to verify the components using experiments to ensure the requirements are met.

4.1.1 Physical Layer

The main component in the ITS domain that directly affects the performance of the physical layer is the microcontroller. This component determines the timeliness of operating the set of commands programmed. Measurements were done on the microcontroller to determine the least amount of time required to run the round-robin loop. The measurements were carried out by the ‘Timer’ function in an online compiler, Mbed.

4.1.2 Communication Layer

The Wi-Fi module in the ITS domain collected and compiled data from the microcontroller via serial communication. The most substantial latency in the CPS was in transmitting data from the Wi-Fi module to a cloud server. Experiments were carried out along with a review of the literature related to overhead latencies. The Wi-Fi module was given two equal-sized data packets to send to a cloud server using HTTP and MQTT. Both experimentations and research concluded MQTT was faster and a more suitable option for CPS applications.

4.1.3 Cyber Layer

The cyber layer verification is separated into the cloud server and DT. The cloud server section discusses the available cloud service providers and the chosen server for the ITS domain. The second section describes the experiments to verify the DT model and its predictive capability.

4.1.3.1 Cloud Server

The verification of the cloud server was carried out by sending sensor data from the vehicle to the server continuously for a set amount of time. However, most cloud service providers limit users by various means for business purposes. The cloud service providers give excellent security features while using the services, but the limitations hinder the freedom of developers to control every component of the CPS. Therefore, a cloud server installed on a remote laptop, the Eclipse Mosquitto MQTT broker, was chosen instead. In addition to having full control over every aspect of the cloud server, the storage space is only limited by the remote laptop's hard disk instead of a commercial limitation. The security of this cloud server can be enhanced by utilizing frameworks developed in [161-163] to further improve the safety of implementing a non-commercial cloud server. Moreover, the remote laptop was using a different network compared to the vehicle to introduce latencies similar to those introduced by commercial servers.

4.1.3.2 Digital Twin

The experiments were separated into verifying the DT model and verifying the prediction capabilities of the DT. DT was first developed as a kinematic model of the steering system using D-H in MATLAB. That model was verified by the experiments on the steering system using open-loop control. After verifying the DT model, it was then integrated with a vehicle dynamics model in MATLAB Simulink block diagram in Figure 26. The behavior DT model developed predicted the performance of the vehicle and sent the trajectory to the physical vehicle. The prediction capabilities were then verified by the second set of experiments on the entire vehicle to perform simple trajectories, including straight lines, circles, and Dubins path.

4.1.3.2.1 Experiments for Verifying the DT Model

After developing the DT model of the steering system, verification was required to ensure its functionality. A series of measurements and open-loop experiments were carried out to verify the DT model of the steering system.

The Ackermann geometry, an ideal steering system, was derived and compared with the results. The significance of this benchmarking was to determine the correlation between the ideal, measured, and simulated results.

The steering system was controlled using a DC servo motor on the vehicle. The microcontroller was programmed to divide the entire steering range into 11 equal positions. The steering system held each position to measure the respective steering angles on the wheels. These data were used to compare with Ackermann geometry, motion capture system, and simulation of the DT.

The second set of experiments was carried out on the steering system using a motion capturing system, Qualisys[164]. The low latency motion capturing system enabled real-time marker tracking. Two rods were attached on both sides of the wheels with markers for the motion capturing system. Then, signal inputs, step, ramp, and sine were introduced to the steering system. The corresponding response of the steering system was captured.

Lastly, the DT model of the steering system was derived and simulated using MATLAB. The benchmark, measurements, experiments, and simulated steering angles were compared.

4.1.3.2.2 Experiments for Verifying the DT Model's Prediction Capabilities

As illustrated in Figure 31, the experimental setup had six Qualisys motion tracking cameras recorded the motion of the vehicle during the experiments. The terrain is a fixed racetrack in Figure 31. The details of the racetrack's information were provided for the standard racetrack used in competitions [165]. A set of trajectories were given to the vehicle to follow for the verification of the DT's predictive capabilities. The trajectories include straight lines, circles of different radii, and Dubins paths.

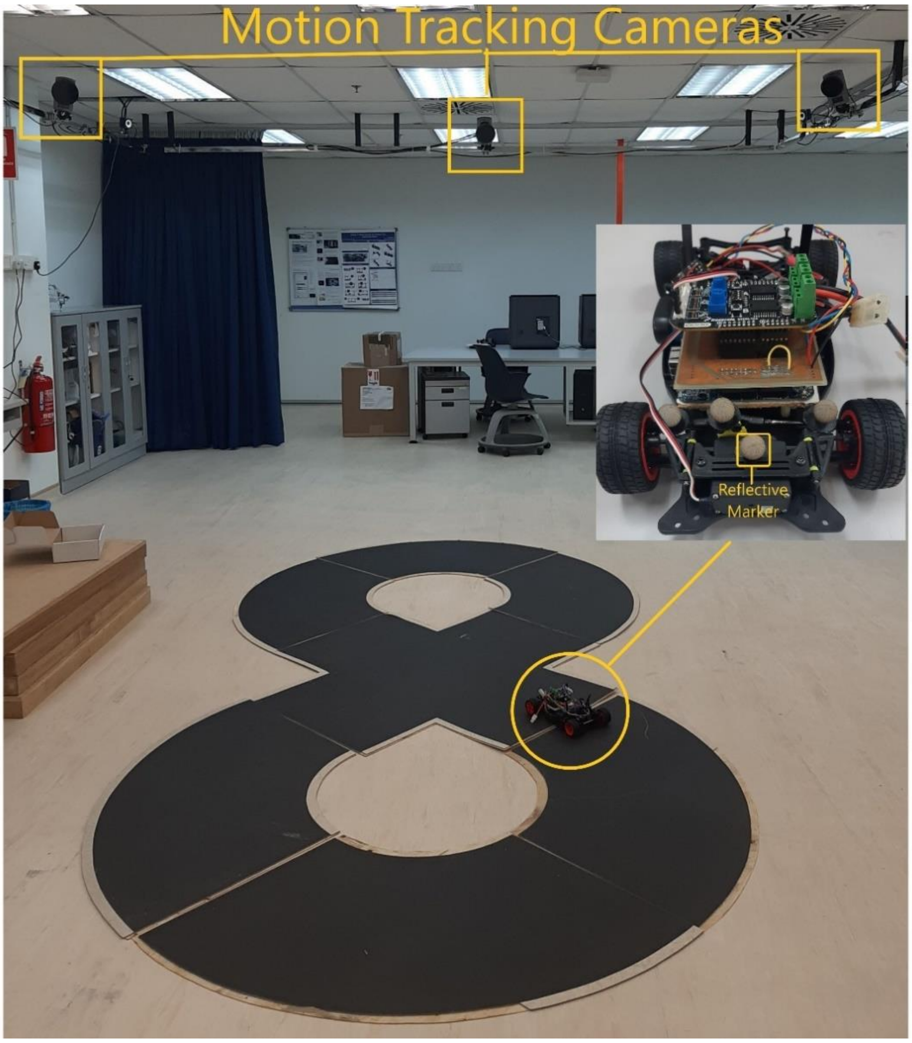


Figure 31 Experimental setup for verifying the framework for the ITS application.

4.1.4 Service Layer

The service layer in the ITS CPS was developed to provide a path planning service. Therefore, a verification of the service layer was done in MATLAB. Firstly, the verification of Dubins path was done by testing if the vehicle was able to meet all the requirements of implementing the path planning service. Dubins path requires the vehicle to maneuver at a constant speed throughout its path. The results of the straight line experiments carried out on the vehicle showed it could maintain constant speed (1.3 m/s or 40% top speed) throughout the tests. Moreover, the open-loop circle experiments concluded the vehicle was able to achieve constant turning performance at a constant speed. Therefore, the first requirement was verified to determine Dubins path is suitable.

The second requirement of implementing Dubins path is changing direction at the exact location of the waypoints from a straight line to curvature or vice versa. The microcontroller was programmed to run numerous tasks, which were to filter, fuse, packet, send the data to an external Wi-Fi module and SD card, and operate as a vehicle within the 25 ms cycle time using RTOS concepts. Since the vehicle was able to change directions at the waypoints within 25 ms, the second requirement was verified, so Dubins path is a suitable path planning method for this vehicle.

Therefore, Dubins path’s equations in [155] were then implemented in a MATLAB Simulink function block diagram (Figure 26) to calculate and send the trajectory needed to the physical vehicle. Lastly, the entire ITS setup was tested, as shown in Figure 31, after integrating the programs written in the physical, communication, cyber, and service layers.

4.2 Verifying the Manufacturing System Components

Each component used in the manufacturing CPS was obtained off-the-shelf. Therefore, the functionality of each element was reliable. However, the integration between the components posed a challenge due to the different communication protocols and the specific training courses required to operate and program each component.

4.2.1 Physical Layer

The PLC used in the manufacturing CPS was Siemens S7-1200 AC/DC/RLY. The software to program the PLC was Totally Integrated Automation (TIA) Portal 14. Various training courses were required to integrate the PLC with the implemented Wi-Fi module. Specific technical training courses were required and documented in the Appendix to allow communication between the controller and the Wi-Fi module via Profinet. The specific training courses were provided by the suppliers of the COTS components on how to effectively integrate different components from various companies using their product.

4.2.2 Communication Layer

The Wi-Fi module used in the manufacturing domain was a gateway by Advantech. The gateway required specific training to configure for retrieving data from various devices. The advantage of using the COTS gateway was its ability to integrate multiple types of ethernet and serial communication protocols. The gateway enabled the integration of various kinds of devices with communication protocols, including Modbus, Profinet, TCP/IP, serial, and other conventional components. Thus, providing a device that fills the communication layer for transmitting data from the PLC to a cloud server.

4.2.3 Cyber Layer

A cloud service provider named CloudMQTT was used in the manufacturing CPS. The internet protocol, MQTT, was chosen to compare with the ITS domain CPS. The time taken to send the data from sensor activation to the cloud server was measured. The difference between the time taken for the two CPS applications to transmit data from the physical system to a cloud server was recorded and compared.

4.3 Chapter Conclusion

This chapter described the process of verifying the CPS in both ITS and manufacturing domain. All the components within the systems were verified using experiments to ensure the functionality in each GCPSF layer.

Chapter 5 Results

The experiments in the previous chapter were intended to verify the research objectives. The research objectives are developing a generic CPS framework, building a near-real-time physical system, developing a DT, and verifying and validating the framework. Based on the framework, the results of the experiments verify the functionality of the ITS and manufacturing domain CPS's components.

In this chapter, the results of implementing the GCPSF on the ITS and manufacturing systems are described. Section 5.1 focuses on the results obtained from the ITS domain to fulfill the research objectives of developing a near-real-time physical system and a DT model of the vehicle's steering system. Section 5.2 describes the results of using COTS components in building a manufacturing CPS.

5.1 Results of Implementing GCPSF in ITS domain

5.1.1 Physical Layer

One of the objectives is to build a near-real-time physical system for the ITS domain, and the microcontroller governs the physical system's operations. Measuring the time taken to execute the round-robin loop in the microcontroller determines how close the physical system is to real-time (22 ms worst-case execution time). Upon obtaining the results, additional time was added to the program cycle time for 88% of the round-robbin loop time. Therefore, the microcontroller was programmed to run the round-robin loop at 25 ms. Within this period, the microcontroller was able to read, filter, emulate, and send data to a Wi-Fi module and SD card, as well as Proportional–Integral–Derivative (PID) control the drive and steering motors.

5.1.2 Communication Layer

Verification of the communication layer ensures that the information from the physical layer is reliably and promptly transmitted to the cyber layer. The results of implementing HTTP and MQTT internet protocol to transfer information concluded that MQTT was a faster option. A benchmarking test of sending double the size of the current data packet across HTTP and MQTT to the same server resulted in the worst-case execution time of 1350ms and 650ms, respectively, using the same microcontroller after 1000 data packets. These results match the research done to compare MQTT with other internet protocols and concluded similar results to determine the fastest option available [166, 167]. Moreover, in the 1000 data packets sent from the microcontroller, all were received on the same server for both HTTP and MQTT. Therefore, MQTT is a reliable internet protocol. However, no additional security features were developed in the GCPSF. The framework implemented the in-built security feature of username and password, and no tests were done to check the system's security. However, the implementation of QoS in MQTT gives additional reliability [168-170], whereas various research was done to improve the existing security measures MQTT provides in [161-163]. Thus, the implementation of the fast, reliable, and secure MQTT in the CPS verified the functions of the communication layer.

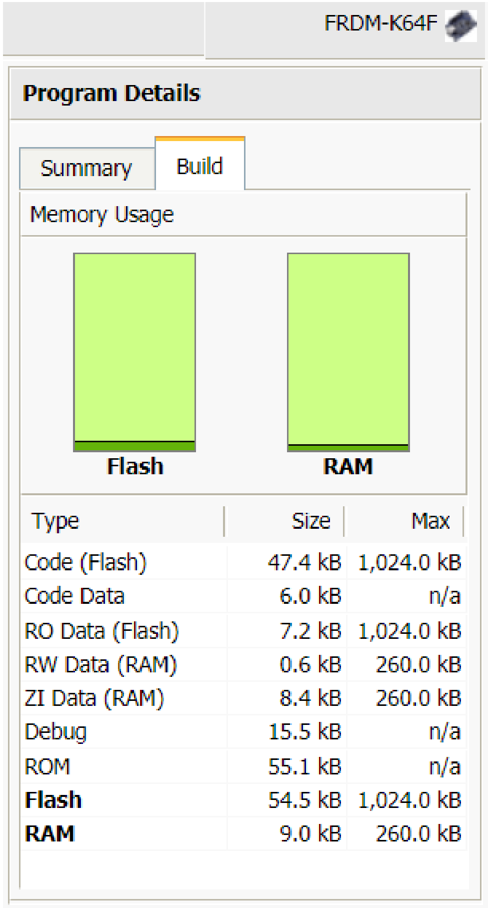
5.1.3 Cyber Layer

The results of the cyber layer are separated into the cloud server and the DT. The results of the cloud server describe the combination of cycle time required for the communications to be reliable. Then, the results of the experiments to verify the DT will be presented.

5.1.3.1 Cloud Server

The limitations of the microcontroller affected the cycle time required for the Wi-Fi module to compile and send data to a cloud server. MQTT was the internet protocol chosen to send data across the internet. Even though MQTT is a fast and lightweight protocol, latency is unavoidable. Thus, additional time was allocated for the retrieval and publishing of the data.

The rate of publishing data onto the Eclipse Mosquitto MQTT broker was set for a 500 ms cycle. This publishing rate was set because the Wi-Fi module collects data packets from the microcontroller at one data packet per 25 ms. Therefore, after collecting 20 data packets, the data was sent to the broker every 500 ms. These cycle times were tested using different combinations to accommodate the computing time required on the microcontroller and the subscriber, MATLAB. By measuring the execution time of the entire code using the microcontroller's clock, it was found that 22 ms was the worst-case execution time. Therefore, by implementing a 25 ms program cycle time, the latency is maintained. The microcontroller was required to read, filter, fuse, packet, and send the data to an external Wi-Fi module and SD card in addition to operating as a vehicle. Figure 32 showed the memory usage when the program was flashed on the 120 MHz max CPU frequency FRDM-K64F microcontroller [171]. The program was manually written in C++, and it was highly optimized using RTOS concepts. In contrast, MATLAB was slow in receiving MQTT data during the experiments using a laptop with specifications and runtime task manager shown in Figure 33 and Figure 34. [172] supported that MATLAB is slow in retrieving data using MQTT. Therefore, MATLAB was programmed to retrieve data from the MQTT broker every 500 ms, which is sufficient in the GCPFS. Thus, these experiments signified the development of real-time physical systems and latency control for a functional CPS.



Type	Size	Max
Code (Flash)	47.4 kB	1,024.0 kB
Code Data	6.0 kB	n/a
RO Data (Flash)	7.2 kB	1,024.0 kB
RW Data (RAM)	0.6 kB	260.0 kB
ZI Data (RAM)	8.4 kB	260.0 kB
Debug	15.5 kB	n/a
ROM	55.1 kB	n/a
Flash	54.5 kB	1,024.0 kB
RAM	9.0 kB	260.0 kB

Figure 32 Memory usage of FRDM-K64F microcontroller for the vehicle.

Figure 33 Windows 10 laptop running the MATLAB GUI.

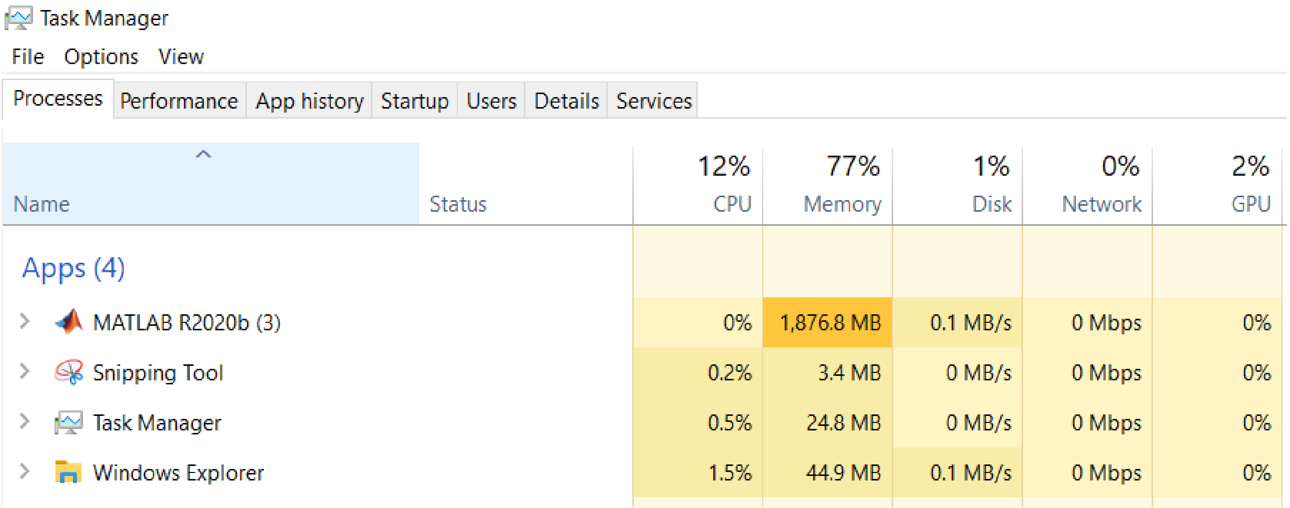


Figure 34 Task manager when running MATLAB using the laptop from Figure 32.

5.1.3.2 Digital Twin

The results in this section are related to the verification of the DT model and its prediction capabilities. The first set of experiments uses the Qualysis Motion Tracking System to capture the steering response by introducing step, ramp, and sine inputs. Figure 35 to Figure 37 showed the results of the experiments in comparison with the DT model's simulation. The discrepancy for the right steering angle is due to the unequal length of link two on both sides of the steering system and semi-flexible joints. The semi-flexible joints were not taken into account in the DT model, and together with the unequal link lengths, the discrepancies on both sides varied.

DT was used because it allows new models that were hitherto unavailable (in the current product lifecycle) to be integrated later with the existing models [134]. The different types of DT models include geometric, physical, behavior, rule, and process (Table 13). A DT model of the physical system starts as a geometrical model describing the size and features of the physical twin. Then, additional models can be developed and integrated on top to improve the functionalities of the DT. The integration of a developed behavior model using physics-based modeling methods improves prediction and control functionalities. Therefore, implementing DT in the cyber layer of GCPSF supplements the framework is both generic and functional.

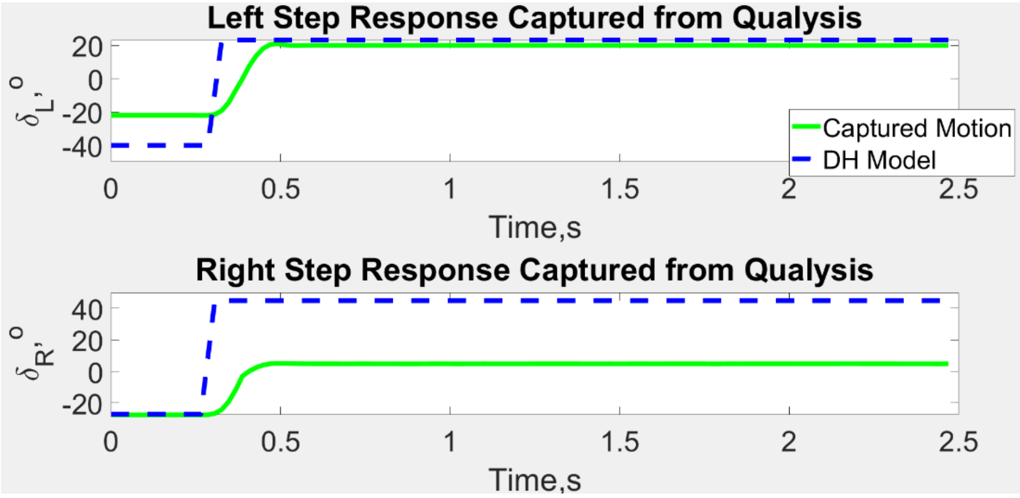


Figure 35 Step input introduced to the steering system and the response as captured by motion tracking cameras and DH simulated values.

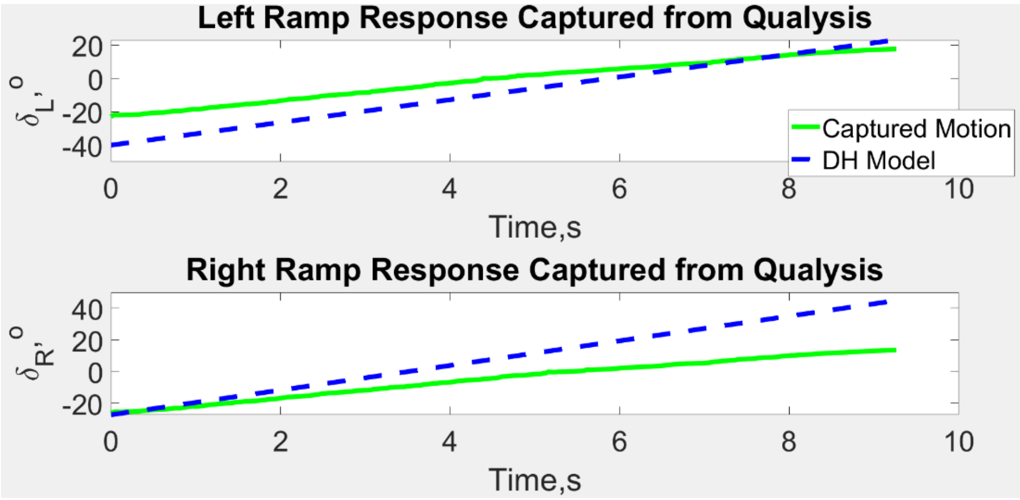


Figure 36 Ramp input introduced to the steering system and the response as captured by motion tracking cameras and DH simulated values.

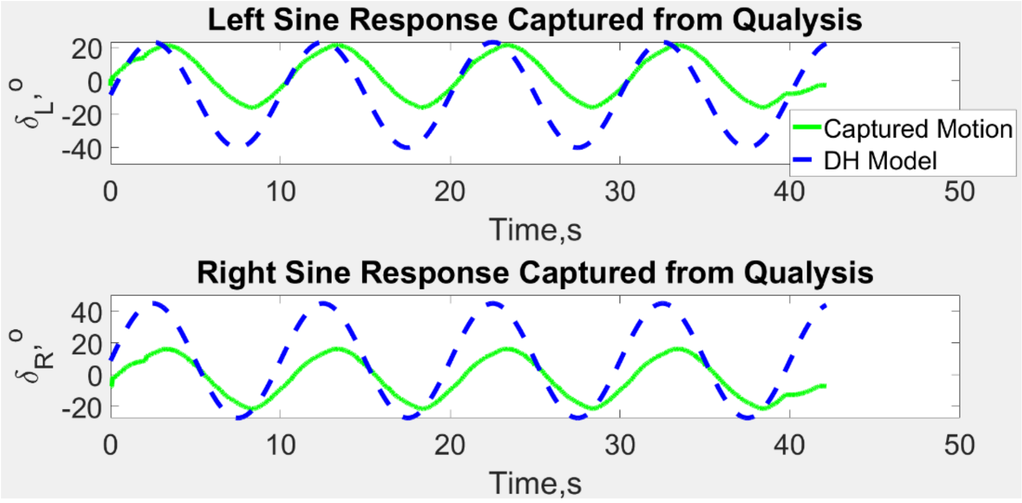


Figure 37 Sine input introduced to the steering system and the response as captured by motion tracking cameras and DH simulated values.

After deriving the DT model of the steering system in MATLAB, simulations illustrated all the positions and orientations of the links and joints. In Figure 38, the different colors represent individual links on both sides of the steering system. The results from the simulation provided the left and right steering angles. An ideal steering system benchmark, Ackermann geometry, was derived and plotted as a reference in Figure 39 to illustrate the difference between the perfect benchmark and the results of the measurements and simulation.

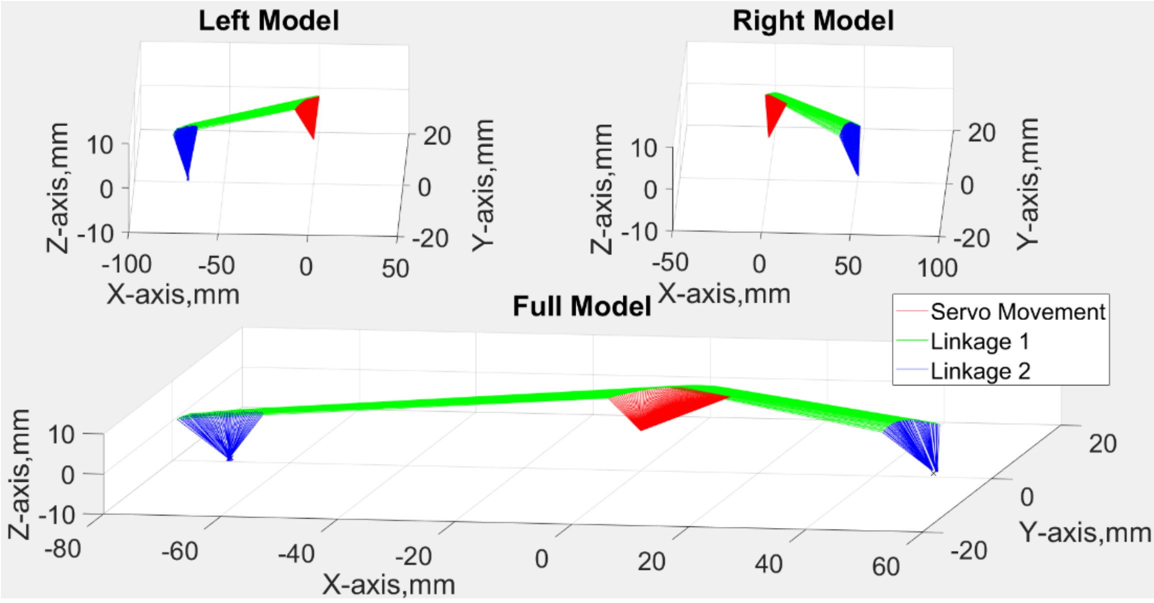


Figure 38 DT simulation of the steering system.

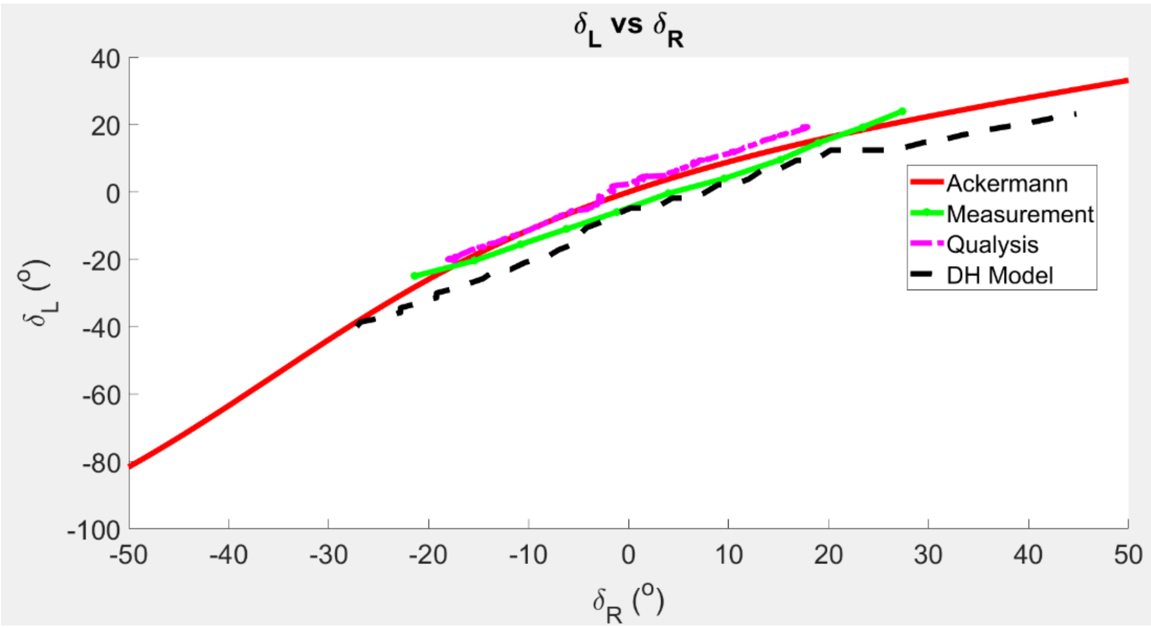


Figure 39 Comparison of the steering angles from the simulation and measurements and the benchmark.

Then, the verification of the performance prediction of DT led to experiments on the vehicle using open-loop controls. These tests verified the DT's ability to predict the performance of the whole system using a subsystem. Using a Qualysis motion tracking system, the constant speed of the vehicle was measured to be 1.3 m/s.

The open-loop experiments consisted of a straight line, varying steering angle, and Dubins path. Each test was carried out by having the vehicle travel at a constant speed. Figure 40 shows the results of the vehicle traveling in a straight line from a stationary position. A total of 30 tests were done for the straight line test. The test was repeated ten runs for different speeds, then the constant speed with the smallest error was chosen. The error was the distance between the ideal end point and the resulted end point. At 30% speed, the mean error was 495 mm; at 40% speed, the mean error was 443 mm, and at 50% speed, the mean error was 550 mm. These results revealed 40% speed had the best performance and was chosen to be implemented for the Dubins path. The experiments confirmed the DT model's prediction that the vehicle will never be able to travel in a straight line using open-loop control.

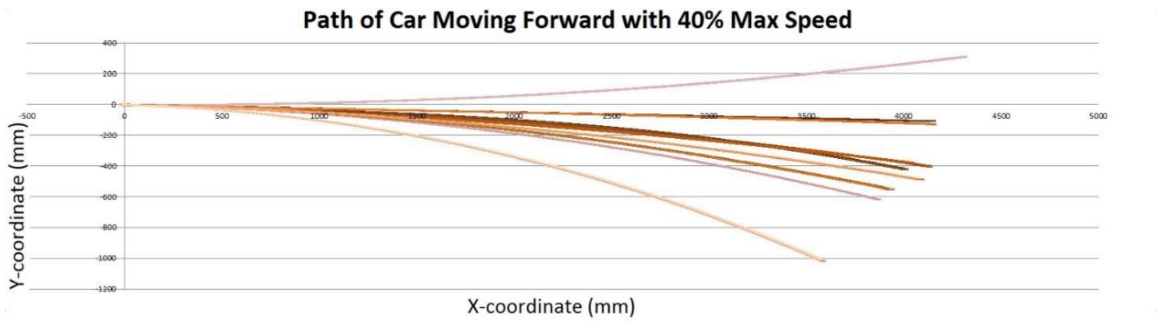


Figure 40 Straight line experiment with constant speed.

The result of varying the steering angles with constant speed is shown in Figure 41. About 70 tests were done to measure the turning performance of the vehicle in an open-loop setting. The experiments were repeated seven circles for each steering input after the vehicle's speed settled at speed constant for both left and right turns. Then the average radius of the vehicle's circle path was taken as the turning performance of the vehicle. The steering performance for the left turn was significantly better than on the right side. The turning radii were collected for open-loop control to track the Dubins path. Moreover, the experiment confirmed the prediction of the DT model that the vehicle has different steering performance on both sides.

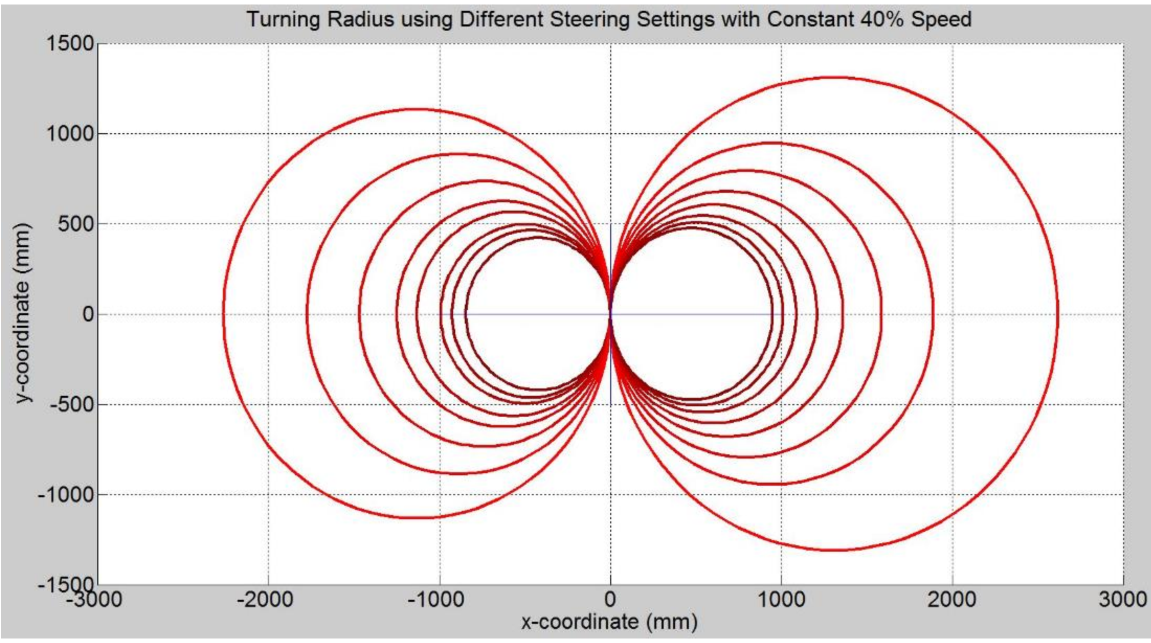


Figure 41 Constant speed with a varying steering angle.

A figure-eight shaped Dubins path was developed according to the turning performance of the vehicle. A Qualysis motion capture system captured the results shown in Figure 42. The test was repeated three times, and eight Dubins paths were continuously tracked in Figure 42 to illustrate the effects of implementing open-loop control on the vehicle. The black line indicates the common point of the path that drifts from the left to the right side of the x-axis. The drift occurs because of the unequal steering performance on the vehicle's left and right sides, as shown in Figure 41. Moreover, the steering delay shown in Figure 35 indicated there would be an error when following paths in the open-loop setting.

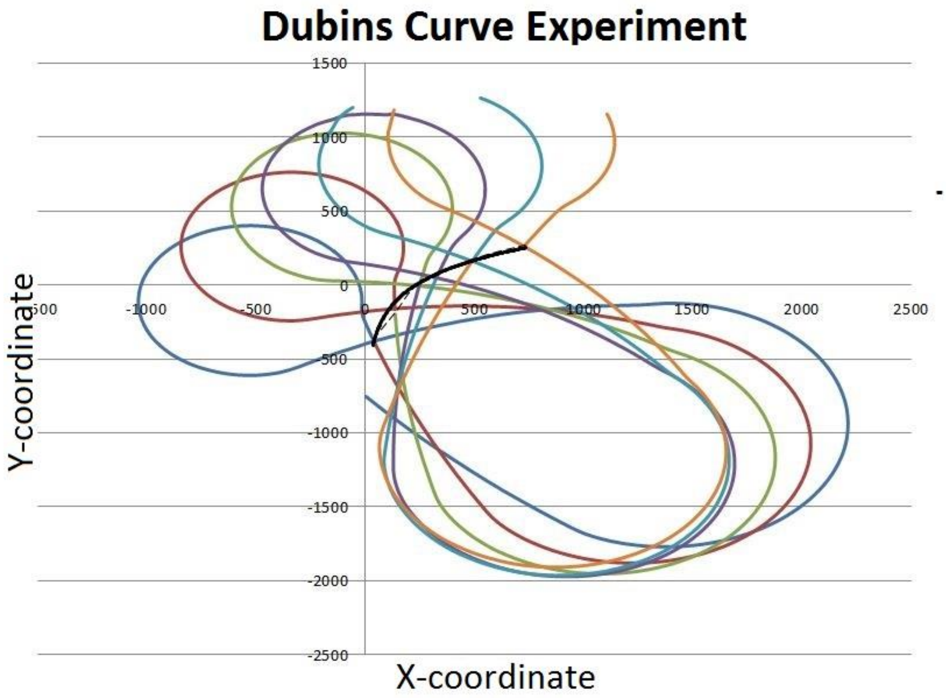


Figure 42 Dubins experiment using open-loop control on the vehicle.

The second objective was to develop a DT for a steering system in the ITS domain. Firstly, the DT model used D-H parameters as state variables to describe the behavior of the steering system. The simulation of the DT model was compared with the results from the two experiments on the steering system to verify the model. The DT simulation from MATLAB represented the entire motion of the steering system within its physical restrictions. A comparison of the DT simulation and test data was made to determine the correlation and validity. The correlation was done using the linear regression model of Ackermann to measure the linear relationship strength and direction of the three results. The R^2 values when using the ideal Ackermann steering model are measurement = 98.14%, motion capture cameras = 97.82%, and DT model = 94.94%. All these values indicated all three results individually have a high positive correlation to Ackermann. Therefore, the Ackermann steering model is an essential benchmark in predicting a vehicle's steering performance.

In order to validate the DT model of the steering system, the prediction capabilities were tested using the straight line, turning, and figure-eight shaped Dubins path open-loop experiments. In Figure 39, the DT model is shown to be below the Ackermann steering curve. This result indicated that the vehicle tends to turn to the right because the left's required steering angle was always lower. The results in Figure 40-42 were aligned with the prediction of the vehicle's performance using that steering system. Therefore, the DT model is validated to be capable of predicting the performance of the vehicle as a whole.

5.1.4 Service Layer

The GUI developed in MATLAB served as the interface to the service layer for the CPS. Figure 43 shows the outline of the GUI. The left graphs depict near-real-time data from the vehicle, whereas the right image displays the path planning by entering the initial and ending positions with the orientation. In operation, the GUI collected data from the vehicle via an MQTT broker using MQTT protocols in different locations. The MQTT broker located off site ran on a Windows laptop with a Wi-Fi connection. The plotting of the left side graphs occurred every 500 ms to show the vehicle's sensor values.

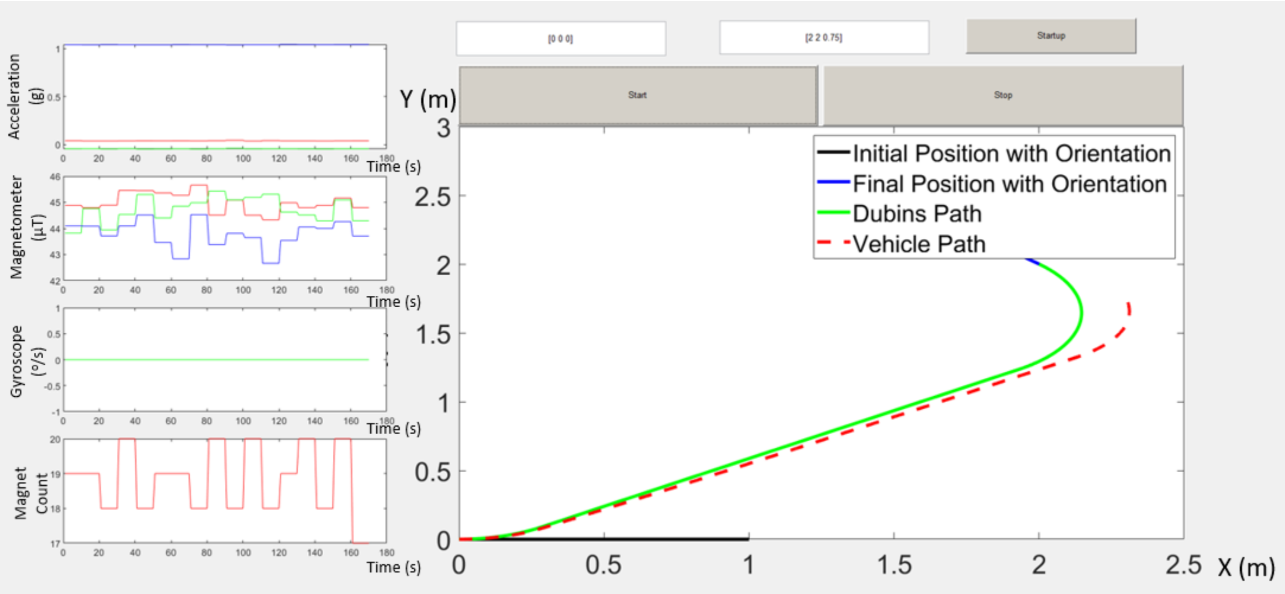


Figure 43 GUI for the CPS in the ITS application.

5.2 Results of Implementing GCPSF in Manufacturing System domain

The verification of implementing the GCPSF on a manufacturing system shows the validity of the framework in a different domain. The resulting off-the-shelf manufacturing CPS is capable of compiling data generated from the physical system and sending the data to a cloud server for analysis.

5.2.1 Physical Layer

Implementing the GCPSF on the physical system resulted in the selection of a conventional PLC for operation. A conventional PLC illustrates that any controller is suitable if the customer's requirements are met. However, the disadvantage of having a large selection is the requirement to check the compatibility with the Wi-Fi module. Therefore, Advantech's ECU1251 gateway was chosen because of its compatibility with a wide array of controllers. Moreover, the specified settings acquired from supplier training enabled communication between the PLC and the gateway. The successful data transmission verified the functionality of the physical and the communication layer. After verifying the capability of transmitting data between the layers, the inclusion of legacy systems can be done. Legacy systems typically have outputs to notify its status to the operator, and these can be used as inputs to the PLC for digitization. The digitization of the legacy systems' outputs enables data analysis for optimization. Moreover, the security of implementing a CPS on top of a legacy system is protected because only the outputs were used, and no inputs were used to manipulate the system.

5.2.2 Communication Layer

With the successful communication between the controller and the gateway using the ECU1251, the next transmission of data is to the cloud server. Using the research done on the communication layer of the ITS domain, MQTT was the internet protocol utilized. However, the time taken from sensing a change in the environment to updating the data in the cloud server is significantly larger than the ITS CPS (2.0s compared to 0.5s). The considerable latency is due to the lack of optimization in both the physical and communication layers. The direct implementation of COTS components in these layers limits the designer's ability to customize and optimize each component effectively. For example, the limited ethernet ports available in the Wi-Fi module require an additional router. The introduction of additional components slowed down the communication between layers because of the increase in communication overhead.

5.2.3 Cyber Layer

The cloud server implemented could receive data from the communication layer and verified the functionalities of this layer. However, the communication overhead caused a significant delay, ITS (0.5 s) and manufacturing system (2.0 s) in receiving data from the physical system. Therefore, signifying the requirement of developing real-time physical systems to accommodate the latency issues from the communication layer. The resulting manufacturing CPS focused on proving that a fully functional CPS is developed by following the GCPSF and integrating equivalent COTS components. The DT of a manufacturing system requires more than a behavior model. As shown in Figure 44, the behavior DT was a simplified state machine that described the behavior of the manufacturing system. However, the inclusion of geometric and physical models is essential to clearly describe a manufacturing CPS. The integration of all these models requires additional proprietary software that was unavailable at the time of research. However, the licensed software developed by Siemens [130] and Altair [123] fills this research gap. Thus, implementing the GCPSF with existing COTS components and software enables academics and researchers to quickly develop a functioning CPS.

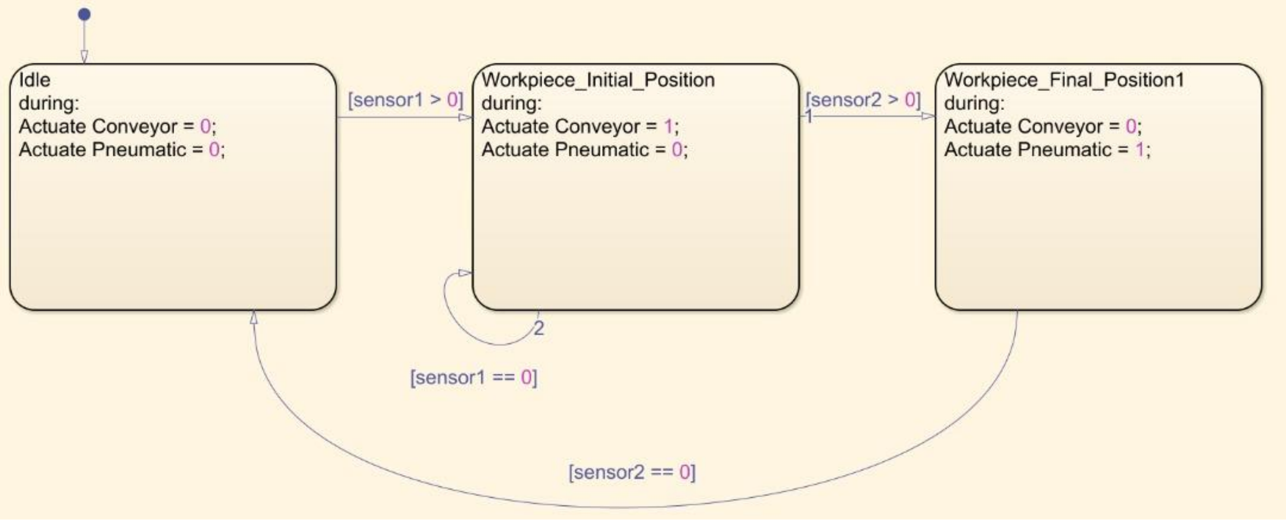


Figure 44 State Machine of Manufacturing CPS

5.3 Chapter Conclusion

This chapter showed and described the results obtained from implementing the GCPSF on both ITS and the manufacturing domain. In the ITS domain, a near-real-time physical system was achievable using a limited computing power microcontroller. Moreover, the results of using a DT model in a CPS determined the prediction and control capability. Lastly, the latency difference between the two domains was significant, which indicated the importance of developing a physical system to function in real-time.

Chapter 6 Discussion

In this chapter, the primary purpose is to discuss the results relating to the four research objectives. The significance and limitations of the results are discussed. Lastly, the results are compared with the existing literature.

6.1 Near-Real-Time Physical System

The physical system developed for the ITS domain uses a simple prototype system to implement the proposed framework. Each component was studied and integrated following the proposed framework. The physical system included in the CPS framework was comprised of the controllers and Wi-Fi modules. The most significant result of the microcontroller was the time taken to run the entire program. The cycle time of the microcontroller determined the frequency at which the Wi-Fi module could send the data to the cyber system. These were the relevant periods that determined how closely the physical systems operated in real-time.

The main findings were derived from building the physical system using RTOS concepts and memory allocation on a limited computing power microcontroller. The microcontroller was programmed to read, filter, emulate, store, and send sensor readings to an SD card and Wi-Fi module. The microcontroller was also responsible for the PID controllers on the drive and steering motors. The cycle time to run the entire program with additional time for uncertainties was 25 ms. However, in a vehicular context, the physical system was able to operate in near-real-time, thus achieving the research objective to build a physical system capable of near-real-time capabilities and describing the methodology.

The cycle time to run the entire program restricted the following components connected to it. Therefore, the microcontroller operating the physical system had to be as close to real-time as possible. This is because the current physical system setup is capped at a 25ms cycle time. If it can be further reduced to 5ms, the vehicle will detect reaching the waypoint within 5mm at 1m/s so that it can change its motion and reduce the drift (black line) shown in Figure 42.

With limited computing power, the microcontroller was able to run the entire program to filter, fuse, packet, and send the data to an external Wi-Fi module and SD card in addition to operating as a vehicle within the 25 ms cycle time using RTOS concepts. However, the cycle time could be improved by implementing a microcontroller with more exceptional computing capabilities.

6.2 Digital Twin Model

The research objectives were to develop a DT model of the CPS and verify the model and framework. The results from the steering system experiments confirmed the validity of the DT model. Measurements of the individual steering angles and ramp response captured indicated a correlation between the simulated results. Validating the DT model ensured that the inference of the model was correct.

The results obtained from measuring the individual wheel steering angles described the response of the steering system. The same response was expected from the steering system when introducing a ramp signal input. Therefore, the simulation of the DT model was compared with the two results and a benchmark Ackermann geometry. The comparison showed the DT model’s discrepancy from experiments and an ideal steering system in Figure 39.

Moreover, the step and sine signal input to the steering system both described the behavior of the steering system. The step response indicated a steering delay during a change of steering angle. The value signified the time taken for the vehicle to change from a straight line to a turning motion and vice versa. As the experimental time on the vehicle increased, the higher the steering error from the propagation of the steering delay, as displayed by the black line in Figure 42. Nevertheless, the sine response in Figure 37 matched the steering delay with a phase shift (0.2 s or 36^{o}) during the experiments. The error between simulated and measured values was due to the unequal links, steering delay, and the semi-flexible joints connecting the various links of the steering system.

Furthermore, the delay of 0.2 s in steering manoeuvres displayed by the step response strengthened the need for real-time control over the system. This is due to the friction of the tire and road surface. Therefore, the earlier the microcontroller can detect it reaches the Dubins path’s waypoint, the earlier it can execute the turning manoeuvre. The significance of these experiments was to validate the DT model of the steering system. Validating the DT model was the first step in developing a DT. After validation, the DT was used to predict the performance of the vehicle as a whole.

The DT model of the steering system was not a replica of the physical twin. The dynamics of the flexible joints were not taken into account. Thus, additional dynamic models integrating the current DT would produce better results. However, the prediction capabilities of the DT were competent in foretelling the performance of the vehicle as a whole.

6.3 Evolving the Digital Twin Model

After validating the DT model of the steering system, the results were used to infer the performance of the vehicle. The steering system is the primary source of manoeuvring the vehicle. Any discrepancies from an ideal benchmark would result in a decline in expected performance. The unequal link lengths caused the steering angle on both sides to never be uniformly straight, resulting in the inability of the vehicle to travel in a straight line in open-loop control. This vehicle was chosen to show that the DT model is capable of predicting the issues that will arise in using unequal links on the steering system in the design stage. Moreover, typical steering systems are made with equal links but with more joints to ensure both wheels' steering angles are unequal when turning [173, 174]. This is because equal steering angles of both wheels during a turn will reduce turning performance and increase the energy lost in tire slip. The steering design of using only two links in Figure 21 showed a simplified design commonly used in scaled vehicles instead of multiple links and joints implemented in an actual vehicle.

Open-loop experiments were carried out to prove the predictions of the DT model. The results from the straight-line test confirmed the predictions of the DT model. Moreover, the turning radii test found the DT model to be accurate in the forecast. The DT model resulted in the steering performance shown in Figure 39, and it showed that the left steering angle is lower as compared to ideal Ackermann steering. Therefore, the DT model predicted the vehicle's inability to move straight and perform equal turning performance on both sides.

The research objectives were to develop and verify a DT for application in the CPS framework. The experiments verified the DT model's ability to predict the performance of the vehicle as a whole. With additional models integrated into the DT, other applications could be implemented.

The figure-eight shaped path derived from two Dubins paths was designed to test the vehicle's path using open-loop controls. The results clearly showed a drift on the right side of the vehicle at every cycle. The unequal steering angles can explain the drift on the vehicle's path. Moreover, the delay in switching from a straight line to turning and vice versa caused the drift to propagate further.

Verifying and validating the DT was essential for the CPS framework. The benefits of implementing a DT in the CPS include a variety of prediction capabilities using behavior models. Geometrical models, such as kinematic skeletons, determined the relative motion of the entire system. However, the DT requires data to function. Thus, ensuring the timely communication of data between the physical and cyber systems is crucial.

6.4 Validating the Generic Cyber-Physical System Framework

The framework illustrated four layers of components that were required to build a functional CPS. The CPS built for the ITS domain displayed a functional GUI interface for trajectory planning and control of an autonomous vehicle. The manufacturing domain's CPS was built using COTS components as a benchmark. Both CPS applications were built using the same framework to prove it is a generic concept.

The manufacturing domain CPS proved that the framework is functional. The use of COTS components showed how components could be replaced with components from a different domain with similar functions. The essential detail for developing a CPS from the ground up was to utilize the framework, thereby enabling researchers to focus on the cyber systems instead of the integration process of the CPS.

The issue with COTS components was not having full control over each device. The Wi-Fi module in the ITS domain was fully customisable, while the manufacturing domain was not. The time taken to transmit data from the controller of the physical system to a cloud server was significantly slower in the manufacturing domain (ITS = 0.5 s, Manufacturing = 2.0 s). Nevertheless, the scalability of the ITS domain CPS was limited due to the tightly scheduled cycle time of each component. However, the manufacturing domain only required minor changes in the cyber layer and the addition/duplication of COTS components in the physical layer. Additionally, the cost of developing the ITS domain CPS was significantly lower when compared to the COTS CPS. This is because the manufacturing COTS components are typically within the USD’000 range, while the ITS domain requires smaller components within the USD’00 range. Moreover, most of the components used in the ITS CPS are open-sourced hardware and software. Therefore, the cost for ITS CPS is significantly lower because there is no cost of Intellectual Property and proprietary overhead of vendors. Quotations and price information were not available, as a general reference, the cost of developing a CPS using open-source hardware and software is cheaper.

The limitations of the ITS application GUI originated from the microcontroller. The cycle time constraint of the microcontroller being 25ms and the time taken to publish the data onto the MQTT broker determined the frequency of retrieving data from the MQTT broker. Therefore, increasing the real-time performance of the physical system improved the response time of the CPS. Moreover, the ITS domain CPS required a significant amount of time to program and test the microcontroller to determine the minimum cycle time. Next, every component after the microcontroller had to be programmed according to the looping frequency. However, the COTS CPS did not take latency management into account. Specific technical training courses were required to set up each of the components individually and integrate the CPS. Due to the various methods and choices of suppliers, the learning curve to integrate a CPS using these components was challenging.

Lastly, the CPS framework proved to be a generic concept in the development of the systems. The framework described the required components in each layer and allowed the engineers to determine a suitable replacement for individual applications. However, the performance of each CPS was determined by the response time of the physical systems. Therefore, the framework in the ITS domains provided an outline for various applications and focused research on cyber systems.

6.5 Chapter Conclusion

This chapter explained the meaning of the results obtained, the key challenges, and solutions for developing a CPS in different domains using the GCPSF.

Table 17 described how the gaps in literature were addressed.

Table 17 Justification of research contributions with the corresponding gap in literature review.

#	Gap in the Literature Review	Research Contribution	Justification
1	Research literatures contain various architectures, but no generic framework for CPS was found.	Proposed a four-layer generic CPS framework (GCPSF) suitable for CPS designers and academics.	The GCPSF was verified for two different domains - ITS and manufacturing domain. By replacing domain-specific modules and components in each layer, it was seen that the GCPSF can be quickly ported for other domains like medical, energy, and civil; hence it is a generic framework. Additional details are included below.
2	A robust, reliable, and safe near-real-time physical and cyber subsystems are required for GCPSF.	Proposed building a DT framework for integration of various models (discrete and continuous) and data.	Experimental verification showed that by implementing RTOS concepts, the subsystems work within the required latency requirements (ITS - 0.5 s; manufacturing 2.0 s)

3	A cyber system that is capable of integrating various cross-discipline models and data is required for the GCPSF.	Proposed building a DT framework for integration of various models and data.	Developed kinematic and dynamic models and integrated them successfully using the developed framework. First, a kinematic model for the steering system (using a discrete model in MATLAB) and a dynamic model for estimating vehicle's behavior (using a continuous model in Simulink) was integrated using the framework to work as a single CPS system.
4	Requirements for the designing and development of CPS vary widely. Hence, it is tedious and challenging to verify and validate the CPS, but no standard method also exists; most researchers use experimentation to verify and validate.	Customized a verification method based on model-based design and Brace framework. This customised method was used to verify GCPS requirements.	The varied requirements of two domains were used to verify the GCPSF. Verification was done by checking all layers of the GCPSF. The validation methodology was verified by applying it on two very different domains.

The results of implementing the GCPSF on both ITS and manufacturing domains showed it is a pragmatic general CPS framework. The GCPSF is modular as it allows the decomposition of the system into functional modules hierarchically into four layers. It then aggregates modules and components in each layer to realize a complex CPS.

For example, there are aspects of the physical layer that are domain-specific, while others are common across domains. In the physical layer, sensors and actuators are present - some general and some domain-specific. However, the concept of data acquisition and sensor fusion are similar across domains. The sensors acquire data at a particular frequency and fuse readings from multiple sources for data analysis. In the manufacturing domain, a camera is used for fault detection. In the ITS domain, a camera is used for path tracking. Therefore, by having layer-specific modules, it is easy to integrate systems across domains.

Although different domains use different types of energy like electrical energy for ITS and pneumatics or hydraulics for the manufacturing domain, they all produce data that can be digitized and sent to the cyber layer. In the cyber layer, all data received from the physical layer are digitized and compiled into a single format (SQL or JSON). Therefore, the cyber layer of CPS is similar in each domain except for the DT models. The same cloud storage and databases can be used across domains. On the supervisory layer, machine learning can be used to provide better routes for the vehicle or give other functionality in different domains. Thus, this shows that the GCPSF is a generic framework that is suitable across domains.

The GCPSF is interoperable mainly because the gateway architecture and the expanse of the communication layer can cater to a wide range of communication protocols to accommodate data collection from different interfaces. Reconfiguration of hardware is mostly done at the gateway layer, simplifying the development. The modularity and interoperability of the framework allow scaling the system to a bigger and more complex CPS by duplicating or adding new physical systems, computational capability with minor modifications, and reconfiguration in the cyber system. Lastly, we demonstrate this by using a COTS gateway, ECU1251, which allows ethernet, RS-232, and RS-485 communication protocols. Most legacy systems have GPIOs and communication ports based on RS-232 and RS-485, which can be easily integrated with the GCPSF using the gateway. Therefore, legacy systems can be connected to the current experimental setup to gather data for purposes like calculating Overall Equipment Effectiveness. In addition, equivalent COTS Wi-Fi modules can be used to replace the ECU1251 for the same function. Thus, the GCPSF filled the research gap where there was no step-by-step generic framework for designers and system integrators to develop a CPS from the ground up or on top of existing systems.

The implementation of RTOS concepts on the physical system enabled communication with the cyber system to work within the permissible latency. The microcontroller was programmed to read, filter, emulate, store, and send sensor readings to an SD card and Wi-Fi module. In addition, it was also responsible for the PID controllers on the drive and steering motors. The cycle time to run the entire program with additional time for uncertainties was 25 ms. Therefore, the physical system was able to operate in near-real-time, thus achieving the research objective to build a physical system capable of near-real-time capabilities.

The DT models for both ITS and manufacturing CPS were developed in MATLAB as behavior models. The prediction capabilities of the DT model in the ITS domain showed the importance of having a model in the design stage to identify problems that will arise. Besides, the DT model in the manufacturing domain showed the importance of integrating different types of DT models to have a holistic view of the entire system.

The verification and validation were done in a customized method on the GCPSF. This is because verifying and validating every component of the CPS is tedious, challenging, and no standard method exists. Therefore, the experiments carried out were used to verify each component and system implemented in the CPS. Then the GCPSF was validated to be a generic framework because it met the requirements of implementing it on two different domains. The stated requirements for ITS was speed, while the requirements for the manufacturing domain was security.

Chapter 7 Conclusion

This thesis addresses the need for developing a generic framework to engineer CPS. The primary focus was to synthesize a general CPS framework that we call GCPSF, based on the requirements to facilitate the rapid development of multi-domain CPS applications. The proposed GCPSF is a four-layered, function-based architecture. This architecture allows seamless aggregation of system requirements. Existing frameworks largely disregard important engineering requirements like latency and real-time constraints, both supported by the proposed GCPSF. Two prototype implementations verify the GCPSF in two diverse research domains - Intelligent Transportation Systems and Smart Manufacturing.

The four-layer GCPSF allows the segregation of the system requirements for developing a CPS from the ground up based on its functions. The proposed framework integrates the advantages of the previous CPS architectures for designer and system integrators. The proposed framework provides a step-by-step methodology for engineering CPS in any domain. The simple methodology enables designers and integrators from different backgrounds to quickly develop a working CPS.

In order to provide the GCPSF with a robust, reliable, and safe physical system, a near-real-time system was developed using RTOS concepts. Directly implementing an RTOS into the complex physical systems would have resulted in an unreliable performance. Therefore, the decoupled approach of programming the microcontroller with limited computing power for the ITS application using a self-written program with RTOS concepts were adapted. The self-written program was able to operate the required functions of the vehicles and communicate with a cloud server. Thus, signifying the importance of latency management from the physical system to ensure timely communication of data. For the manufacturing CPS, using COTS components limited the programmability of the controllers. Therefore, the CPS was built on top of the physical system to communicate data from controllers to a cloud server showing the implementation of the framework on an ‘existing’ system.

Diverse fields of research have different software and modelling techniques. DT modelling provides a guideline for integrating various models into the proposed framework. The integration of geometric and behavior models in the ITS CPS predicted the trajectory tracking performance of the vehicle. On the other hand, the manufacturing CPS had significant latency issues due to the various communication protocols of COTS hardware which are needed to be decoupled functionally at the nodes before integrating it with a DT model.

Lastly, the general nature of the proposed four-layer CPS framework was verified and validated for the domains of ITS and manufacturing. The functionality of each layer of the proposed framework was verified in both CPS applications. The required functions are carried out by the corresponding layers resulting in a scalable framework. For example, as the service layer is modular, the functional complexity required by future CPS applications can be met quickly through the addition of AI models.

For future work, the ITS CPS can be further improved by adding unmanned aerial vehicles and implementing Dubins path for three-dimensional trajectory. The addition of unmanned aerial vehicles illustrates the framework to be generic that can add new physical systems with some changes in the cyber system. Moreover, the service layer of the GCPSF can be enhanced by the addition of artificial intelligence for pattern recognition for more efficient control.

The proposed four-layer GCPSF can be implemented in different domains such as the electrical grid, agriculture, healthcare, and home safety. System integrators and academics alike can implement this GCPSF to develop a functioning CPS quickly. Every layer of the GCPSF can be built using equivalent COTS components for easy development. Thus, the implementation of the GCPSF on various domains will show it is a generic framework.

References

[1] B. G. Jeppesen. "Realizing the Fourth Industrial Revolution." https://mjolner.dk/2015/01/14/realizing-the-fourth-industrial-revolution/ (accessed April 28, 2020).

[2] H. Gill, "NSF perspective and status on cyber-physical systems," *Austin. Internet: http://varma. ece. cmu. edu/CPS/Presentations/gill. pdf [zuletzt aufgesucht am 1.04. 2015],* 2006.

[3] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for securing cyber physical systems," in *Workshop on future directions in cyber-physical systems security*, 2009, vol. 5, no. 1.

[4] K.-J. Park, R. Zheng, and X. Liu, "Cyber-physical systems: Milestones and research challenges," 2012.

[5] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008: IEEE, pp. 363-369.

[6] X. Wang *et al.*, "Secure rtos architecture for building automation," in *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, 2015, pp. 79-90.

[7] D. Calvaresi, M. Marinoni, A. Sturm, M. Schumacher, and G. Buttazzo, "The challenge of real-time multi-agent systems for enabling IoT and CPS," in *Proceedings of the international conference on web intelligence*, 2017, pp. 356-364.

[8] S. Mitsch and A. Platzer, "ModelPlex: Verified runtime validation of verified cyber-physical system models," *Formal Methods in System Design,* vol. 49, no. 1-2, pp. 33-74, 2016.

[9] X. Zheng and C. Julien, "Verification and validation in cyber physical systems: research challenges and a way forward," in *2015 IEEE/ACM 1st International Workshop on Software Engineering for Smart Cyber-Physical Systems*, 2015: IEEE, pp. 15-18.

[10] X. Zheng, C. Julien, M. Kim, and S. Khurshid, "Perceptions on the state of the art in verification and validation in cyber-physical systems," *IEEE Systems Journal,* vol. 11, no. 4, pp. 2614-2627, 2015.

[11] J. C. Jensen, D. H. Chang, and E. A. Lee, "A model-based design methodology for cyber-physical systems," in *2011 7th International Wireless Communications and Mobile Computing Conference*, 2011: IEEE, pp. 1666-1671.

[12] J. M. Bradley and E. M. Atkins, "Optimization and control of cyber-physical vehicle systems," *Sensors,* vol. 15, no. 9, pp. 23020-23049, 2015.

[13] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing letters,* vol. 3, pp. 18-23, 2015.

[14] K. Schweichhart, "Reference architectural model industrie 4.0 (rami 4.0)," *An Introduction. Available online: https://www. plattform-i40. de I,* vol. 40, 2016.

[15] S.-W. Lin *et al.*, "Industrial internet reference architecture," *Industrial Internet Consortium (IIC), Tech. Rep,* 2015.

[16] F. Tao, M. Zhang, and A. Y. C. Nee, *Digital twin driven smart manufacturing*. Academic Press, 2019.

[17] H. Gill, "From vision to reality: cyber-physical systems," in *HCSS national workshop on new research directions for high confidence transportation CPS: automotive, aviation, and rail*, 2008.

[18] R. Mitchell and R. Chen, "Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems," *IEEE Transactions on Dependable and Secure Computing,* vol. 12, no. 1, pp. 16-30, 2014.
[19] I. Lee *et al.*, "Challenges and research directions in medical cyber–physical systems," *Proceedings of the IEEE,* vol. 100, no. 1, pp. 75-90, 2011.
[20] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing,* vol. 5, no. 1, pp. 108-119, 2015.
[21] O. Kocabas, T. Soyata, and M. K. Aktas, "Emerging security mechanisms for medical cyber physical systems," *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 13, no. 3, pp. 401-416, 2016.
[22] I. Lee and O. Sokolsky, "Medical cyber physical systems," in *Design automation conference*, 2010: IEEE, pp. 743-748.
[23] E. M. Atkins and J. M. Bradley, "Aerospace cyber-physical systems education," in *AIAA Infotech@ Aerospace (I@ A) Conference*, 2013, p. 4809.
[24] D. Winter and B. P. Works, "Cyber physical systems-an aerospace industry perspective," *Boeing Management Company, Seattle, WA, USA,* 2008.
[25] L. Zhang, "Multi-dimensional Analysis and Design Method for Aerospace Cyber-physical Systems," in *2013 12th International Symposium on Distributed Computing and Applications to Business, Engineering & Science*, 2013: IEEE, pp. 197-201.
[26] L. Zhang, "Multi-view approach for modeling aerospace cyber-physical systems," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013: IEEE, pp. 1319-1324.
[27] J. M. Bradley, "Toward Co-Design of Autonomous Aerospace Cyber-Physical Systems," 2014.
[28] A. Wasicek, P. Derler, and E. A. Lee, "Aspect-oriented modeling of attacks in automotive cyber-physical systems," in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2014: IEEE, pp. 1-6.
[29] S. Chakraborty, M. A. Al Faruque, W. Chang, D. Goswami, M. Wolf, and Q. Zhu, "Automotive cyber–physical systems: A tutorial introduction," *IEEE Design & Test,* vol. 33, no. 4, pp. 92-108, 2016.
[30] D. Goswami *et al.*, "Challenges in automotive cyber-physical systems design," in *2012 International Conference on Embedded Computer Systems (SAMOS)*, 2012: IEEE, pp. 346-354.
[31] Z. Zhang, E. Eyisi, X. Koutsoukos, J. Porter, G. Karsai, and J. Sztipanovits, "A co-simulation framework for design of time-triggered automotive cyber physical systems," *Simulation modelling practice and theory,* vol. 43, pp. 16-33, 2014.
[32] J. Wan, A. Canedo, and M. A. Al Faruque, "Functional model-based design methodology for automotive cyber-physical systems," *IEEE Systems Journal,* vol. 11, no. 4, pp. 2028-2039, 2015.
[33] S. Karnouskos, "Cyber-physical systems in the smartgrid," in *2011 9th IEEE International Conference on Industrial Informatics*, 2011: IEEE, pp. 20-23.
[34] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet of Things Journal,* vol. 4, no. 6, pp. 1802-1831, 2017.
[35] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Design Automation Conference*, 2010: IEEE, pp. 731-736.
[36] C. Singh and A. Sprintson, "Reliability assurance of cyber-physical power systems," in *IEEE PES General Meeting*, 2010: IEEE, pp. 1-6.
[37] X. Yu and Y. Xue, "Smart grids: A cyber–physical systems perspective," *Proceedings of the IEEE,* vol. 104, no. 5, pp. 1058-1070, 2016.

[38] R. Taormina, S. Galelli, N. O. Tippenhauer, E. Salomons, and A. Ostfeld, "Characterizing cyber-physical attacks on water distribution systems," *Journal of Water Resources Planning and Management,* vol. 143, no. 5, p. 04017009, 2017.

[39] G. Hackmann, W. Guo, G. Yan, Z. Sun, C. Lu, and S. Dyke, "Cyber-physical codesign of distributed structural health monitoring with wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 25, no. 1, pp. 63-72, 2013.

[40] H.-M. Huang, T. Tidwell, C. Gill, C. Lu, X. Gao, and S. Dyke, "Cyber-physical systems for real-time hybrid structural testing: a case study," in *Proceedings of the 1st ACM/IEEE international conference on cyber-physical systems*, 2010, pp. 69-78.

[41] D. Legatiuk, K. Dragos, and K. Smarsly, "Modeling and evaluation of cyber-physical systems in civil engineering," *PAMM,* vol. 17, no. 1, pp. 807-808, 2017.

[42] T. Tidwell, X. Gao, H.-M. Huang, C. Lu, S. Dyke, and C. Gill, "Towards configurable real-time hybrid structural testing: a cyber-physical system approach," in *2009 IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, 2009: IEEE, pp. 37-44.

[43] R. F. Babiceanu and R. Seker, "Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook," *Computers in Industry,* vol. 81, pp. 128-137, 2016.

[44] C. Yu, X. Xu, and Y. Lu, "Computer-integrated manufacturing, cyber-physical systems and cloud manufacturing–concepts and relationships," *Manufacturing letters,* vol. 6, pp. 5-9, 2015.

[45] L. Wang, M. Törngren, and M. Onori, "Current status and advancement of cyber-physical systems in manufacturing," *Journal of Manufacturing Systems,* vol. 37, pp. 517-527, 2015.

[46] L. Monostori *et al.*, "Cyber-physical systems in manufacturing," *Cirp Annals,* vol. 65, no. 2, pp. 621-641, 2016.

[47] L. Zhang, "Aspect-oriented approach to modeling railway cyber physical systems," in *2013 12th International Symposium on Distributed Computing and Applications to Business, Engineering & Science*, 2013: IEEE, pp. 29-33.

[48] S. Marrone, R. J. Rodríguez, R. Nardone, F. Flammini, and V. Vittorini, "On synergies of cyber and physical security modelling in vulnerability assessment of railway systems," *Computers & Electrical Engineering,* vol. 47, pp. 275-285, 2015.

[49] B. Chen *et al.*, "Security analysis of urban railway systems: the need for a cyber-physical perspective," in *International Conference on Computer Safety, Reliability, and Security*, 2014: Springer, pp. 277-290.

[50] D. Basile, F. Di Giandomenico, and S. Gnesi, "Statistical model checking of an energy-saving cyber-physical system in the railway domain," in *Proceedings of the Symposium on Applied Computing*, 2017, pp. 1356-1363.

[51] L. Bu *et al.*, "Toward online hybrid systems model checking of cyber-physical systems' time-bounded short-run behavior," *ACM SIGBED Review,* vol. 8, no. 2, pp. 7-10, 2011.

[52] J. E. Kim and D. Mosse, "Generic framework for design, modeling and simulation of cyber physical systems," *ACM SIGBED Review,* vol. 5, no. 1, pp. 1-2, 2008.

[53] X. Hu, T. H. Chu, H. C. Chan, and V. C. Leung, "Vita: A crowdsensing-oriented mobile cyber-physical system," *IEEE Transactions on Emerging Topics in Computing,* vol. 1, no. 1, pp. 148-165, 2013.

[54] J. Sztipanovits *et al.*, "Toward a science of cyber–physical system integration," *Proceedings of the IEEE,* vol. 100, no. 1, pp. 29-44, 2011.

[55] C. Talcott, "Cyber-physical systems and events," in *Software-Intensive Systems and New Computing Paradigms*: Springer, 2008, pp. 101-115.

[56] M. Kim, M.-O. Stehr, J. Kim, and S. Ha, "An application framework for loosely coupled networked cyber-physical systems," in *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2010: IEEE, pp. 144-153.
[57] J. Lin, S. Sedigh, and A. Miller, "A general framework for quantitative modeling of dependability in cyber-physical systems: a proposal for doctoral research," in *2009 33rd Annual IEEE International Computer Software and Applications Conference*, 2009, vol. 1: IEEE, pp. 668-671.
[58] T. Karhela, A. Villberg, and H. Niemistö, "Open ontology-based integration platform for modeling and simulation in engineering," *International Journal of Modeling, Simulation, and Scientific Computing,* vol. 3, no. 02, p. 1250004, 2012.
[59] S. Schütte, S. Scherfke, and M. Tröschel, "Mosaik: A framework for modular simulation of active components in Smart Grids," in *2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS)*, 2011: IEEE, pp. 55-60.
[60] P. Palensky, E. Widl, and A. Elsheikh, "Simulating cyber-physical energy systems: Challenges, tools and methods," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 44, no. 3, pp. 318-326, 2013.
[61] J. Eker *et al.*, "Taming heterogeneity-the Ptolemy approach," *Proceedings of the IEEE,* vol. 91, no. 1, pp. 127-144, 2003.
[62] C. Kim, M. Sun, S. Mohan, H. Yun, L. Sha, and T. F. Abdelzaher, "A framework for the safe interoperability of medical devices in the presence of network failures," in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, 2010, pp. 149-158.
[63] J. Zalewski, S. Drager, W. McKeever, and A. J. Kornecki, "Threat modeling for security assessment in cyberphysical systems," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, 2013, pp. 1-4.
[64] D. Kundur, X. Feng, S. Liu, T. Zourntos, and K. L. Butler-Purry, "Towards a framework for cyber attack impact analysis of the electric smart grid," in *2010 First IEEE International Conference on Smart Grid Communications*, 2010: IEEE, pp. 244-249.
[65] A. Petrovski, P. Rattadilok, and S. Petrovski, "Designing a context-aware cyber physical system for detecting security threats in motor vehicles," in *Proceedings of the 8th International Conference on Security of Information and Networks*, 2015, pp. 267-270.
[66] T. L. Crenshaw, E. Gunter, C. L. Robinson, L. Sha, and P. Kumar, "The simplex reference model: Limiting fault-propagation due to unreliable components in cyber-physical system architectures," in *28th IEEE International Real-Time Systems Symposium (RTSS 2007)*, 2007: IEEE, pp. 400-412.
[67] L. Wu and G. Kaiser, "FARE: A framework for benchmarking reliability of cyber-physical systems," in *2013 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 2013: IEEE, pp. 1-6.
[68] J. Meseguer and C. Talcott, "Semantic models for distributed object reflection," in *European Conference on Object-Oriented Programming*, 2002: Springer, pp. 1-36.
[69] C. L. Talcott, "Coordination models based on a formal model of distributed object reflection," *Electronic Notes in Theoretical Computer Science,* vol. 150, no. 1, pp. 143-157, 2006.
[70] L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia Cirp,* vol. 17, pp. 9-13, 2014.
[71] E. M. Atkins, "Cyber-physical aerospace: Challenges and future directions in transportation and exploration systems," in *Proceedings of the 2006 National Science Foundation Workshop On Cyber-Physical Systems*, 2006.

[72] H. Yoo and T. Shon, "Challenges and research directions for heterogeneous cyber–physical system based on IEC 61850: Vulnerabilities, security requirements, and security architecture," *Future generation computer systems,* vol. 61, pp. 128-136, 2016.
[73] E. K. Wang, Y. Ye, X. Xu, S.-M. Yiu, L. C. K. Hui, and K.-P. Chow, "Security issues and challenges for cyber physical system," in *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, 2010: IEEE, pp. 733-738.
[74] K. Wan, D. Hughes, K. L. Man, T. Krilavicius, and S. Zou, "Investigation on composition mechanisms for cyber physical systems," *International Journal of Design, Analysis and Tools for Integrated Circuits and Systems,* vol. 2, no. 1, p. 30, 2011.
[75] S. H. Ahmed, G. Kim, and D. Kim, "Cyber Physical System: Architecture, applications and research challenges," in *2013 IFIP Wireless Days (WD)*, 2013: IEEE, pp. 1-5.
[76] H. Harikrishnan, "A Cyber-Physical Systems Approach to IoT Standards," ed. IoT For All, 2017.
[77] A. Redelinghuys, K. Kruger, and A. Basson, "A six-layer architecture for digital twins with aggregation," in *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, 2019: Springer, pp. 171-182.
[78] J. Leng, H. Zhang, D. Yan, Q. Liu, X. Chen, and D. Zhang, "Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop," *Journal of ambient intelligence and humanized computing,* vol. 10, no. 3, pp. 1155-1166, 2019.
[79] T. Westermann, H. Anacker, R. Dumitrescu, and A. Czaja, "Reference architecture and maturity levels for cyber-physical systems in the mechanical engineering industry," in *2016 IEEE International Symposium on Systems Engineering (ISSE)*, 2016: IEEE, pp. 1-6.
[80] L. Sha *et al.*, "Real time scheduling theory: A historical perspective," *Real-time systems,* vol. 28, no. 2-3, pp. 101-155, 2004.
[81] E. a. R. Staff, Gina and Pao, Charles and Hillar, Gastón and Roos, Gina and Yoshida, Junko. "RTOS trends and challenges - Embedded.com." Embedded.com. https://www.embedded.com/rtos-trends-and-challenges/ (accessed April 28, 2020).
[82] "Real-Time Operating System What it is and why you might want to use one." Rtos.com. https://rtos.com/rtos/ (accessed April 28, 2020).
[83] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in cps-based production systems," *Procedia Manufacturing,* vol. 11, pp. 939-948, 2017.
[84] B. A. Talkhestani, N. Jazdi, W. Schlögl, and M. Weyrich, "A concept in synchronization of virtual production system with real factory based on anchor-point method," *Procedia Cirp,* vol. 67, pp. 13-17, 2018.
[85] A. Karakra, F. Fontanili, E. Lamine, and J. Lamothe, "HospiT'Win: A Predictive Simulation-Based Digital Twin for Patients Pathways in Hospital," in *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, 2019: IEEE, pp. 1-4.
[86] T. Gabor, L. Belzner, M. Kiermeier, M. T. Beck, and A. Neitz, "A simulation-based architecture for smart cyber-physical systems," in *2016 IEEE International Conference on Autonomic Computing (ICAC)*, 2016: IEEE, pp. 374-379.
[87] M. Bajaj, B. Cole, and D. Zwemer, "Architecture to geometry-integrating system models with mechanical design," in *AIAA SPACE 2016*, 2016, p. 5470.
[88] M. Schluse and J. Rossmann, "From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems," in

2016 IEEE International Symposium on Systems Engineering (ISSE), 2016: IEEE, pp. 1-6.

[89] H. Zhang, Q. Liu, X. Chen, D. Zhang, and J. Leng, "A digital twin-based approach for designing and multi-objective optimization of hollow glass production line," *Ieee Access,* vol. 5, pp. 26901-26911, 2017.

[90] K. M. Alam and A. El Saddik, "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE access,* vol. 5, pp. 2050-2062, 2017.

[91] R. Stark, S. Kind, and S. Neumeyer, "Innovations in digital modelling for next generation manufacturing system design," *CIRP Annals,* vol. 66, no. 1, pp. 169-172, 2017.

[92] M. Ciavotta, M. Alge, S. Menato, D. Rovere, and P. Pedrazzoli, "A microservice-based middleware for the digital factory," *Procedia Manufacturing,* vol. 11, pp. 931-938, 2017.

[93] G. Knapp *et al.*, "Building blocks for a digital twin of additive manufacturing," *Acta Materialia,* vol. 135, pp. 390-399, 2017.

[94] B. Brenner and V. Hummel, "Digital twin as enabler for an innovative digital shopfloor management system in the ESB Logistics Learning Factory at Reutlingen-University," *Procedia Manufacturing,* vol. 9, pp. 198-205, 2017.

[95] C. Weber, J. Königsberger, L. Kassner, and B. Mitschang, "M2DDM–a maturity model for data-driven manufacturing," *Procedia CIRP,* vol. 63, pp. 173-178, 2017.

[96] R. Söderberg, K. Wärmefjord, J. S. Carlson, and L. Lindkvist, "Toward a Digital Twin for real-time geometry assurance in individualized production," *CIRP Annals,* vol. 66, no. 1, pp. 137-140, 2017.

[97] C. Zhuang, J. Liu, and H. Xiong, "Digital twin-based smart production management and control framework for the complex product assembly shop-floor," *The international journal of advanced manufacturing technology,* vol. 96, no. 1-4, pp. 1149-1163, 2018.

[98] B. Gockel, A. Tudor, M. Brandyberry, R. Penmetsa, and E. Tuegel, "Challenges with structural life forecasting using realistic mission profiles," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1813.

[99] E. Tuegel, "The airframe digital twin: some challenges to realization," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1812.

[100] P. K. Majumdar, M. FaisalHaider, and K. Reifsnider, "Multi-physics response of structural composites and framework for modeling using material geometry," in *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2013, p. 1577.

[101] K. Reifsnider and P. Majumdar, "Multiphysics stimulated simulation digital twin methods for fleet management," in *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2013, p. 1578.

[102] J. Lee, E. Lapira, B. Bagheri, and H.-a. Kao, "Recent advances and trends in predictive manufacturing systems in big data environment," *Manufacturing letters,* vol. 1, no. 1, pp. 38-41, 2013.

[103] A. Cerrone, J. Hochhalter, G. Heber, and A. Ingraffea, "On the effects of modeling as-manufactured geometry: toward digital twin," *International Journal of Aerospace Engineering,* vol. 2014, 2014.

[104] Y. Bazilevs, X. Deng, A. Korobenko, F. Lanza di Scalea, M. Todd, and S. Taylor, "Isogeometric fatigue damage prediction in large-scale composite structures driven by dynamic sensor data," *Journal of Applied Mechanics,* vol. 82, no. 9, 2015.
[105] C. Li, S. Mahadevan, Y. Ling, L. Wang, and S. Choze, "A dynamic Bayesian network approach for digital twin," in *19th AIAA Non-Deterministic Approaches Conference*, 2017, p. 1566.
[106] R. Asimov, S. Chernoshey, I. Kruse, and V. Osipovich, "Digital twin in the Analysis of a Big Data," *Big Data and Advanced Analytics,* no. 4, pp. 70-79, 2018.
[107] Z. Liu, N. Meyendorf, and N. Mrad, "The role of data fusion in predictive maintenance using digital twin," in *AIP Conference Proceedings*, 2018, vol. 1949, no. 1: AIP Publishing LLC, p. 020023.
[108] M. Shafto *et al.*, "Modeling, simulation, information technology & processing roadmap," *National Aeronautics and Space Administration,* 2012.
[109] J. Ríos, J. C. Hernández, M. Oliva, and F. Mas, "Product Avatar as Digital Counterpart of a Physical Individual Product: Literature Review and Implications in an Aircraft," in *ISPE CE*, 2015, pp. 657-666.
[110] R. Rosen, G. Von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine,* vol. 48, no. 3, pp. 567-572, 2015.
[111] B. Bielefeldt, J. Hochhalter, and D. Hartl, "Computationally efficient analysis of SMA sensory particles embedded in complex aerostructures using a substructure approach," in *ASME 2015 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, 2015: American Society of Mechanical Engineers Digital Collection.
[112] M. Abramovici, J. C. Göbel, and H. B. Dang, "Semantic data management for the development and continuous reconfiguration of smart products and systems," *CIRP Annals,* vol. 65, no. 1, pp. 185-188, 2016.
[113] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital twin data modeling with automationml and a communication methodology for data exchange," *IFAC-PapersOnLine,* vol. 49, no. 30, pp. 12-17, 2016.
[114] A. Canedo, "Industrial IoT lifecycle via digital twins," in *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, 2016, pp. 1-1.
[115] E. M. Kraft, "The air force digital thread/digital twin-life cycle integration and use of computational and experimental knowledge," in *54th AIAA Aerospace Sciences Meeting*, 2016, p. 0897.
[116] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, "Shaping the digital twin for design and production engineering," *CIRP Annals,* vol. 66, no. 1, pp. 141-144, 2017.
[117] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *The International Journal of Advanced Manufacturing Technology,* vol. 94, no. 9-12, pp. 3563-3576, 2018.
[118] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics,* vol. 15, no. 4, pp. 2405-2415, 2018.
[119] S. Haag and R. Anderl, "Digital twin–Proof of concept," *Manufacturing Letters,* vol. 15, pp. 64-66, 2018.
[120] G. E. Modoni, E. G. Caldarola, M. Sacco, and W. Terkaj, "Synchronizing physical and digital factory: benefits and technical challenges," in *Proceedings of 12th CIRP conference on intelligent computation in manufacturing engineering*, 2018.
[121] R. Ganguli and S. Adhikari, "The digital twin of discrete dynamic systems: Initial approaches and future challenges," *Applied Mathematical Modelling,* vol. 77, pp. 1110-1128, 2020.

[122] Q. Qi, F. Tao, Y. Zuo, and D. Zhao, "Digital twin service towards smart manufacturing," *Procedia Cirp,* vol. 72, pp. 237-242, 2018.
[123] Altair. "Digital Twin." https://www.altair.com/digital-twin/#:~:text=The%20Altair%20digital%20twin%20integration,transformation%20vision%20on%20your%20terms. (accessed July 1, 2020).
[124] Ansys. "Digital Twin Physics-Based Simulation & Analysis." https://www.ansys.com/products/systems/digital-twin#:~:text=Digital%20twins%20created%20using%20Ansys,virtually%20before%20doing%20physical%20repairs. (accessed July 1, 2020).
[125] Dassault. "Manufacturing Excellence through Virtual Factory Replication." https://www.3ds.com/products-services/delmia/resources/digital-twin-whitepaper/ (accessed July 1, 2020).
[126] GE. "Digital Twin Framework." https://www.ge.com/research/project/digital-twin-framework#:~:text=A%20Digital%20Twin%20is%20a,engine%20or%20a%20wind%20turbine.&text=Digital%20Twin%20Framework%20(GENIX)%20technology,of%20their%20own%20unique%20assets. (accessed July 1, 2020).
[127] IBM. "How will digital twin be used?" https://www.ibm.com/internet-of-things/trending/digital-twin (accessed July 1, 2020).
[128] Oracle. "About the Oracle IoT Digital Twin Implementation." https://docs.oracle.com/en/cloud/paas/iot-cloud/iotgs/oracle-iot-digital-twin-implementation.html (accessed July 1, 2020).
[129] S. Thompson. "What is Digital Twin Technology?" https://www.ptc.com/en/product-lifecycle-report/what-is-digital-twin-technology (accessed July 1, 2020).
[130] Siemens. "Digital Twins Simulation at Siemens." https://new.siemens.com/global/en/company/stories/research-technologies/digitaltwin/digital-twin.html#:~:text=Digital%20Twins%20at%20Siemens&text=The%20digital%20twin%20has%20long,opens%20up%20new%20business%20opportunities. (accessed July 1, 2020).
[131] L. Wiley, "Congratulations, it's Twins," vol. 2020, ed, 2018.
[132] SAP. "Digital Twin." https://www.sap.com/sea/products/digital-supply-chain/digital-twin.html (accessed July 1, 2020).
[133] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine,* vol. 51, no. 11, pp. 1016-1022, 2018.
[134] F. Tao, Q. Qi, L. Wang, and A. Nee, "Digital twins and cyber–physical systems toward smart manufacturing and Industry 4.0: correlation and comparison," *Engineering,* vol. 5, no. 4, pp. 653-661, 2019.
[135] K. Josifovska, E. Yigitbas, and G. Engels, "Reference framework for digital twins within cyber-physical systems," in *2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)*, 2019: IEEE, pp. 25-31.
[136] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary perspectives on complex systems*: Springer, 2017, pp. 85-113.
[137] S. Yun, J.-H. Park, and W.-T. Kim, "Data-centric middleware based digital twin platform for dependable cyber-physical systems," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2017: IEEE, pp. 922-926.
[138] F. Tao, M. Zhang, Y. Liu, and A. Nee, "Digital twin driven prognostics and health management for complex equipment," *CIRP Annals,* vol. 67, no. 1, pp. 169-172, 2018.

[139] G. Schroeder *et al.*, "Visualising the digital twin using web services and augmented reality," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016: IEEE, pp. 522-527.
[140] M. Zhang, Y. Zuo, and F. Tao, "Equipment energy consumption management in digital twin shop-floor: A framework and potential applications," in *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, 2018: IEEE, pp. 1-5.
[141] A. Singh and A. Singla, "Kinematic modeling of robotic manipulators," *Proceedings of the national academy of sciences, India section A: physical sciences,* vol. 87, no. 3, pp. 303-319, 2017.
[142] S. K. V. Ragavan and M. Shanmugavel, "Engineering cyber-physical systems—Mechatronics wine in new bottles?," in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2016: IEEE, pp. 1-5.
[143] N. P. Suh, "Complexity in engineering," *CIRP annals,* vol. 54, no. 2, pp. 46-63, 2005.
[144] P. N. Sheth and J. Uicker Jr, "A generalized symbolic notation for mechanisms," 1971.
[145] W. Khalil and J. Kleinfinger, "A new geometric notation for open and closed-loop robots," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, 1986, vol. 3: IEEE, pp. 1174-1179.
[146] U. Thomas, I. Maciuszek, and F. M. Wahl, "A unified notation for serial, parallel, and hybrid kinematic structures," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, 2002, vol. 3: IEEE, pp. 2868-2873.
[147] J. C. Chou and M. Kamel, "Quaternions approach to solve the kinematic equation of rotation, a/sub a/a/sub x/= a/sub x/a/sub b/, of a sensor-mounted robotic manipulator," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 1988: IEEE, pp. 656-662.
[148] R. Gilmore, *Lie groups, physics, and geometry: an introduction for physicists, engineers and chemists*. Cambridge University Press, 2008.
[149] C. Rocha, C. Tonetto, and A. Dias, "A comparison between the Denavit–Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators," *Robotics and Computer-Integrated Manufacturing,* vol. 27, no. 4, pp. 723-728, 2011.
[150] M. M. U. Atique and M. A. R. Ahad, "Inverse Kinematics solution for a 3DOF robotic structure using Denavit-Hartenberg Convention," in *2014 international conference on informatics, electronics & vision (ICIEV)*, 2014: IEEE, pp. 1-5.
[151] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of cyber-physical systems," in *2011 international conference on wireless communications and signal processing (WCSP)*, 2011: IEEE, pp. 1-6.
[152] X. Yao, J. Zhou, Y. Lin, Y. Li, H. Yu, and Y. Liu, "Smart manufacturing based on cyber-physical systems and beyond," *Journal of Intelligent Manufacturing,* vol. 30, no. 8, pp. 2805-2817, 2019.
[153] K.-J. Lin and M. Panahi, "A real-time service-oriented framework to support sustainable cyber-physical systems," in *2010 8th IEEE International Conference on Industrial Informatics*, 2010: IEEE, pp. 15-21.
[154] S. V. Ragavan, I. K. Kusnanto, and V. Ganapathy, "Service oriented framework for industrial automation systems," *Procedia Engineering,* vol. 41, pp. 716-723, 2012.
[155] M. Shanmugavel, A. Tsourdos, B. White, and R. Żbikowski, "Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs," *Control Engineering Practice,* vol. 18, no. 9, pp. 1084-1092, 2010.

[156] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.
[157] *Ethernet I/O Module with 16-channels DI*, 2014. [Online]. Available: http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7000_et7200/document/data_sheet/et-7x51_pet-7x51.pdf.
[158] IEEE Std. 610, "IEEE Std 610: Standard Computer Dictionary," *Institute of Electrical and Electronic Engineers,* 1991. [Online]. Available: http://www.mit.jyu.fi/ope/kurssit/TIES462/Materiaalit/IEEE_SoftwareEngGlossary.pdf.
[159] E.-Y. Kang, D. Mu, L. Huang, and Q. Lan, "Verification and validation of a cyber-physical system in the automotive domain," in *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2017: IEEE, pp. 326-333.
[160] F. S. Gonçalves, G. V. Raffo, and L. B. Becker, "Managing CPS complexity: Design method for Unmanned Aerial Vehicles," *IFAC-PapersOnLine,* vol. 49, no. 32, pp. 141-146, 2016.
[161] M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar, "Secure mqtt for internet of things (iot)," in *2015 Fifth International Conference on Communication Systems and Network Technologies*, 2015: IEEE, pp. 746-751.
[162] G. Perrone, M. Vecchio, R. Pecori, and R. Giaffreda, "The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyber-attack Carried Out through an Army of IoT Devices," in *IoTBDS*, 2017, pp. 246-253.
[163] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2017: IEEE, pp. 1-6.
[164] Q. M. C. System. https://www.qualisys.com/ (accessed April 28, 2020).
[165] NXP. *NXP CUP Track Configurations 2018*, 2018. [Online]. Available: https://community.nxp.com/t5/University-Programs-Knowledge/Purchasing-a-NXP-Cup-Track/ta-p/1121390.
[166] G. Aures and C. Lübben, "DDS vs. MQTT vs. VSL for IoT," *Network,* vol. 1, 2019.
[167] S. Profanter, A. Tekat, K. Dorofeev, M. Rickert, and A. Knoll, "OPC UA versus ROS, DDS, and MQTT: performance evaluation of industry 4.0 protocols," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, 2019.
[168] S. Lee, H. Kim, D.-k. Hong, and H. Ju, "Correlation analysis of MQTT loss and delay according to QoS level," in *The International Conference on Information Networking 2013 (ICOIN)*, 2013: IEEE, pp. 714-717.
[169] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in *2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)*, 2014: IEEE, pp. 1-6.
[170] R. Atmoko, R. Riantini, and M. Hasin, "IoT real time data acquisition using MQTT protocol," in *J. Phys. Conf. Ser*, 2017, vol. 853, no. 1.
[171] N. Semiconductors, *Kinetis K64F Sub-Family Data Sheet*, 2016. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/K64P144M120SF5.pdf.
[172] MathWorks Internet of Things Team. "MQTT in MATLAB." https://www.mathworks.com/matlabcentral/fileexchange/64303-mqtt-in-matlab (accessed.
[173] H. Bohner and M. Moser, "Vehicle steering system," ed: Google Patents, 2000.
[174] M. E. Salmon, "Vehicle steering system," ed: Google Patents, 1987.

[175] Lauszus, "A practical approach to Kalman filter and how to implement it," ed. TKJ Electronics, 2012.
[176] B. Delporte, L. Perroton, T. Grandpierre, and J. Trichet, "Accelerometer and magnetometer based gyroscope emulation on smart sensor for a virtual reality application," 2012.
[177] M. Tuupola. "How to Calibrate a Magnetometer?" https://appelsiini.net/2018/calibrate-magnetometer/ (accessed April 28, 2020).
[178] I. Freescale Semiconductor, *Kinetis KL25 Sub-Family*, 2014. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/KL25P80M48SF0.pdf.
[179] Mbed. "mbed Compiler." https://os.mbed.com/handbook/mbed-Compiler (accessed April 28, 2020).
[180] E. Systems, *ESP8266EX Datasheet*, 2020. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
[181] A. Corsaro. Building the Internet of Things. Available: https://www.slideshare.net/Angelo.Corsaro/building-the-internet-of-things-22196885?from_search=24.
[182] O. Oman. Internet of Things and Device Management Protocols, Clients and Servers. Available: https://www.youtube.com/watch?v=WtRpFLx34BY.
[183] T. Harwood, "IoT Standards and Protocols." [Online]. Available: https://www.postscapes.com/internet-of-things-protocols/
[184] V. Sarafov, "Comparison of iot data protocol overhead," *Network,* vol. 7, 2018.
[185] Steve. "How to Install The Mosquitto MQTT Broker on Windows." http://www.steves-internet-guide.com/install-mosquitto-broker/ (accessed April 28, 2020).
[186] L. Zhejiang Zhongde Electric Co. "Zhejiang Zhongde Electric Co., Ltd." http://proximity-sensorswitch.sell.everychina.com/p-109564076-npn-dc-capacitive-proximity-switch-capacitive-proximity-detector-normally-open.html (accessed April 28, 2020).
[187] *Series 3 valves and solenoid valves*, 2019. [Online]. Available: http://catalogue.camozzi.com/CATALOGUES/CCC-GENCAT/00153/PDF/ENG.4.2.10.pdf.
[188] Siemens. "Simatic S7-1200 Programmable Controller System Manual." https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf (accessed April 28, 2020).
[189] Siemens, *Learn-/Training Textbook TIA Portal Modules for Automation System SIMATIC S7-1200 from Version V14 SP1*, 2017. [Online]. Available: https://www.automation.siemens.com/sce-static/learning-training-documents/tia-portal/learn-training-textbook-s7-1200-en.pdf.
[190] Advantech, *User Manual ECU-1251 Series*, 2017. [Online]. Available: http://advdownload.advantech.com/productfile/Downloadfile1/1-1D43F3Q/ECU-1251_User_Manual_Ed.1.pdf.
[191] Advantech, *User Manual WISE-4000 Series IoT Ethernet I/O Module*, 2016. [Online]. Available: http://advdownload.advantech.com/productfile/Downloadfile2/1-14JNLJL/UM-WISE-4000-Ed.4-EN.pdf.
[192] Advantech. "Advantech EdgeLink Studio 2.6.0.2." https://support.advantech.com/support/DownloadSRDetail_New.aspx?SR_ID=1-1U4PNVY&Doc_Source=Download (accessed 28/4, 2020).

Appendix

ITS Domain CPS

Sensors

In the ITS domain, the autonomous vehicles require the information on the vehicle's speed and direction for control. The vehicle must maintain constant speed throughout the experiments due to the prerequisite of implementing the Dubins path planning service. The speed of the vehicle was measured using hall effect sensors and a set of magnets fitted into a designed 3D printed mounting in Figure 45 Magnet mounting on rear wheels for sensing vehicle speed. The number of magnets inserted in the mounting and the vehicle's speed decide the frequency of detection. Therefore, a high-frequency sensing rate is required, and this signifies the real-time performance objective.

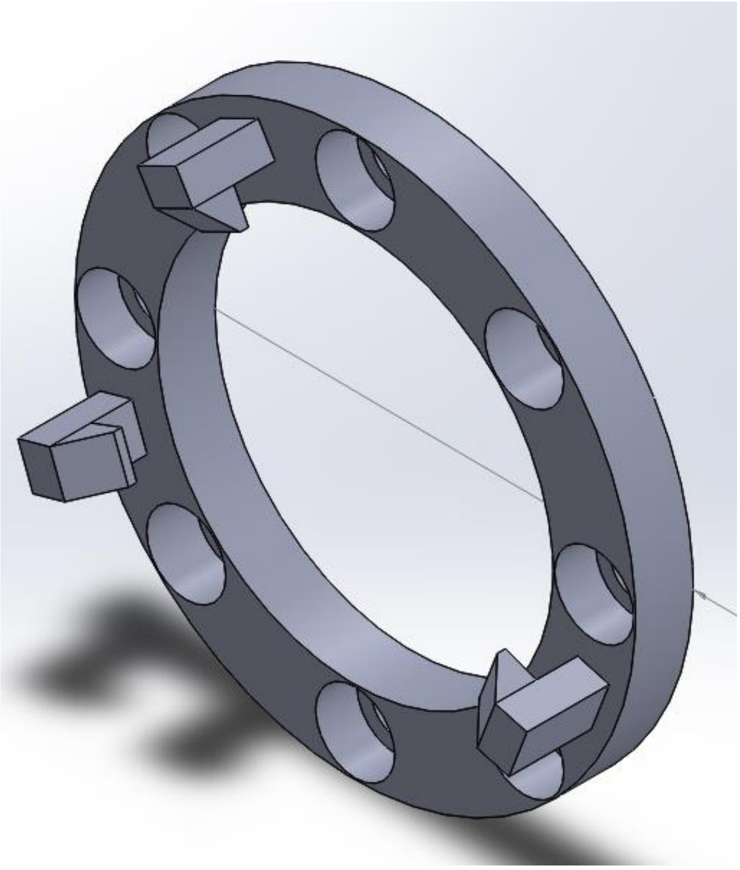


Figure 45 Magnet mounting on rear wheels for sensing vehicle speed.

Besides the speed of the vehicle, estimating the vehicle's direction is crucial for path following. One of the primary means of determining the direction is the yaw of the vehicle. The simplest method is to integrate a gyroscope's yaw rate for the yaw. However, gyroscopes tend to drift and cause the readings to be unreliable during the experiment. In order to reduce the drift effect, a Kalman filter[175] was implemented using the gyroscope readings and emulated gyroscope readings. A three-axis accelerometer and three-axis magnetometer readings were used to emulate a gyroscope practicing the method from [176]. Since the magnetometer is near the DC motors with magnetic fields, a soft and hard iron distortion filter was also implemented using [177] and the results are in Figure 46. Therefore, sensing the direction requires capturing sensor data and filtering that entails computation. The computation of sensor data must be performed in real-time for the vehicle to follow the path accurately.

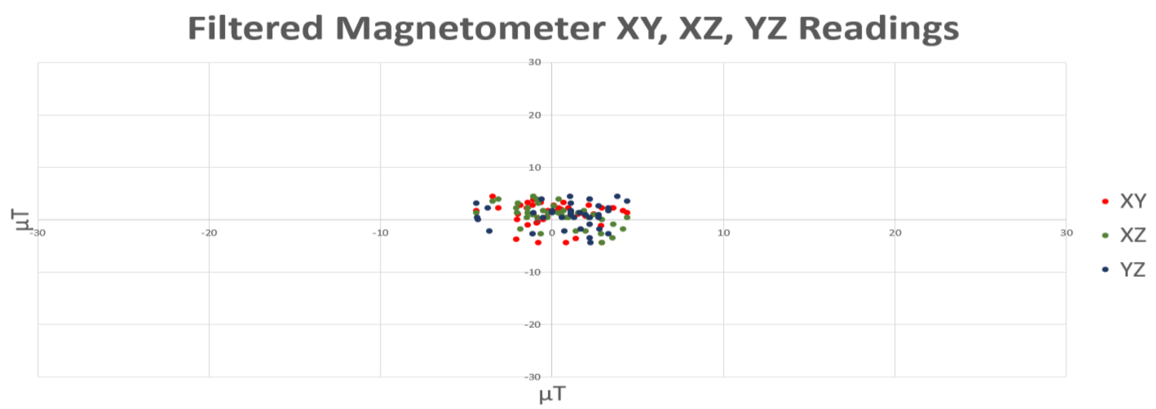


Figure 46 Normalized magnetometer readings after soft and hard iron distortion.

Actuators

The actuation for the ITS domain is divided into driving and steering the vehicle. For scaled vehicles, the components attached are typically battery-powered due to size constraints. DC motor and DC servo motor are the components that provide the drive and steering input, respectively. These motors are controlled using pulse width modulation (PWM) with periods of 5 ms and 25 ms. The periods of the PWM are the interval of time for processing sensor data and reacting to variations. Thus, signifying the importance of developing a real-time physical system.

Microcontroller

The vehicle in this work was a standard-issue scaled race car for an autonomous vehicle competition named NXP cup. The standard-issue microcontroller was the FRDM-K25Z[178], which can be replaced by a newer model at the time, the FRDM-K64F[171]. The newer model offers higher computing power and a three-axis magnetometer. The point of upgrading the microcontroller was to operate the sensors and actuators quicker. However, the motor driver that can be directly mounted on top of the standard microcontroller is not operational on the FRDM-K64F. Therefore, Figure 47 shows a designed PCB that connects the microcontroller and motor driver. The pin locations of the two boards were able to mate but the connections were incorrect. The self-designed PCB solved the issue to ensure reliable connection between the two boards. Upon connecting the two boards, the entire source code in Mbed [179] had to be re-written to accommodate the changes and the addition of sensors. However, the microcontroller's limit was the operation of the entire program within 25ms.

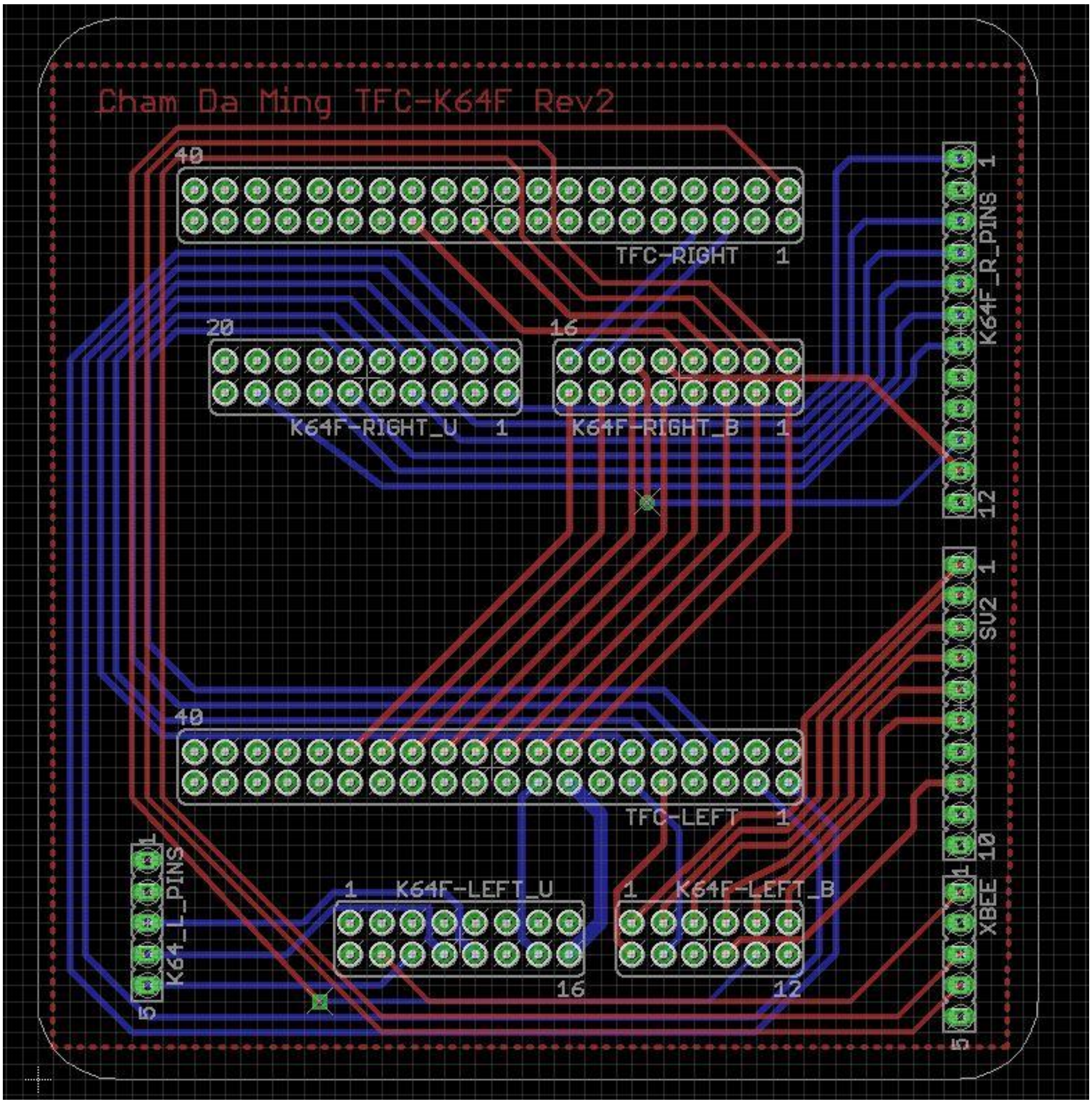


Figure 47 Designed PCB to interface FRDM-K64F and TFC-Shield showed a complex connection between the two boards.

Wi-Fi Module

The core component that bridges the physical and cyber system is the ESP8266[180] Wi-Fi module. This module collects information generated from the microcontroller using serial ports and uses MQTT to send the information wirelessly to the cyber system. Therefore, the transmission of information from physical to cyber systems relies on the timeliness of the microcontroller. Moreover, MQTT was chosen as the primary internet protocol because of transmission speed and security[166, 167, 181-184]. The seamless connection rendered by the Wi-Fi module is one of the foundations for developing the DT.

Cloud Server

There are numerous cloud services for various needs. However, not all are suitable for educational use because of cost, data packet limitations, API request limitations and others. Microsoft Azure is a powerful cloud service provider alongside Amazon Web Service (AWS), Google Cloud Platform and IBM Cloud Services. However, these are premium services that is costly to implement for long periods of time. Altair SmartCore and CloudMQTT are free cloud service providers for limited data packets and API request. Therefore, CPS developers will have to research and chose a suitable cloud server according to the needs of the system. For the generic CPS framework, fast, secure and full control over the cloud server is preferred. Thus, Eclipse Mosquitto MQTT broker was chosen for the ITS cloud server. The MQTT broker was installed and built on a personal laptop outside of campus to simulate servers with different networks. The guide written by Steve [185] described a step-by-step installation of the Mosquitto MQTT broker on a Windows PC. The MQTT broker developed enabled full access to every aspect of the server for monitoring and data storage.

Manufacturing Domain CPS

Sensor

In the manufacturing domain, there is a large quantity of legacy machinery, and the cost of developing a new machine is typically not economical. Therefore, the inclusion of the legacy systems was considered in the development of the generic CPS framework. A simple transport and eject manufacturing system were built to show the implementation of the proposed framework on a legacy manufacturing system. The components of the manufacturing CPS developed using the framework are all COTS components. These COTS components illustrate that CPS developers can replace equivalent components in that layer. NPN DC Capacitive Proximity Switch [186] was used in the manufacturing CPS to detect the workpiece position for actuation. Since it is an NPN sensor, an additional relay was used to change it to PNP in the following figure for digital input into the Programmable Logic Controller (PLC).

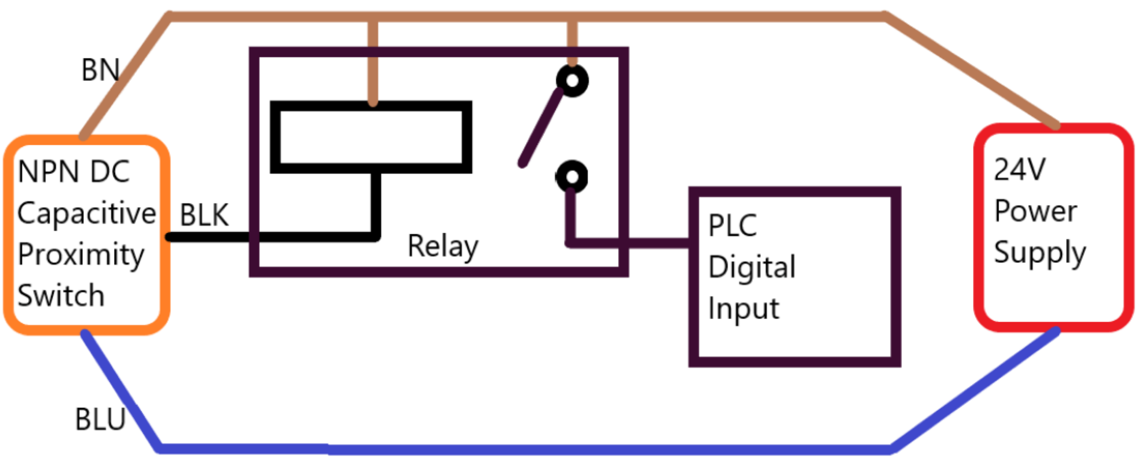


Figure 48 Circuit diagram of NPN DC Capacitive Proximity Switch converting to PNP using a relay and connecting to digital input pin of a PLC.

Actuators

As explained earlier, the physical system of the manufacturing CPS is built from COTS components that can be replaced with equivalent parts for different applications. A 5/2-way double solenoid directional control valve [187] and a pneumatic linear actuator was used to eject the work piece from an electrical conveyor belt. The process functions as a loop to show a simple process for validating the proposed CPS framework.

Microcontroller

One of the standard controllers for manufacturing systems is the PLC, for the purpose of validating the proposed framework, Siemens S7-1200 AC/DC/RLY [188] was used because it is a popular brand for plug and play. The software used to program the PLC is TIA Portal V14 [189]. However, setting up the PLC to communicate with the Advantech ECU1251 gateway required several steps shown in this section from Figure 50 to Figure 53. All these steps are not included in the documentation provided by the Advantech supplier, specific configurations were only given in arranged meetings. The ladder diagram to control the manufacturing system is shown in Figure 54 and Figure 55.

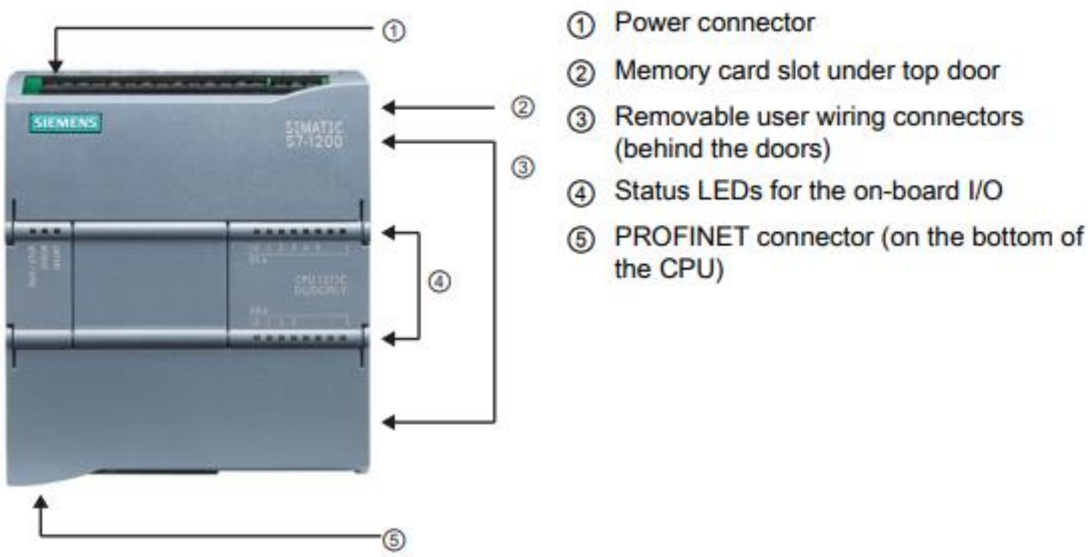


Figure 49 Simatic S7-1200 AC/DC/RLY PLC for controlling the sensors and actuators, also communicating the data to a gateway (Wi-Fi module) using PROFINET.

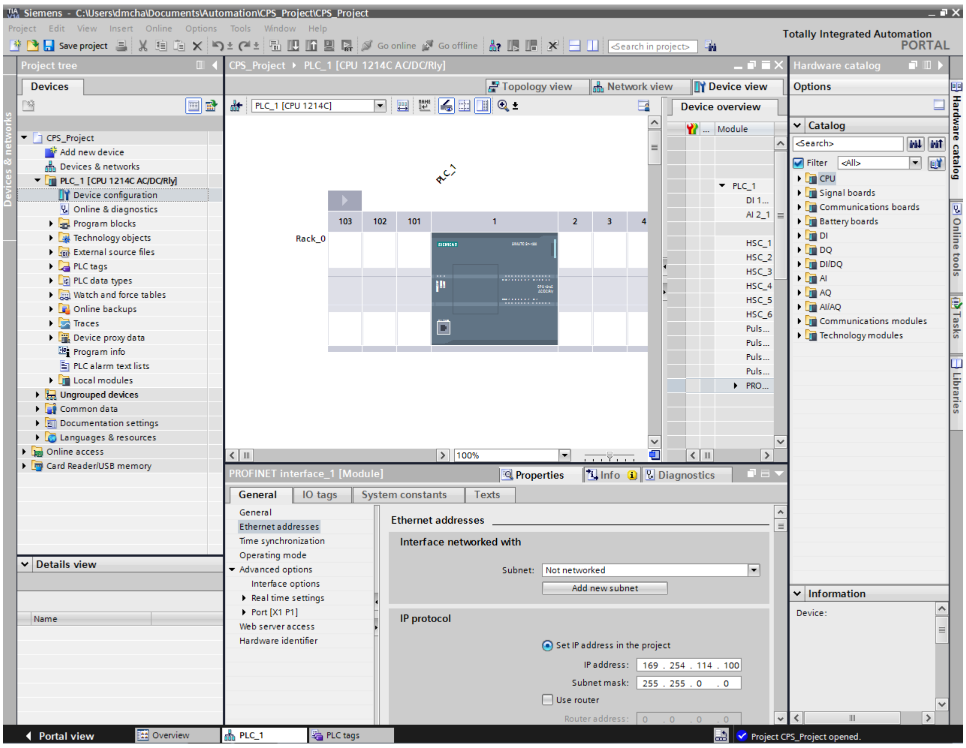


Figure 50 Setting up the PLC with a static IP address, subnet mask, and subnet for PROFINET communication with Advantech ECU1251 Gateway.

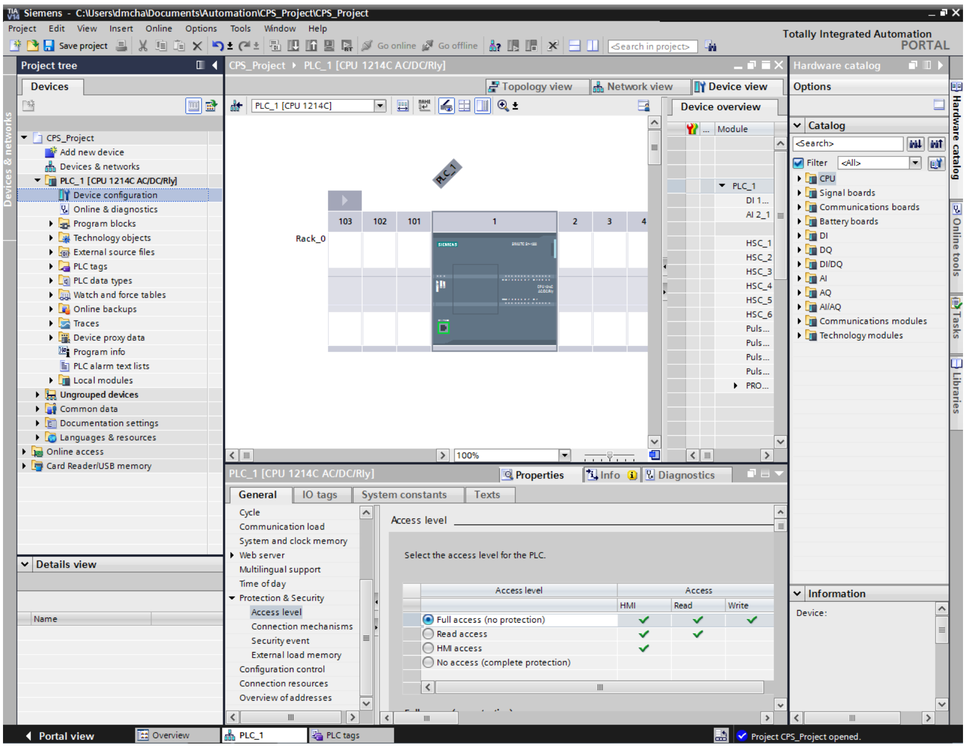


Figure 51 Setting up the access level for PROFINET communication with Advantech ECU1251 Gateway.

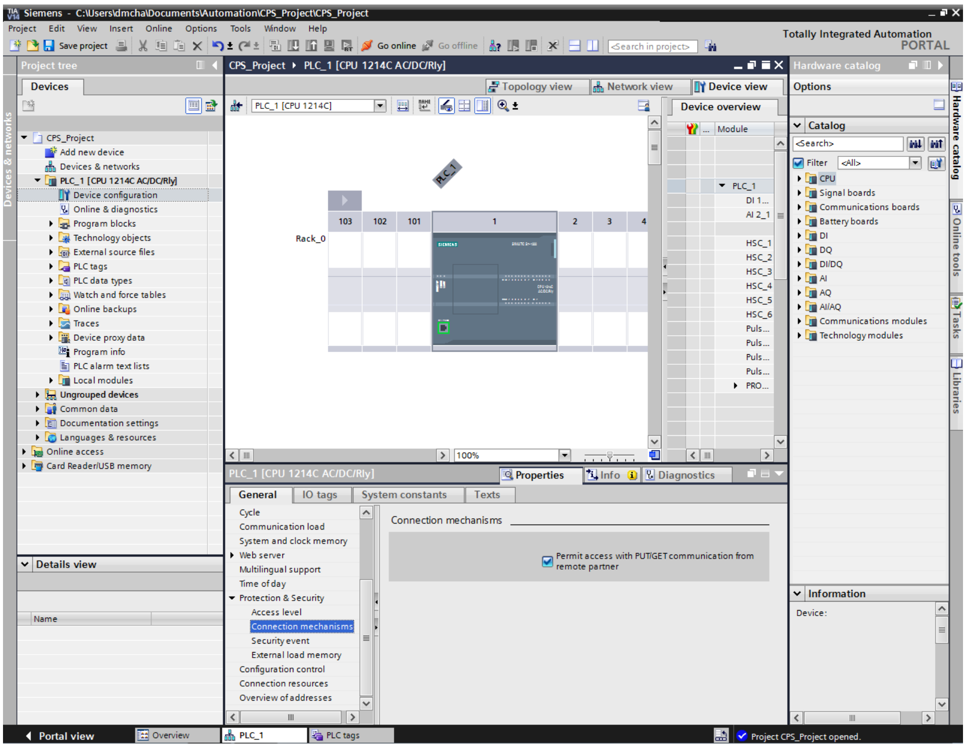


Figure 52 Setting up the connection mechanism for PROFINET communication with Advantech ECU1251 Gateway.

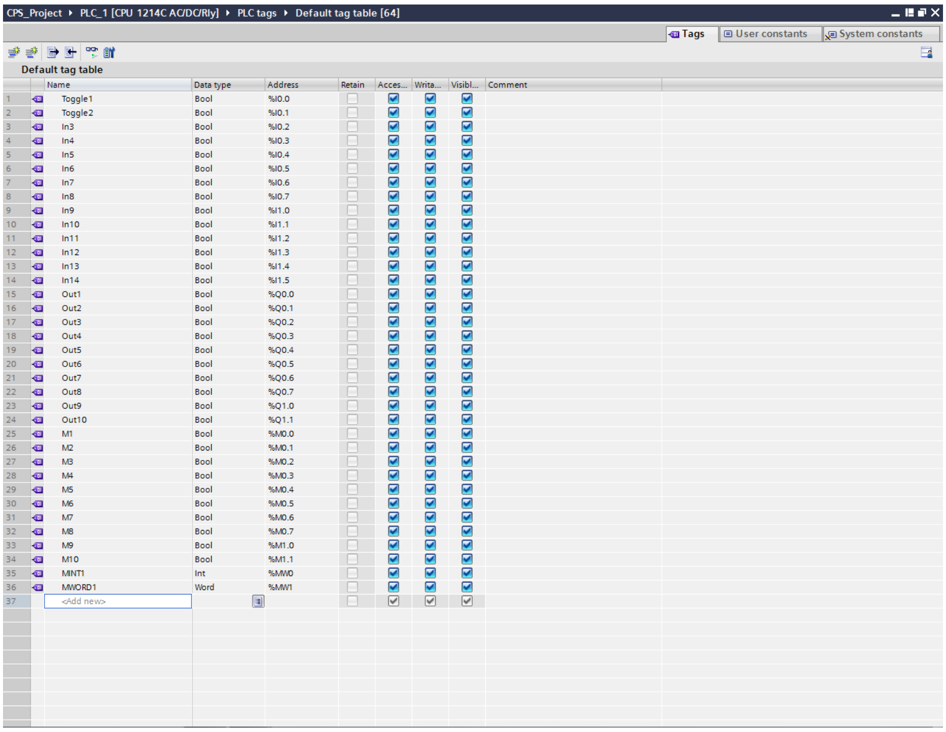


Figure 53 Declaring variables in the PLC to communicate sensor and actuator values to Advantech ECU1251 Gateway.

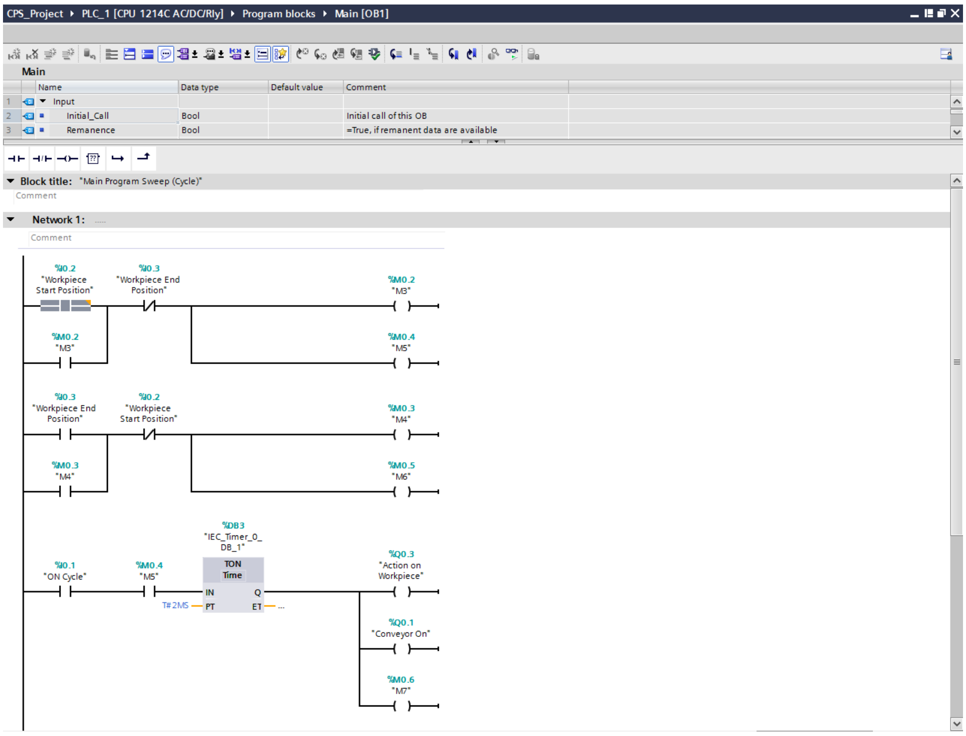


Figure 54 Ladder diagram on the PLC to control the manufacturing system.

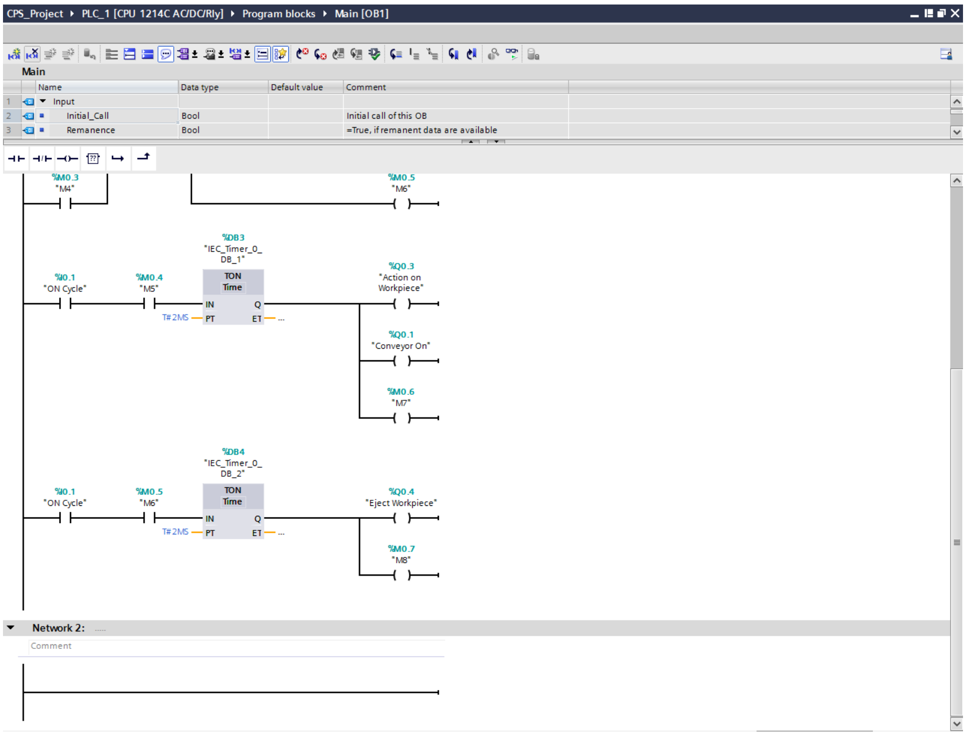


Figure 55 Ladder diagram on the PLC to control the manufacturing system, continued from Figure 54.

Wi-Fi Module (Gateway)

The Advantech ECU1251 gateway [190] is the bridge between physical systems connected to PLC and cloud systems. The gateway was chosen because it is compatible with a variety of physical and cyber systems, due to its ethernet and serial ports. An architecture was drawn out in Figure 56 to illustrate the types of communications across different components. A router was implemented because the ECU1251 only has 2 ethernet ports, the WISE 4051/ WISE 4060 are portable Wi-Fi modules that can be attached to individual manufacturing systems to communicate data to the ECU1251 gateway. The configuration of the WISE-4000 series is documented in [191].

However, configuring the ECU1251 is a tedious process that was done in EdgeLink Studio [192], shown in Figure 59 to Figure 68. The PC with the EdgeLink studio installed must set its Ethernet address to be static, as shown in Figure 57. The steps to correctly configure every component of the manufacturing CPS was challenging because of all the different kinds of communications protocols. Moreover, meetings with suppliers were a must to obtain information that is undocumented for the setup of each component.

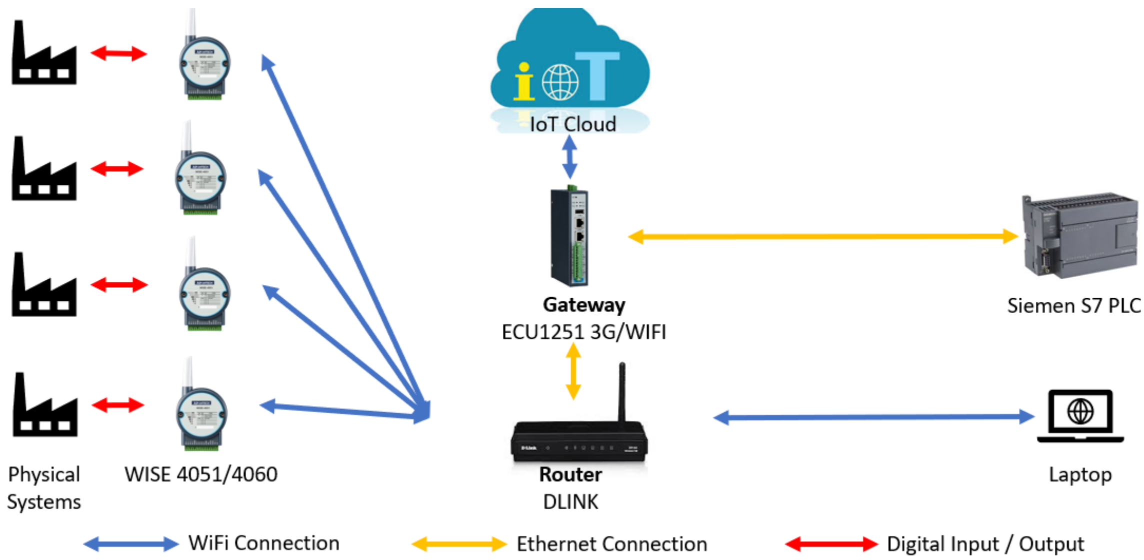


Figure 56 Cyber-Physical System architecture of the manufacturing system illustrating the types of communications.

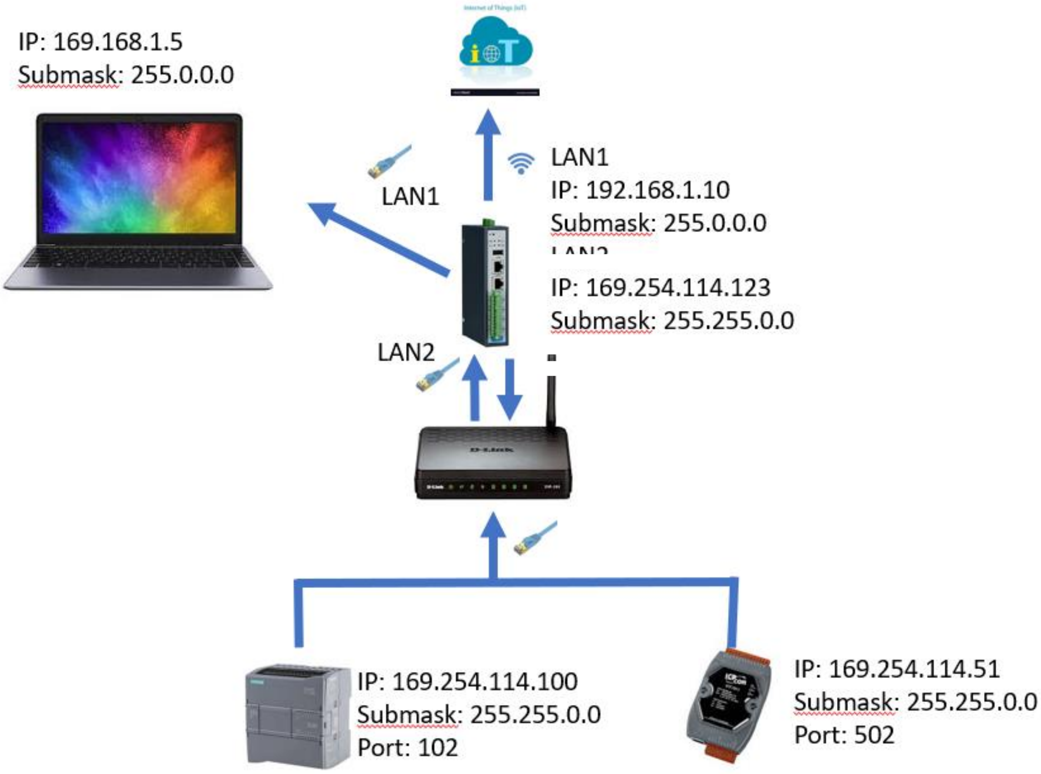


Figure 57 IP addresses on each component in the manufacturing CPS to enable communication between modules.

Figure 58 Advantech ECU1251 gateway for bridging the physical and cyber systems.

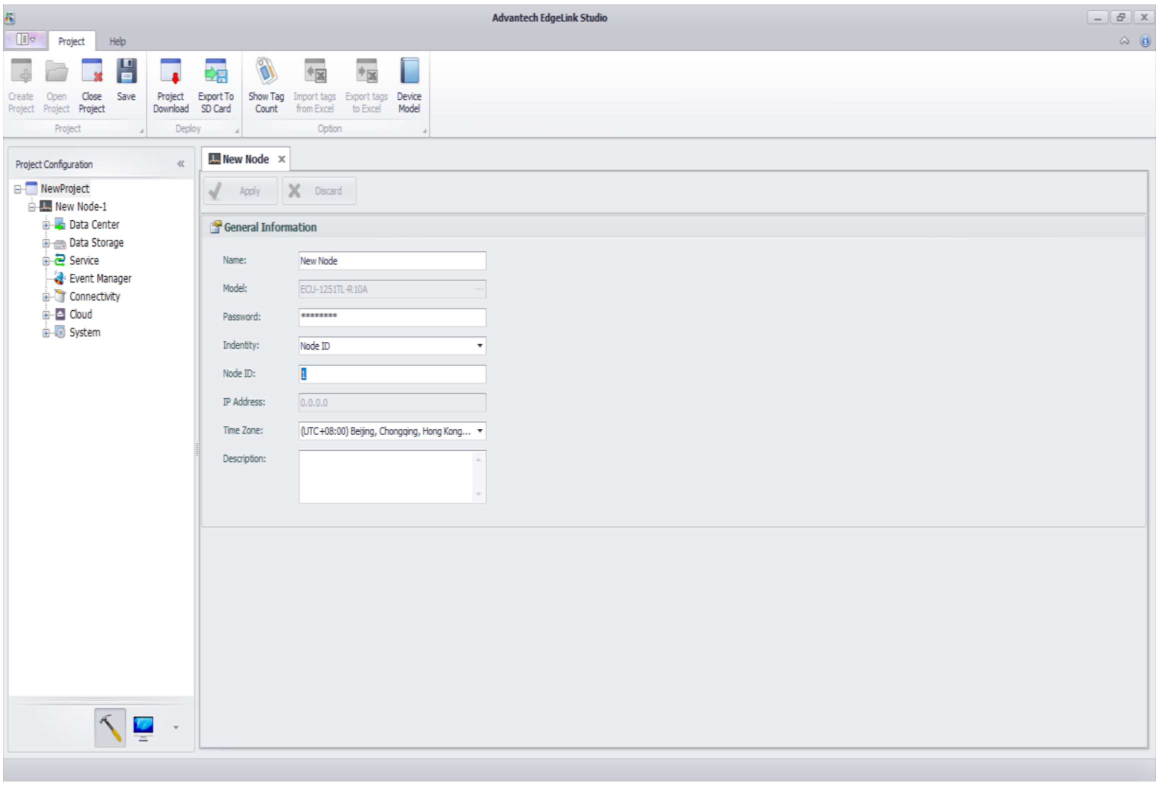


Figure 59 In EdgeLink Studio, create a project with the specifications in the figure. The "Node" is the binary number on the array of DIP switches on the ECU1251.

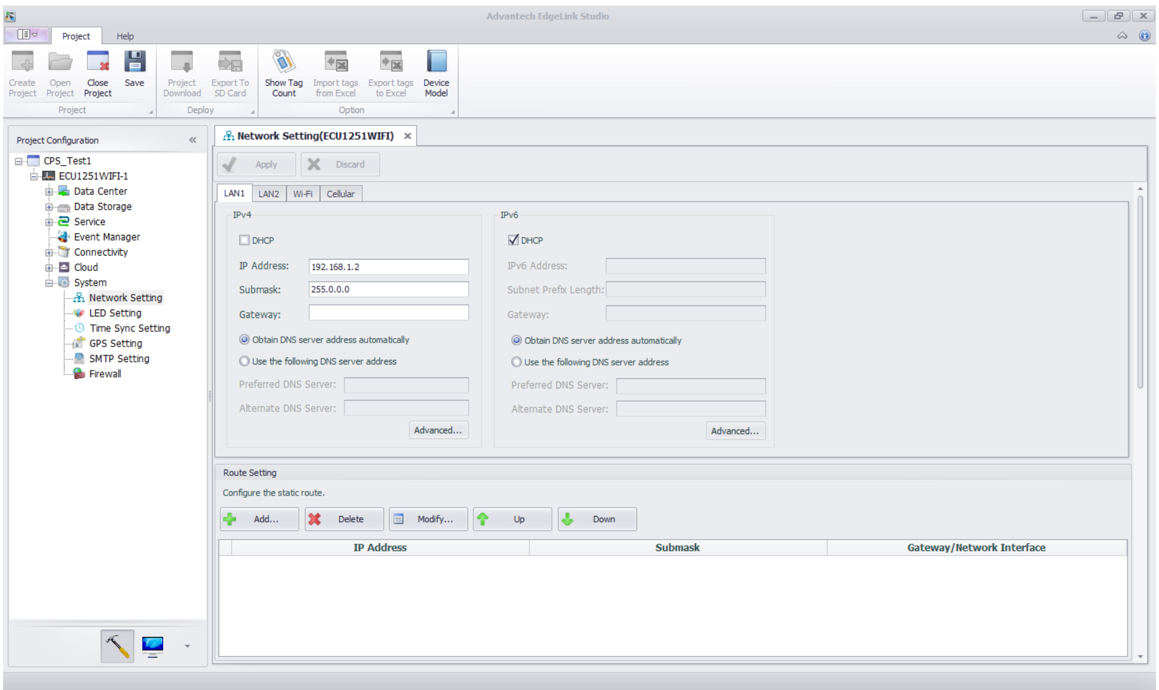


Figure 60 Setting up the IP address and submask following the predefined values above.

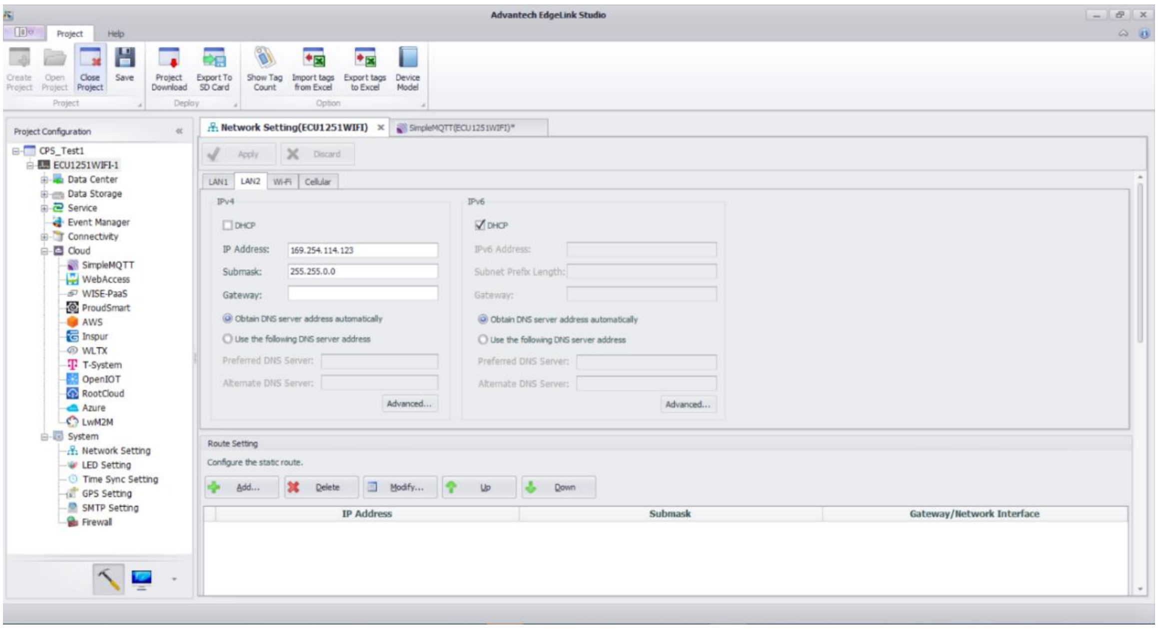


Figure 61 Setting up the IP address and submask following the predefined values above, continued from Figure 60.

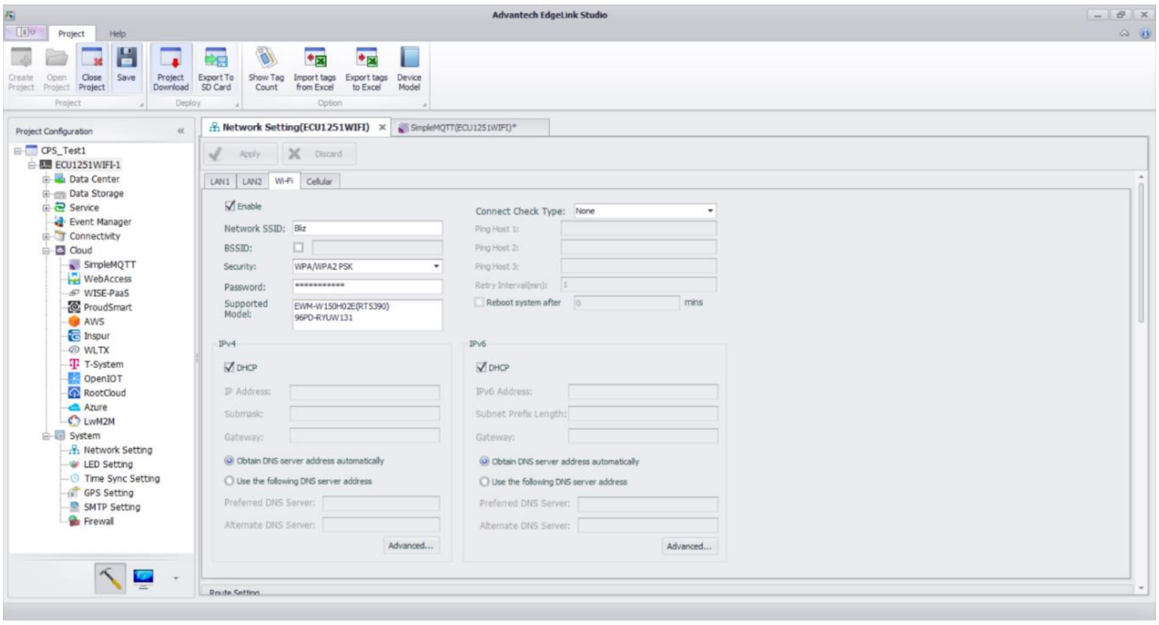


Figure 62 Setting up the Wi-Fi version of ECU1251 to connect to a mobile hotspot. The reason of implementing the mobile hotspot is the restrictions on university's Wi-Fi authentication. However, a 3G version of ECU1251 can also be implemented with different settings in "Cellular" tab.

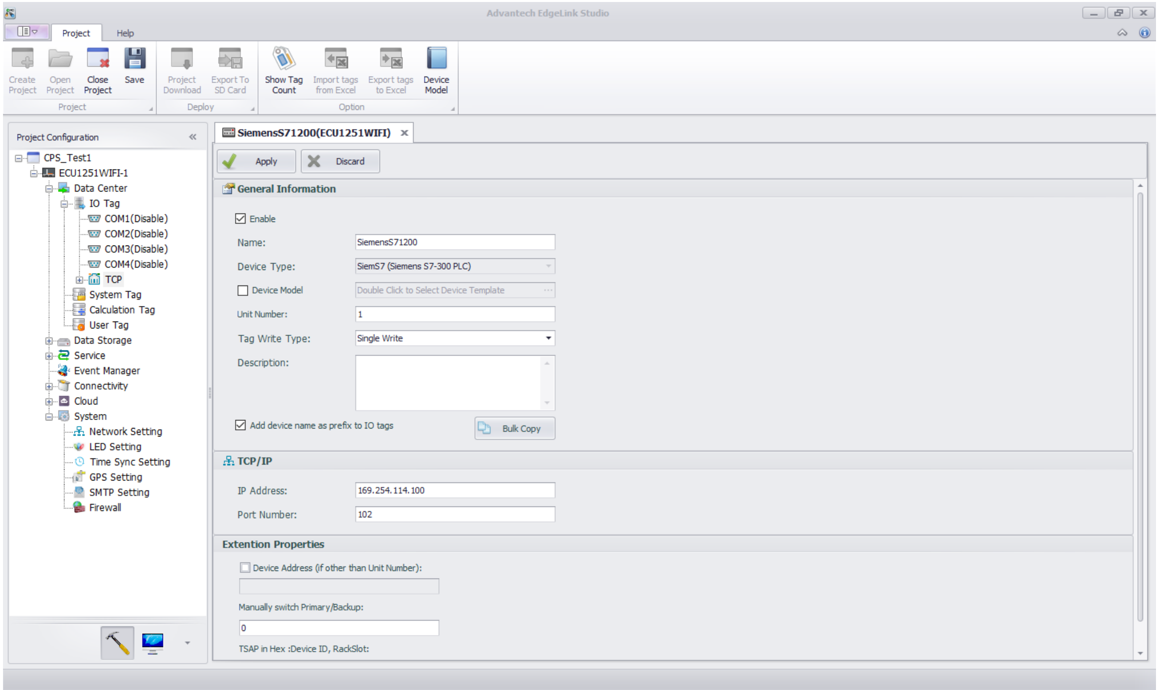


Figure 63 Setting the connection to Siemens S7-1200 AC/DC/RLY PLC with the configuration above. The "Unit Number" follows the device number according to the TIA Portal's device number.

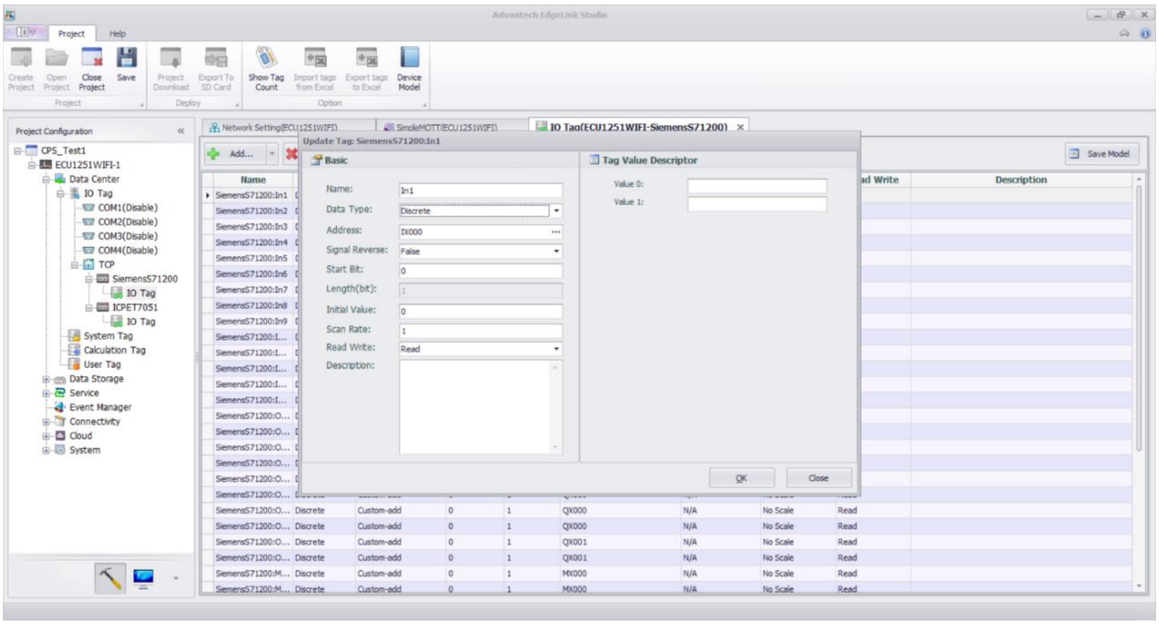


Figure 64 Setting up the communication of digital input pins values of the PLC to the gateway. These values correspond to the components connected to the pin number on the PLC.

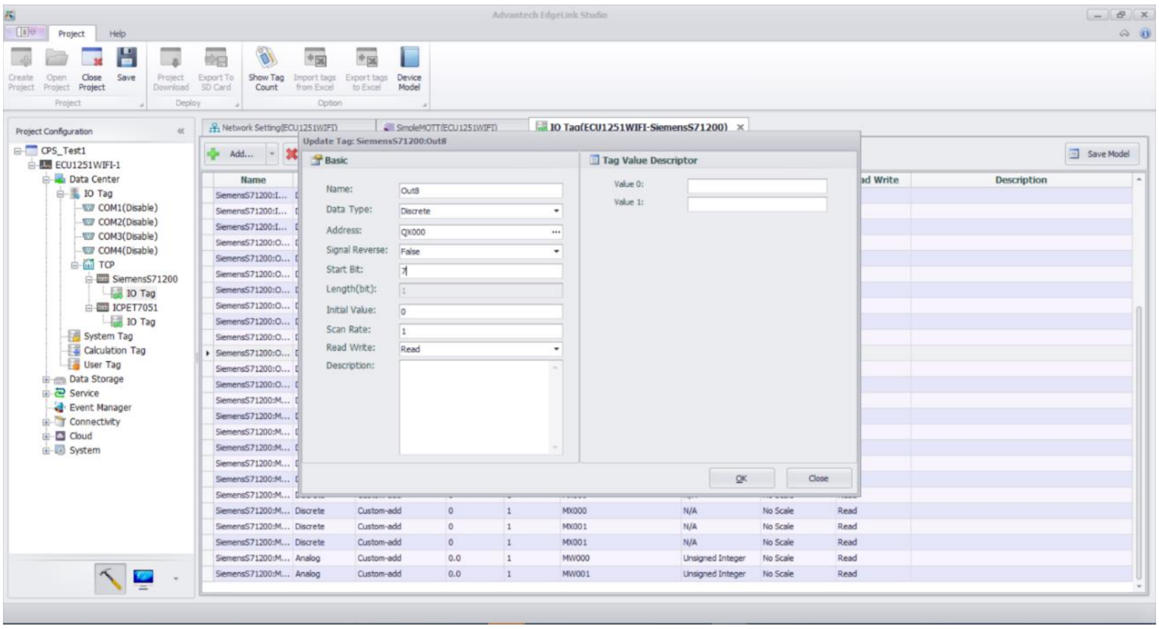


Figure 65 Setting up the communication of digital output pins values of the PLC to the gateway. These values correspond to the components connected to the pin number on the PLC.

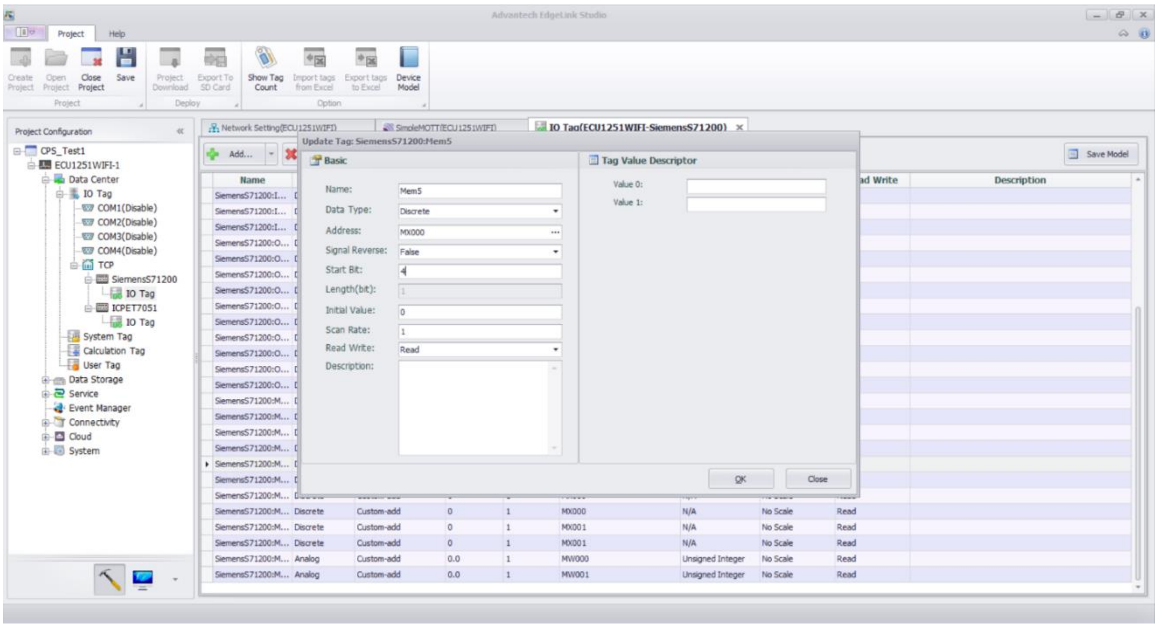


Figure 66 Setting up the communication of memory pins values of the PLC to the gateway. These values correspond to the components connected to the pin number on the PLC.

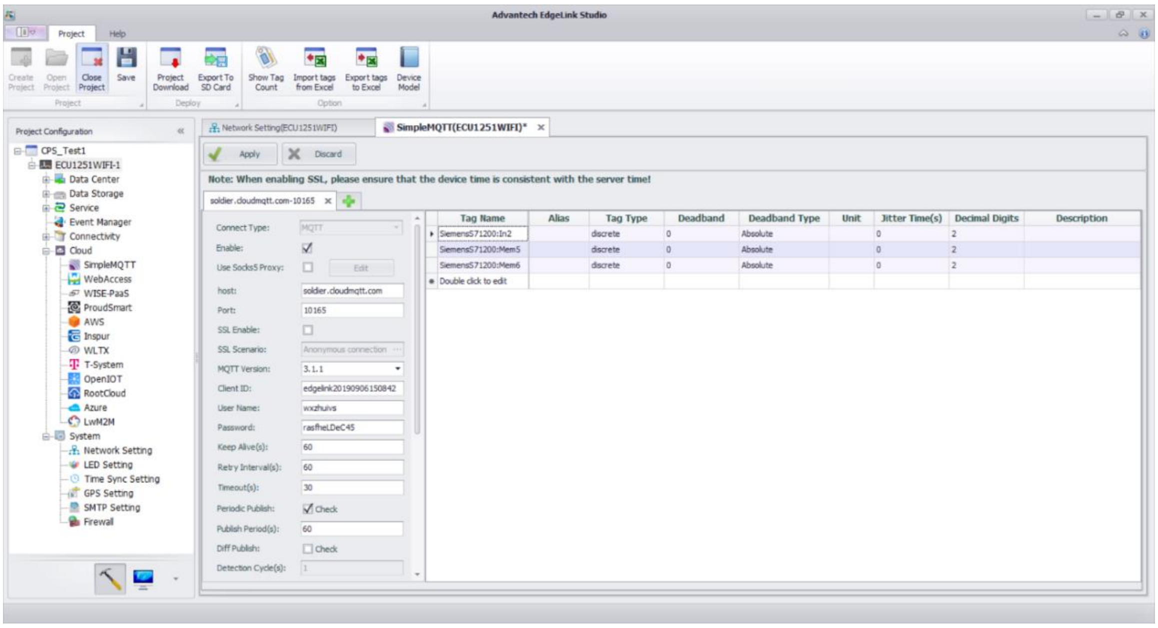


Figure 67 Setting up the MQTT internet protocol to communicate with CloudMQTT a cloud service provider. The three data with tag names will be sent to CloudMQTT. Additional tags can be added in the table.

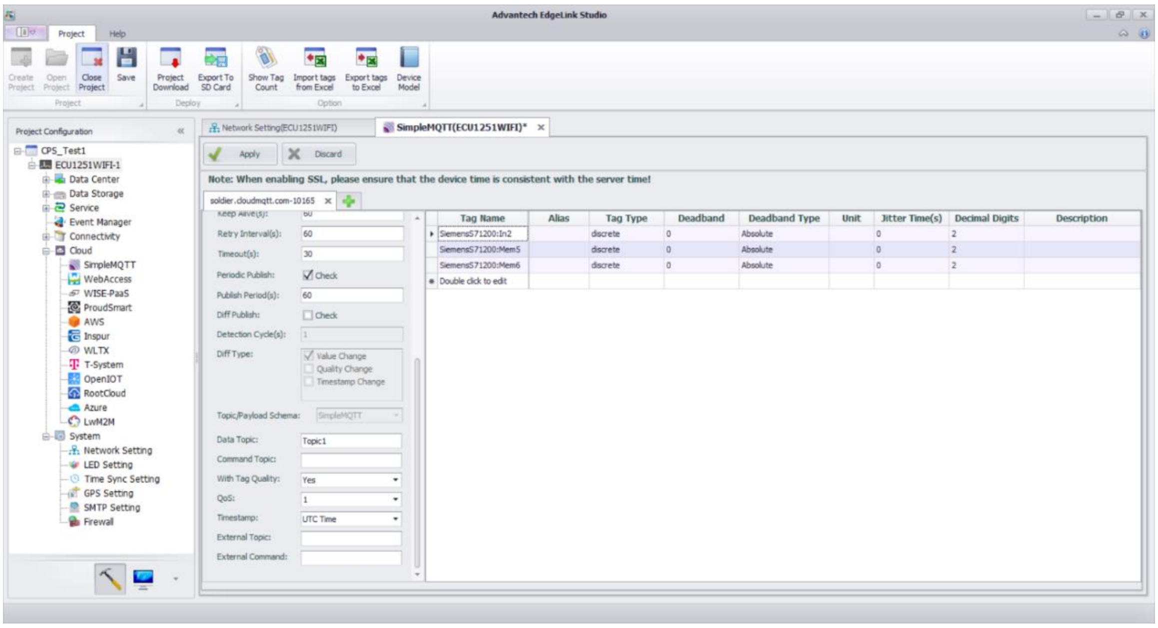


Figure 68 The MQTT internet protocol setup showing the Topic to send the data, continued from Figure 74.

Cloud Server

The cloud service chosen for the manufacturing CPS was CloudMQTT. The free but limited MQTT broker services from CloudMQTT was adequate in Figure 69. Moreover, the ECU1251, when connected to a router, can be accessed by entering its corresponding IP address in a web browser, as shown in Figure 70. However, both these communications are very slow when compared to the ITS domain CPS. This is because of the additional security and PROFINET communication that increased the latency of the CPS.

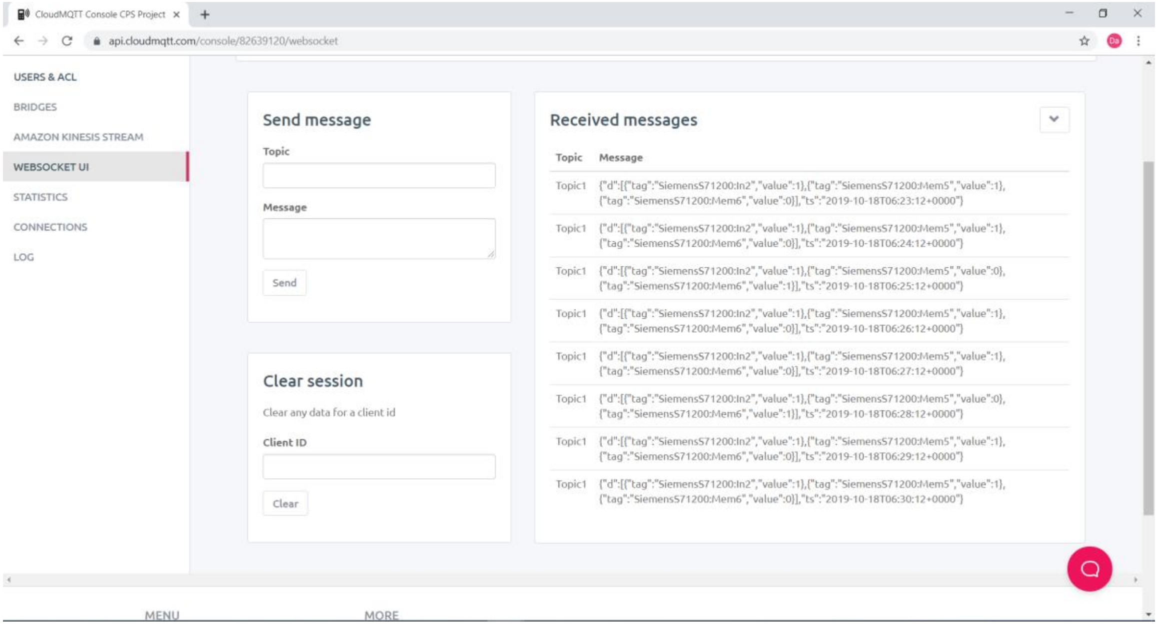


Figure 69 Results from setting up the devices correctly. However, the latency of these networked devices is substantially higher compared to the ITS version.

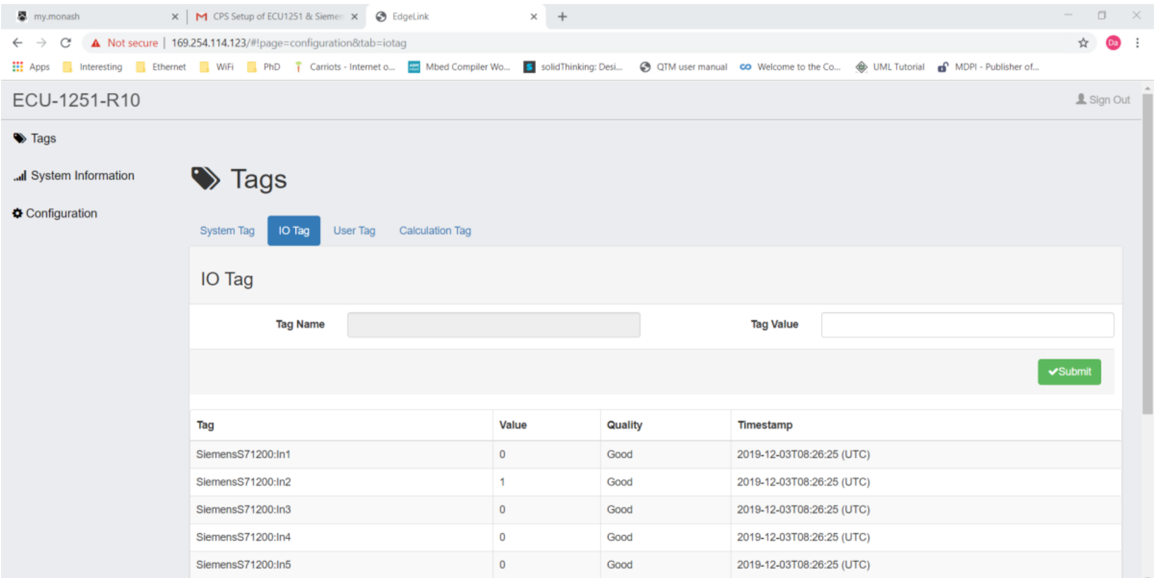


Figure 70 Connecting a PC to the router, entering 169.254.114.123 in a browser and signing in using a password, the tag values from PLC are shown.