

Titre: Jumeau numérique pour la gestion adaptative de la maintenance et de la recharge : approche de synchronisation en temps réel

Auteur: Hedir Oukassi

Date: 2025

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Oukassi, H. (2025). Jumeau numérique pour la gestion adaptative de la maintenance et de la recharge : approche de synchronisation en temps réel [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/70586/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/70586/>
PolyPublie URL:

Directeurs de recherche: Soumaya Yacout, Ali Aidibe, & Amel Jaoua
Advisors:

Programme: Maîtrise recherche en génie industriel
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Jumeau numérique pour la gestion adaptative de la maintenance et de la
recharge : approche de synchronisation en temps réel**

HEDIR OUKASSI

Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie industriel

Décembre 2025

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Jumeau numérique pour la gestion adaptative de la maintenance et de la
recharge : approche de synchronisation en temps réel**

présenté par **Hedir OUKASSI**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Camélia DADOUCHI, présidente

Soumaya YACOUT, membre et directrice de recherche

Ali AIDIBE, membre et codirecteur de recherche

Amel JAOUA, membre et codirectrice de recherche

Mohamed Ossama HASSAN, membre

DÉDICACE

*À tous les étudiants internationaux, qui poursuivent leurs rêves loin de chez eux,
Vous êtes des champions...
À ma famille en Tunisie et à mon mari,
je vous aime...*

REMERCIEMENTS

Je tiens à exprimer toute ma gratitude à Madame Soumaya YACOUT, ma directrice de recherche, pour sa rigueur scientifique, son expertise et ses précieux conseils. Je la remercie également pour le cadre de travail chaleureux et stimulant qu'elle a su instaurer, ainsi que pour son appui constant, tant sur le plan professionnel que personnel, tout au long de ce parcours.

J'exprime également ma reconnaissance à Monsieur Ali AIDIBE, codirecteur de recherche, pour son accompagnement attentif, son souci du détail et la clarté de ses explications. Sa rigueur scientifique, ses remarques constructives et ses encouragements constants ont été essentiels à l'avancement et à la qualité de ce travail.

Mes sincères remerciements vont aussi à Madame Amel JAOUA, codirectrice de recherche, pour son accompagnement rigoureux, ses remarques constructives et ses encouragements stimulants, qui m'ont incitée à dépasser mes limites et ont contribué au développement scientifique de ce mémoire.

Je tiens également à adresser mes sincères remerciements à Madame Camélia DADOUCHI et Monsieur Mohamed Ossama HASSAN, membres du jury, pour avoir accepté d'évaluer ce travail et pour le temps qu'ils y ont consacré.

Je tiens également à remercier Monsieur Jean GHANTOUS, technicien du laboratoire SCP-I, pour son aide précieuse, sa disponibilité et son soutien technique tout au long de mes expérimentations.

RÉSUMÉ

L'intégration des jumeaux numériques (JN) basés sur la simulation à événements discrets (SED) constitue une nouvelle approche pour soutenir la prise de décision en temps réel dans les environnements manufacturiers et logistiques. Ce mémoire présente la conception et la mise en œuvre d'un jumeau numérique à base de SED au sein du Laboratoire des systèmes cyber-physiques intelligents (SCP-I) de Polytechnique Montréal. L'objectif est de développer un mécanisme de synchronisation en temps réel permettant de gérer de manière adaptative les opérations de maintenance et de recharge des robots mobiles autonomes (Autonomous Mobile Robot, ou AMR) dans un entrepôt intelligent.

Le cadre proposé combine deux modes de synchronisation complémentaires : la synchronisation périodique (Time-Triggered, ou TT) pour les actions de commande et la synchronisation déclenchée par événement (Event-Triggered, ou ET) pour la surveillance et le suivi des changements d'état. L'architecture, conforme à la norme ISO 23247, intègre un modèle SED servant de noyau décisionnel et assurant la synchronisation en temps réel entre le système physique et sa contrepartie numérique.

Les expérimentations menées au SCP-I montrent que le mécanisme de synchronisation sélective réduit la latence de communication entre le système physique et sa contrepartie numérique, améliorant ainsi la continuité opérationnelle. Ce travail propose une démarche méthodique pour la conception de jumeaux numériques à base de SED capables de soutenir la prise de décision en temps réel dans les systèmes industriels.

ABSTRACT

The integration of Discrete-Event Simulation (DES)-based Digital Twins (DT) supports real-time decision-making in manufacturing and logistics systems. This master's thesis presents the design and implementation of a DES-based DT developed within the Intelligent Cyber-Physical Systems (I-CPS) Laboratory at Polytechnique Montréal. The objective is to develop a real-time synchronization mechanism that enables adaptive management of maintenance and charging operations for Autonomous Mobile Robots (AMRs) in a smart warehouse environment.

The proposed framework combines two synchronization mechanisms: Time-Triggered (TT) synchronization for control actions and Event-Triggered (ET) synchronization for monitoring and state updates. The architecture, aligned with the ISO 23247 standard, embeds a DES model as the decision-making kernel to maintain temporal consistency between the physical system and its digital counterpart.

Experimental results obtained in the I-CPS Laboratory show that the selective synchronization mechanism reduces communication latency between the physical and digital systems, maintaining operational continuity. This work defines a methodology for developing DES-based DT that support real-time operational decision-making in industrial systems.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vi
LISTE DES TABLEAUX	ix
LISTE DES FIGURES	x
LISTE DES SIGLES ET ABRÉVIATIONS	xii
LISTE DES ANNEXES	xiv
CHAPITRE 1 INTRODUCTION	1
1.1 Contexte de recherche	1
1.2 Problématique	2
1.3 Question de recherche	4
1.4 Suppositions générales	4
1.5 Objectifs	5
1.6 Plan du mémoire	5
CHAPITRE 2 REVUE DE LITTÉRATURE	6
2.1 Méthodologie de recherche	6
2.2 De la simulation au jumeau numérique : définitions des concepts et clarification des frontières	9
2.2.1 Définitions fondamentales	10
2.2.2 Relation entre jumeau numérique et simulation	13
2.3 Connectivité dans les jumeaux numériques	14
2.3.1 Protocoles de communication industriels et d'échange de données . .	15
2.3.2 Synchronisation et gestion du temps réel	17
2.3.3 Représentation numérique des jumeaux numériques : Système Virtuel	19
2.4 Cadres normatifs des jumeaux numériques : ISO 23247	23
2.5 Lacunes identifiées et positionnement de cette étude	24

CHAPITRE 3	ARTICLE 1: REAL-TIME DATA SYNCHRONIZATION MECHANISMS FOR DISCRETE EVENT SIMULATION-BASED DIGITAL TWINS	27
3.1	Introduction	28
3.2	Related work	29
3.2.1	DES-based DT approach	30
3.2.2	Standardization efforts for DT integration	31
3.2.3	DT synchronization	32
3.2.4	Key insights from this literature review	34
3.3	Conceptual solution design	35
3.3.1	Architecture of the Proposed Synchronization Framework	35
3.3.2	Design Logic of Event and Time Triggered Synchronization Points	37
3.4	Use case of the proposed DES-Based DT Framework	39
3.4.1	Use case at the I-CPS laboratory	39
3.4.2	System Architecture	41
3.4.3	Data Representation and Modeling of Observable Manufacturing Elements	43
3.4.4	Implementation of Event-Triggered Synchronization	44
3.4.5	Implementation of Time-Triggered Synchronization	46
3.5	Experimentation and Result	49
3.5.1	Experimental Assessment of Synchronization Performance	49
3.5.2	Experimental Evaluation of the Number of Topics and Update Frequency on Latency	51
3.5.3	Experiments under selective TT and ET mechanism	54
3.6	Conclusion	56
CHAPITRE 4	DISCUSSION GÉNÉRALE	58
CHAPITRE 5	CONCLUSION	60
5.1	Synthèse des travaux	60
5.2	Limitations de la solution proposée	61
5.3	Améliorations futures	61
RÉFÉRENCES		62
ANNEXES		68

LISTE DES TABLEAUX

Tableau 2.1	Plan de concepts par axes de recherche	7
Tableau 2.2	État de l’art des protocoles, systèmes virtuels et mesures temporelles des jumeaux numériques	22
Tableau 2.3	Synthèse de la revue de littérature sur les caractéristiques fonctionnelles, les applications et les normes des jumeaux numériques	26
Table 3.1	MQTT Topics Defining the Synchronization Points of the DES-based DT	50
Table 3.2	Two-way ANOVA results for latency	52
Tableau A.1	Résumé du cas d’usage conforme à la norme ISO 23247.	68

LISTE DES FIGURES

Figure 1.1	le laboratoire des systèmes cyber-physiques intelligents (SCP-I) à Polytechnique Montréal	2
Figure 2.1	Évolution des publications sur les jumeaux numériques sur SCOPUS	8
Figure 2.2	Méthodologie de sélection des articles pour la revue de littérature . .	9
Figure 2.3	Évolution du lien entre le système physique et la réplique virtuelle selon les trois niveaux de développement : (a) modèle numérique, (b) ombre numérique et (c) jumeau numérique.	10
Figure 2.4	Cadre technique de jumeau numérique proposé par la norme ISO 23247	24
Figure 3.1	The conceptual design of the proposed synchronization mechanism for the DES-based DT	37
Figure 3.2	The UML sequence diagram illustrating the design logic of the ET and TT synchronization mechanisms within the proposed DES-based DT framework	38
Figure 3.3	(a) The DT user interface showing real-time charge level and motor status of the AMR; (b) The I-CPS laboratory testbed at Polytechnique Montréal (Québec, Canada) with transporter and Dual-Arm Manipulator AMRs, charging, and maintenance stations; (c) The DT user interface of the reception and shipping storage area with a real-time stock-level dashboard; (d) The Physical reception and shipping storage area equipped with proximity sensors for real-time stock monitoring	40
Figure 3.4	The communication architecture of the DES-based DT implemented in the I-CPS Laboratory	42
Figure 3.5	The Entity–Relationship Diagram illustrating the data representation and modeling of the OME within the DES-based DT framework implemented in the I-CPS Laboratory	44
Figure 3.6	The UML sequence diagram illustrating the implementation logic of ET synchronization within the proposed DES-based DT framework .	46
Figure 3.7	The UML sequence diagram of TT synchronization in the proposed DES-based DT framework.	47
Figure 3.8	The UML sequence diagram illustrating the control response and bidirectional data flow between the DT and the physical system. . . .	48

Figure 3.9	The message latency for Topic 1 as additional topics are incrementally activated	51
Figure 3.10	The effect of number of MQTT topics on message latency (mean trend across five topic levels); (b) The effect of update interval (T) on message latency (comparison of $T = 0.01$ s and $T = 0.1$ s); (c) The interaction effect between number of topics and update interval. . . .	54
Figure 3.11	The baseline configuration for all OMEs synchronized through a TT mechanism ($\Delta T = 0.1$ s), latency shown on a logarithmic scale. . . .	55
Figure 3.12	The proposed configuration for topic4 synchronized through a TT mechanism ($\Delta T = 0.1$ s); other OMEs synchronized through ET mechanisms.	56
Figure B.1	Diagramme d'activité UML illustrant le transfert du produit de la machine CNC vers la zone de réception dans le SCP-I	69
Figure C.1	Diagramme d'activité UML du processus de réception, d'identification, d'inspection et d'expédition des produits dans le SCP-I.	70

LISTE DES SIGLES ET ABRÉVIATIONS

ADS	Automation Device Specification
AGV	Automated Guided Vehicle
AMR	Autonomous Mobile Robot
ANOVA	Analysis of Variance
AP	Access Point
API	automate programmable industriel
CAO	conception assistée par ordinateur
CAN bus	Controller Area Network bus
CNC	Computer Numerical Control
CPS	Cyber-Physical System
CSV	Comma-Separated Values
DCDCE	Data Collection and Device Control Entity
DES	Discrete-Event Simulation
DoE	Design of Experiment
DT	Digital Twin
ET	Event-Triggered
I4.0	Industrie 4.0
IE	Industrial Ethernet
IoT	Internet of Things
JN	jumeau numérique
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
NRT	Near Real-Time

OME	Observable Manufacturing Element
OPC UA	Open Platform Communications – Unified Architecture
PPC	Production Planning and Control
PLC	Programmable Logic Controller
PROFINET	Process Field Network
QoS	Quality of Service
QR	Quick Response
RFID	Radio-Frequency Identification
ROS 2	Robot Operating System 2
RT	Real-Time
SBA	simulation à base d’agents
SCP	systèmes cyber-physiques
SCP-I	systèmes cyber-physiques intelligents
SED	simulation à événements discrets
SoC	State of Charge
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol / Internet Protocol
TT	Time-Triggered
UML	Unified Modeling Language
WS	Work Station
XML	eXtensible Markup Language
YAML	Yet Another Markup Language

LISTE DES ANNEXES

Annexe A	CAS D'USAGE	68
Annexe B	PHASE 1 : TRANSFERT DU PRODUIT VERS LA ZONE DE RÉCEPTION	69
Annexe C	PHASE 2 : RÉCEPTION, IDENTIFICATION, INSPECTION ET EXPÉDITION DU PRODUIT	70

CHAPITRE 1 INTRODUCTION

1.1 Contexte de recherche

Le développement rapide du commerce électronique, e-commerce, a transformé les attentes des consommateurs en matière de livraison. Les clients exigent désormais une livraison rapide et conforme aux délais annoncés. Cette évolution impose aux acteurs de la chaîne logistique et aux entrepôts une adaptation afin de répondre à la variabilité de la demande, à la diversité des produits et aux exigences croissantes liées au service [1].

La quatrième révolution industrielle, ou l'Industrie 4.0 (I4.0), favorise l'émergence de solutions connectées telles que les jumeaux numériques (JN), qui permettent de synchroniser en temps réel le système physique et sa représentation virtuelle. Cette synchronisation vise à surveiller les opérations, détecter les anomalies et déclencher automatiquement des actions correctives, afin de préserver la continuité des flux logistiques et de respecter les engagements de livraison.

Ce projet s'inscrit dans le cadre du développement et de l'intégration d'un JN au sein du Laboratoire des systèmes cyber-physiques intelligents (SCP-I) de Polytechnique Montréal, en collaboration avec un partenaire industriel spécialisé dans la maintenance des véhicules électriques. La problématique étudiée concerne la gestion adaptative de la maintenance et de la recharge, dans le but de réduire les temps d'arrêt liés à ces opérations.

Le SCP-I intègre plusieurs acteurs interconnectés qui influencent la réactivité du système logistique, notamment trois robots mobiles autonomes (Autonomous Mobile Robots, AMR), deux de type transporteurs et un manipulateur à bras robotisé. Ces ressources collaborent pour assurer la réception, la manutention et l'expédition des marchandises dans le respect des délais. Dans cette étude, les robots constituent une analogie des véhicules électriques du partenaire industriel, permettant de reproduire, dans un environnement contrôlé, les principes de fonctionnement et de prise de décision associés aux opérations de maintenance et de recharge.

Le laboratoire SCP-I constitue un banc d'essai pour la validation des mécanismes de synchronisation et de prise de décision en temps réel avant leur déploiement dans le cas industriel.



FIGURE 1.1 le laboratoire des systèmes cyber-physiques intelligents (SCP-I) à Polytechnique Montréal

1.2 Problématique

L'intégration des robots dans le secteur industriel n'est pas une avancée récente. Déjà présente lors de la révolution de l'automatisme avec l'Industrie 3.0, cette robotisation s'est accélérée avec l'ère de I4.0, marquée par l'émergence de systèmes connectés et intelligents. Ces systèmes s'appuient sur des technologies telles que les capteurs, les automates programmables industriels (API) et les protocoles de communication. L'I4.0 a transformé la manière dont les systèmes industriels sont pilotés, en instaurant une connexion bidirectionnelle entre le système physique et le système virtuel, ce qu'on appelle jumeau numérique [2, 3].

Dans le contexte de l'I4.0, les entrepôts jouent un rôle clé dans la performance de la chaîne d'approvisionnement. Leur capacité à réagir rapidement aux variations de la demande, à maintenir la disponibilité des ressources et à garantir la continuité des flux logistiques devient déterminante. Le marché impose désormais aux entreprises manufacturières des exigences strictes en matière de temps de réponse, de flexibilité, de production personnalisée et de maîtrise des coûts [4]. Toute interruption ou mauvaise allocation des ressources peut allonger les délais d'exécution, retarder la livraison, et nuire à la compétitivité [5].

Or, les entrepôts automatisés, en particulier ceux reposant sur des AMR, restent confrontés

à des interruptions d'activité dues à deux types de causes principales : les pannes, qui immobilisent les robots et perturbent les flux physiques ; et/ou les cycles de recharge, qui réduisent la disponibilité des ressources.

Ces interruptions, si elles ne sont pas anticipées, planifiées ou compensées dynamiquement, engendrent des déséquilibres dans la répartition des tâches, une surcharge de certains postes et une perte d'efficacité globale [6]. De plus, les méthodes de planification classiques, fondées sur des approches centralisées ou statiques, ne permettent pas d'intégrer en temps réel l'évolution de l'état du système [7].

Face à cette situation, les JN offrent une alternative. En établissant une connexion bidirectionnelle entre le système physique et son modèle virtuel, ils permettent la collecte des données issues du terrain, la mise à jour en temps réel de l'état du système simulé et l'envoi d'instructions au système physique à l'aide de protocoles de communication adaptés [8, 9].

Dans ce cadre, le laboratoire SCP-I de Polytechnique Montréal constitue un environnement de test pour un système cyber-physique (SCP) appliqué à un entrepôt. Il intègre plusieurs composants physiques, notamment des robots mobiles autonomes (Autonomous Mobile Robot, ou AMR), des capteurs, une machine CNC, des dispositifs de communications et des étagères de stockage. Cet environnement est utilisé dans ce travail comme plateforme d'expérimentation pour l'implémentation et l'évaluation d'un JN connecté à un système physique.

Cependant, malgré cette infrastructure, la coordination des ressources reste une problématique. Les décisions d'expédition de robots ne prennent pas en compte leur état de batterie. Les opérations de recharge, lorsqu'elles ne sont pas pilotées par un JN, ne tiennent pas compte des conditions en temps réel relatives à la charge de travail et à la priorité des missions. L'arrêt des robots peut entraîner des retards ou un arrêt du flux logistique dans l'entrepôt.

Dans ce cadre, la problématique soulevée dans ce mémoire porte sur la capacité du JN à gérer ces interruptions. Il s'agit de déterminer si ce type de système permet d'améliorer les décisions liées à la recharge des robots ainsi que la détection et la prise en compte des pannes [10, 11].

L'objectif de ce travail est donc de tester expérimentalement la capacité d'un JN à améliorer la réactivité et la continuité des opérations, dans un contexte de flux physiques pilotés par des AMR. Cette expérimentation repose sur trois capacités essentielles : (1) la synchronisation en temps réel entre le système physique et le système virtuel ; (2) la communication bidirectionnelle entre le JN et les composants du système ; et (3) l'intégration de règles de décision prenant en compte les contraintes opérationnelles (niveau de batterie, disponibilité

des ressources, affectation des missions, signaux de défaut).

Ce travail vise à évaluer dans quelle mesure un JN peut répondre aux contraintes imposées par les exigences actuelles du marché, notamment la rapidité, la flexibilité et la fiabilité dans l'exécution des processus logistiques [12].

1.3 Question de recherche

Dans un entrepôt, la continuité des opérations dépend de la disponibilité des ressources, telles que AMR. Les interruptions causées par des opérations de maintenance ou de recharge non planifiées peuvent affecter la performance globale du système. Dans ce contexte, les JN offrent des capacités de surveillance en temps réel, d'analyse de l'état du système et de déclenchement d'actions prédictives.

Ce travail s'inscrit dans cette perspective et cherche à comprendre dans quelle mesure le JN peut soutenir la prise de décision opérationnelle.

La question de recherche à la base de ce projet est donc la suivante :

Comment la technologie du jumeau numérique contribue-t-elle à la prise de décision adaptative en gestion d'entrepôt, notamment dans le contrôle en temps réel des interruptions dues à la maintenance et à la recharge des robots ?

1.4 Suppositions générales

Dans le cadre de ce travail, les suppositions suivantes sont posées afin de définir les conditions de fonctionnement du système étudié :

- **Supposition 1 — Fonctionnement en continu :**

L'entrepôt est supposé fonctionner en continu, 24 heures sur 24. Cette configuration implique un besoin permanent de planification , incluant la gestion d'expédition des robots , les cycles de recharge et les interventions de maintenance.

- **Supposition 2 — Seuil de recharge :**

Les robots ne quittent leur station de recharge que lorsque le niveau de batterie, défini comme le pourcentage d'énergie restante par rapport à la capacité maximale état de charge (State of Charge, ou SoC), atteint au moins 20 %, conformément aux recommandations du fabricant.

- **Supposition 3 — Stabilité du réseau Internet local :**

La connexion Internet est présumée stable, assurant la transmission des données en temps réel et l'envoi des commandes aux robots.

1.5 Objectifs

L'objectif de ce travail est de réduire les temps d'arrêt dans le laboratoire SCP-I de Polytechnique Montréal en intégrant un mécanisme de synchronisation en temps réel au sein d'un jumeau numérique pour la gestion des ressources robotiques et des opérations d'entrepôt. Cette intégration repose sur le suivi en temps réel des flux physiques et des flux de données afin de piloter les ressources, coordonner les opérations et déclencher des actions adaptatives. Il s'agit également d'évaluer dans quelle mesure ce JN permet la gestion des arrêts de service en tenant compte des différentes composantes du système : robots, machines et capteurs.

Pour atteindre cet objectif, les sous-objectifs suivants ont été définis :

1. Piloter les cycles de recharge, les opérations de maintenance et les missions d'expédition des robots en s'appuyant sur les données du système en temps réel (niveau de batterie, signaux de défaillance), afin d'assurer la continuité des opérations.
2. Superviser l'état global du système à l'aide du JN (état des équipements, niveau de stock, occupation des postes) dans les décisions de gestion des ressources robotiques.

1.6 Plan du mémoire

Ce mémoire est structuré en quatre chapitres. Le chapitre 1 présente le contexte de recherche, la problématique, la question de recherche, les suppositions et les objectifs. Le chapitre 2 est consacré à la revue de littérature. Il expose les concepts fondamentaux de ce travail, tels que les JN, la simulation et les SCP, et introduit le cadre normatif de référence, notamment la norme ISO 23247. Ce chapitre permet également d'identifier les limites des travaux existants et de situer la contribution de cette recherche. Le chapitre 3 présente un article scientifique issu de ce travail de recherche. Enfin, le chapitre 4 conclut le mémoire en synthétisant les contributions, en précisant les limites et en ouvrant sur des perspectives de recherche futures.

CHAPITRE 2 REVUE DE LITTÉRATURE

Ce chapitre présente la revue de littérature menée dans le cadre de ce travail. Il commence par une description de la méthode utilisée pour formuler les requêtes et sélectionner les publications en lien avec la question de recherche. Ensuite, une analyse des publications permet d'identifier les limites des travaux existants et les lacunes de l'état de l'art. Enfin, le chapitre précise le positionnement de l'étude par rapport aux travaux existants et expose la contribution de ce travail à la littérature.

2.1 Méthodologie de recherche

Cette section présente la méthodologie adoptée pour identifier les lacunes dans la littérature scientifique. La base de données SCOPUS a été exploitée comme source principale.

La première étape a consisté à élaborer un plan de concepts structuré selon trois axes : les JN, les robots et l'amélioration des processus. Le premier axe concerne la technologie étudiée, à savoir les JN, qui établissent un lien entre le système physique et sa représentation virtuelle. Le second axe porte sur les robots, en particulier les AMR et les bras robotisés, qui constituent les ressources physiques à piloter dans l'environnement d'étude. Le troisième axe est centré sur les processus opérationnels, avec une attention particulière à la maintenance, à la recharge, à la coordination des opérations et à la gestion des temps d'arrêt.

Les mots-clés de chaque axe ont été combinés à l'aide de l'opérateur logique OU, afin de prendre en compte la diversité terminologique utilisée dans les publications scientifiques. Ensuite, les trois axes ont été reliés entre eux à l'aide de l'opérateur logique ET, dans le but de cibler les études qui abordent simultanément les trois dimensions retenues. Cette structuration permet de filtrer les travaux qui exploitent les JN pour piloter ou superviser des ressources robotiques dans un contexte de gestion des interruptions et de coordination des flux. Ce plan de concept est présenté dans le **Tableau 2.1** ci-dessous.

TABLEAU 2.1 Plan de concepts par axes de recherche

Jumeaux Numériques	Robots	Amélioration des processus
Digital twin*	AMR	Downtime
Digital shadow	AGV	Failure
Digital replica	Autonomous mobile robot	Repair
Digital model	Automated guided vehicle	Maintenance
Digital representation	ARM robot	Predictive maintenance
		Adaptive maintenance
		Resource allocation
		Process improvement
		Warehouse
		Smart warehouse
		Inventory
		Simulation
		Agent-based
		Discrete event
		Lead time
		Reduce
		Minimize delay
		Cyber physical system
		CPS
		Cyber physical systems
		Cyber-physical
		Dispatch*
		Planning
		Scheduling
		Battery
		Charging

* : Caractère de troncature afin d'inclure les variantes d'un mot.

Une série de requêtes a été formulée dans SCOPUS à partir de ce plan de concepts. Ces requêtes ont été exécutées entre janvier et mai 2025. Cela a permis d'identifier un ensemble

initial de 117 documents. Une première exclusion a été appliquée pour retirer les documents de type « revue de conférence », qui ne contiennent que les titres des communications sans contenu scientifique complet. Cette étape a réduit le corpus à 105 documents.

Une deuxième étape a consisté à limiter la sélection aux articles rédigés en anglais et publiés entre 2016 et 2025, réduisant le corpus à 101 articles. Ce choix est motivé par l'évolution récente et marquée de l'intérêt pour le concept de JN. Comme l'illustre la Figure 2.1, les publications contenant les termes « digital twin » dans le titre, le résumé ou les mots-clés ont connu une croissance exponentielle à partir de 2016. Cela confirme le choix de cette période pour l'analyse bibliographique.

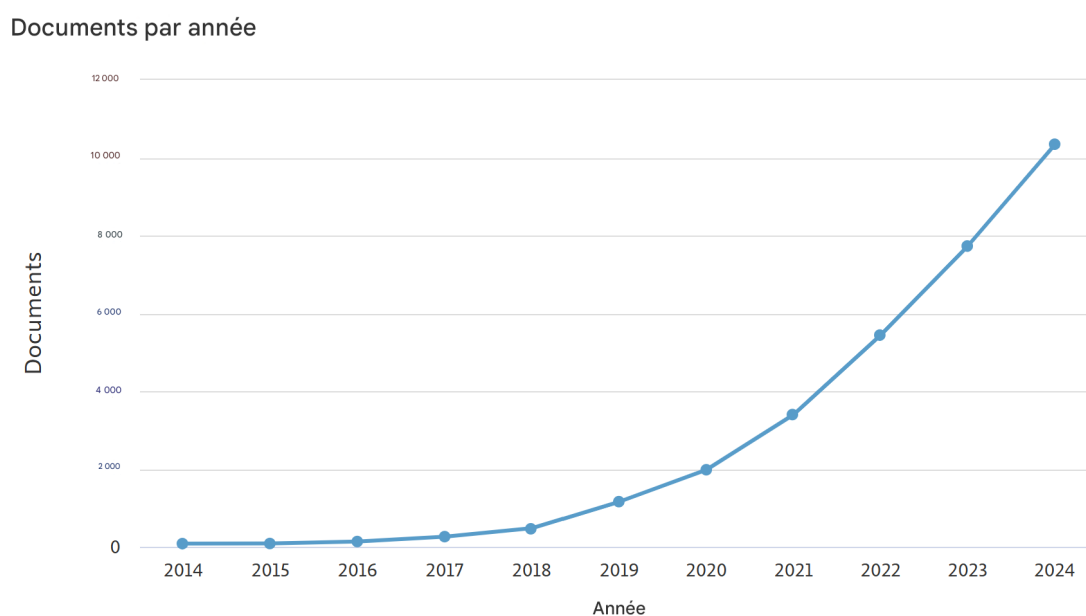


FIGURE 2.1 Évolution des publications sur les jumeaux numériques sur SCOPUS

Une troisième exclusion a été réalisée pour écarter les articles issus de domaines non liés au champ d'étude, tels que la médecine, la biologie ou les sciences sociales. Cette étape a conduit à un sous-ensemble de 87 articles. Un filtrage supplémentaire a ensuite été effectué à partir de mots-clés d'auteurs considérés comme non pertinents par rapport à l'objet de recherche (par exemple : robots agricoles, robots collaboratifs, terminaux portuaires ou ferroviaires). Ce filtrage a ramené le nombre d'articles à 78.

Les mots-clés indexés par SCOPUS ont ensuite été utilisés pour écarter des thématiques éloignées des objectifs de l'étude, comme l'apprentissage profond, la 5G, ou l'apprentissage par renforcement, ce qui a réduit l'échantillon à 43 articles. Après une lecture des résumés,

22 articles ont été retenus. Enfin, après lecture complète, seuls 9 articles ont été retenus car ils sont en lien avec la problématique et les objectifs de ce travail.

À ces 9 articles s'ajoutent 15 articles identifiés par d'autres moyens, notamment par suivi des références bibliographiques ou par recommandations de la direction de recherche. Ainsi, le corpus final analysé dans cette étude est constitué de 24 articles. La méthodologie de sélection des articles est présentée dans la Figure 2.2.

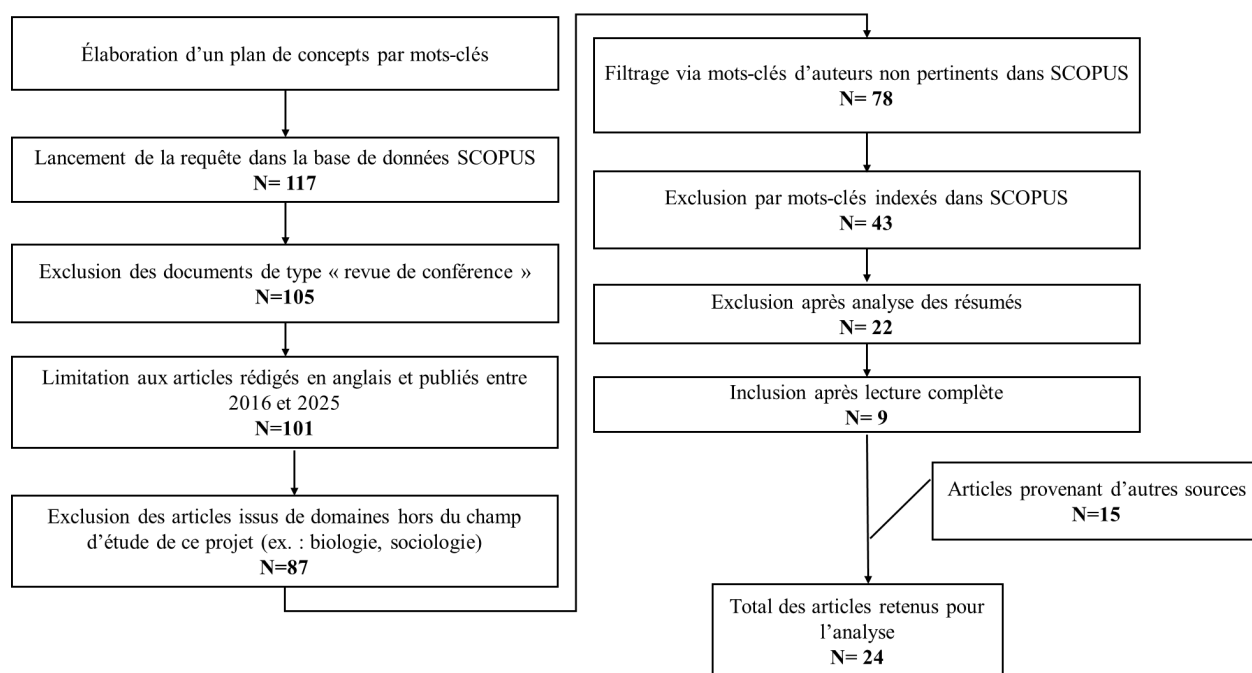


FIGURE 2.2 Méthodologie de sélection des articles pour la revue de littérature

2.2 De la simulation au jumeau numérique : définitions des concepts et clarification des frontières

Avec l'émergence de l'I4.0, plusieurs approches se sont imposées pour représenter et piloter les systèmes manufacturiers et logistiques : la simulation, les systèmes cyber-physiques (SCP) et les JN. Ces notions, bien que complémentaires, présentent des finalités et des niveaux d'intégration différents. Cette section vise à en préciser les définitions et à clarifier leurs relations, afin d'établir les frontières qui distinguent un modèle de simulation d'un JN dans le cadre des SCP.

2.2.1 Définitions fondamentales

Jumeaux Numériques :

Les JN ont été présentés pour la première fois en 1960 par la NASA, dans le cadre du programme Apollo, afin de disposer de répliques physiques et virtuelles permettant de diagnostiquer à distance les engins spatiaux [13]. Depuis, de nombreuses définitions ont été proposées dans la littérature, souvent divergentes selon les disciplines et les contextes d'application. Toutefois, les travaux récents tendent à converger vers un schéma conceptuel commun distinguant trois niveaux de développement.

Le modèle numérique (Digital Model) est une réplique virtuelle d'un système physique permettant d'en reproduire les comportements, mais sans connexion automatique avec celui-ci ; la mise à jour des données s'effectue alors manuellement. L'ombre numérique (Digital Shadow) correspond à un modèle numérique connecté de manière unidirectionnelle, du système physique vers le système virtuel, sans rétroaction inverse. Enfin, le jumeau numérique (Digital Twin) désigne un modèle numérique reposant sur une connexion bidirectionnelle avec le système physique, permettant à la fois la mise à jour en temps réel et l'envoi de commandes ou d'ajustements vers celui-ci [12, 14].

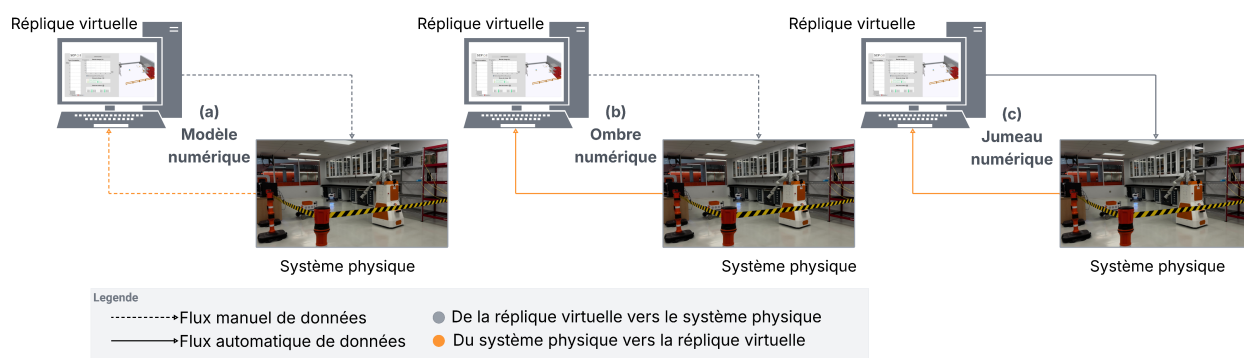


FIGURE 2.3 Évolution du lien entre le système physique et la réplique virtuelle selon les trois niveaux de développement : (a) modèle numérique, (b) ombre numérique et (c) jumeau numérique.

Une autre définition met l'accent sur les capacités opérationnelles nécessaires pour différencier un JN d'une simple simulation ou d'une émulation. Dans un premier temps, il s'agit de disposer d'une représentation numérique fidèle du système physique, construite à partir de données et/ou d'algorithmes. Cette représentation doit être capable de reproduire les résultats du système réel à partir d'entrées issues de celui-ci, tout en pouvant fonctionner de manière partiellement indépendante. Au niveau le plus avancé, le JN assure une synchronisation

complète et un échange bidirectionnel de données entre le système physique et sa réplique virtuelle, permettant l'intégration de la prise de décision en boucle fermée et en temps réel [15].

D'autres auteurs complètent cette approche en soulignant l'importance de la synchronisation en temps réel avec le système physique [6]. Dans cette optique, le JN joue un double rôle : d'une part, observer ou simuler l'état du système physique afin de fournir des données pour la visualisation et l'analyse ; d'autre part, interagir avec le système physique en envoyant des commandes afin de synchroniser et déclencher les actions appropriées [16].

Les applications des JN s'étendent de la modélisation d'un composant ou d'un équipement isolé [11, 17] à la représentation intégrée d'un système complet ou d'un processus [12]. Elles couvrent divers secteurs d'activité, tels que la production manufacturière, l'énergie, l'automobile, la santé, la robotique, l'I4.0, la logistique, l'aviation et la production manufacturière étant le domaine le plus fréquemment documenté [15]. Selon le contexte, le JN est mobilisé pour l'optimisation des processus, la maintenance prédictive, la formation, la simulation de scénarios opérationnels ou l'intégration de nouvelles configurations dans un environnement contrôlé [12].

Enfin, [5, 12] proposent de définir le JN comme un système constitué de deux entités interconnectées : un système physique et un système virtuel. Le premier regroupe l'ensemble des dispositifs intelligents, tels que les capteurs, les API, les actionneurs, les machines et les robots, qui assurent la collecte et la transmission des données en temps réel. Le second, fondé sur des modèles numériques et des systèmes d'information, reproduit fidèlement le comportement et l'état du système physique. Ces deux entités interagissent via une architecture de communication bidirectionnelle, garantissant une synchronisation des données.

Les systèmes cyber-physiques (SCP)

Ce travail s'inscrit dans le cadre du Laboratoire des SCP-I de Polytechnique Montréal. Il est donc pertinent de commencer par définir ce qu'est un système cyber-physique. Ce concept a été introduit en 2006 par la fondation nationale pour la science pour désigner des systèmes dont la description dépasse la terminologie traditionnelle des technologies de l'information [18].

Le terme a rapidement retenu l'attention du secteur industriel et de la recherche. Les SCP s'inscrivent dans le cadre de l'I4.0 et repose sur la connectivité et l'intégration de systèmes capables de communiquer et d'interagir en réseau. Un SCP associe le monde physique,

constitué de machines, d'équipements, de capteurs, d'actionneurs et de robots, et le monde cyber, qui regroupe la couche numérique assurant la collecte, le transfert, le traitement, l'analyse et l'exploitation des données issues du monde physique. Ces deux environnements sont reliés par des flux de données et des boucles de rétroaction, permettant la détection en temps réel, le retour d'information et le contrôle.

Selon [18], il existe une corrélation entre SCP et JN. Les SCP fournissent l'infrastructure physique et la connectivité entre les composants, tandis que les JN en constituent la réplique virtuelle. Les données collectées à partir du SCP sont exploitées par les JN pour analyser, améliorer, contrôler et piloter la prise de décision sur le fonctionnement du système.

Simulation à événements discrets (SED)

La simulation à événements discrets (SED) est l'une des techniques les plus répandues en recherche opérationnelle pour résoudre des problématiques de planification, d'ordonnancement, de logistique ou de gestion des files d'attente. Elle se distingue par sa capacité à tester différents scénarios et à évaluer leurs effets avant leur mise en œuvre réelle [10].

Elle repose sur la modélisation dynamique d'un système dont l'état évolue uniquement aux instants où se produisent des événements spécifiques [19]. Dans ce type de simulation, le modèle virtuel est construit à partir de données historiques collectées sur le système physique, lesquelles sont ensuite intégrées dans le logiciel de simulation. Ces données servent à comprendre et reproduire le comportement du système, à vérifier et valider le modèle, puis à mener des analyses permettant d'observer l'impact de modifications de paramètres ou de configurations opérationnelles [15].

La SED permet ainsi de représenter la structure, les éléments, ainsi que leurs caractéristiques, comportements et fonctions, qu'il s'agisse d'un système physique ou abstrait. Elle offre la possibilité de prévoir le comportement futur du système dans des conditions nouvelles et d'identifier des configurations optimales sur la base des résultats obtenus. Elle consiste en un processus informatique décrivant un système dont les changements d'état se manifestent à un nombre dénombrable de points dans le temps, chacun correspondant à un événement [12]. Cette approche est largement utilisée pour analyser, planifier et optimiser des systèmes de production, de services ou de logistique [15].

2.2.2 Relation entre jumeau numérique et simulation

La simulation recouvre plusieurs modèles, comme SED ou la simulation à base d'agents (SBA). Le choix du modèle dépend des objectifs de l'étude. Par exemple, la SBA repose sur la modélisation du comportement d'agents autonomes, tels que des individus, des machines ou des groupes de processus, interagissant entre eux et avec leur environnement [10]. La SED, quant à elle, est principalement utilisée pour représenter et analyser des flux ou des processus plutôt que des comportements d'agents [12].

Dans la revue systématique de [10], 92 % des publications examinées mobilisent la SED, contre seulement 5,3 % pour la simulation à base d'agents (SBA) et 2,7 % qui combinent les deux approches dans un cadre de simulation hybride.

Pour l'étude de processus et de flux, la SED est généralement plus adaptée que la SBA, qui se concentre davantage sur la modélisation du comportement d'agents autonomes [10].

Avec l'essor de l'I4.0, la distinction entre simulation et JN fait l'objet d'une confusion dans la littérature. Bien que ces deux notions présentent des points communs, elles se différencient par leurs finalités et leurs capacités. Une classification évoquée par [15] permet de les distinguer selon leurs niveaux fonctionnels. La simulation se caractérise par quatre niveaux (4S) : Modeling (S1), Analyzing (S2), Predicting (S3) et Prescribing (S4). Ces niveaux correspondent respectivement à : (S1) la modélisation d'un système à partir de données historiques, (S2) l'analyse par expérimentation et variation des paramètres, (S3) la prédiction du comportement futur dans des scénarios alternatifs et (S4) la prescription d'actions ou de configurations optimales [15].

Le jumeau numérique (JN) repose sur quatre capacités fondamentales (4R) : (R1) la représentation, soit la modélisation numérique fidèle du système physique à partir de données et d'algorithmes ; (R2) la réplique, qui permet de reproduire virtuellement le comportement du système réel et d'obtenir des résultats équivalents à partir des mêmes entrées ; (R3) la réalité, correspondant à la capacité du modèle à générer des prédictions de manière autonome sur la base d'entrées similaires ; et (R4) la relation, assurant la synchronisation continue et l'échange bidirectionnel de données entre le système physique et son jumeau virtuel [15,20,21].

Selon [15], de nombreux travaux se présentant comme des JN ne dépassent pas les premières capacités (R1 ou R2) et relèvent, en réalité, davantage de la simulation. Certains auteurs [5, 13,22] associent une simulation à quelques éléments d'un JN, mais sans atteindre la connexion continue et l'échange bidirectionnel caractéristiques du niveau R4. Dans plusieurs cas, il s'agit simplement de modèles de simulation qualifiés à tort de JN, ne comportant ni collecte de données en temps réel ni connexion bidirectionnelle entre les mondes physique et virtuel.

Cette étude de [15], portant sur 120 articles, a révélé que 33 % d'entre eux s'identifiaient comme JN, alors qu'ils ne présentaient aucune capacité propre à ce concept et se limitaient à de simples simulations. Par ailleurs, si l'utilisation du temps réel ou de la réplication (R2,R3) est relativement fréquente, la bidirectionnalité complète reste rare.

La SED et le JN se fondent sur des approches de modélisation comparables puisqu'ils reposent tous deux sur la modélisation numérique d'un système physique en vue d'en analyser le comportement et d'optimiser ses performances. Dans certains cas, la SED constitue même le cœur du modèle virtuel d'un JN. Toutefois, leur logique d'exploitation diffère : la SED traditionnelle fonctionne généralement hors ligne, à partir de données historiques, et ne nécessite pas une connexion continue avec le système réel, alors que le JN intègre des flux de données en temps réel, une synchronisation permanente entre mondes physique et virtuel, et un échange bidirectionnel permettant de transmettre des instructions au système physique [12, 15].

Selon l'analyse bibliométrique et de contenu menée par [12], appliquée au contexte de la logistique d'entrepôt, l'intégration conjointe SED-JN est encore limitée : seuls 12 % des 50 articles examinés par [12] abordent explicitement leur couplage, contre 48 % centrés uniquement sur la SED et 40 % exclusivement sur le JN. L'étude souligne également que les approches combinées présentent un fort potentiel, notamment pour améliorer la prise de décision en temps réel, mais que ce champ demeure encore peu exploré dans la littérature scientifique.

En conclusion, le JN et la SED sont deux concepts distincts, mais pouvant être intégrés dans un même cadre applicatif. La SED peut contribuer à la modélisation virtuelle d'un JN, à condition que ce dernier intègre les éléments caractéristiques qui le définissent : collecte et exploitation de données en temps réel, synchronisation continue entre systèmes physique et virtuel, ainsi qu'un échange bidirectionnel permettant d'agir sur le système réel. En l'absence de ces éléments, le système relève d'une simulation plutôt que d'un JN. Il convient donc, dans les travaux scientifiques comme dans les applications pratiques, de distinguer les cas où la SED est utilisée de manière indépendante de ceux où elle s'inscrit dans un JN.

2.3 Connectivité dans les jumeaux numériques

L'I4.0 se distingue des précédentes par la place centrale qu'occupe la connectivité. Grâce aux avancées des technologies de communication et à l'essor de l'Internet, il est désormais possible d'échanger en temps réel des données, des informations et des commandes entre systèmes physiques et virtuels. Cette capacité d'interconnexion constitue un fondement essentiel des

JN, en permettant une interaction continue, une mise à jour dynamique et, dans certains cas, un pilotage direct du système physique depuis sa réplique numérique [11].

Dans cette section, nous examinerons d’abord les protocoles de communication industriels. Nous aborderons ensuite les différentes formes de représentation numérique pouvant constituer le système virtuel, qu’il s’agisse de SED, de SBA, d’environnements immersifs ou de modèles issus de la conception assistée par ordinateur (CAO). Enfin, nous traiterons des enjeux liés à la synchronisation et à la gestion du temps réel, en mettant l’accent sur les mécanismes d’actualisation et la cohérence entre les données des mondes physique et virtuel.

2.3.1 Protocoles de communication industriels et d’échange de données

La mise en œuvre d’un JN repose sur l’utilisation de protocoles de communication industrielle dont les caractéristiques varient en fonction des besoins en rapidité de transmission, en sécurité, en compatibilité et en gestion de volumes de données [17]. Le choix du protocole dépend du contexte applicatif ainsi que des contraintes matérielles et logicielles du système. Un critère déterminant réside dans la capacité de ces protocoles à maintenir une synchronisation entre le système physique et sa réplique numérique [7]. L’absence d’un tel mécanisme peut entraîner une augmentation du temps d’acquisition et de traitement des données, ainsi qu’une perte de cohérence entre les deux représentations, compromettant la capacité du JN à refléter fidèlement l’état du système [6]. La continuité des échanges de données deviennent alors essentielles pour assurer la transmission et permettre l’ajustement continu des règles opérationnelles. Ce principe est notamment inscrit dans certaines normes industrielles, comme l’ISO 3691-4 :2020, qui impose des exigences de connectivité afin de garantir la sécurité et la productivité des systèmes [23].

Comme l’illustre le **Tableau 2.2** ci-dessous, la littérature présente plusieurs protocoles de communication pour répondre à ces besoins, parmi lesquels le protocole de Transport de télémessure avec mise en file d’attente de messages (Message Queuing Telemetry Transport, ou MQTT) et le protocole d’Architecture Unifiée des Plateformes de Communication Ouverte (Open Platform Communications Unified Architecture, ou OPC-UA) figurent parmi les plus utilisés. En particulier, MQTT est un protocole léger de messagerie fondé sur un modèle publier/s’abonner, conçu pour la communication machine à machine et l’Internet des objets en s’appuyant sur (Transmission Control Protocol/Internet Protocol, ou TCP/IP) [9, 24]. Il se caractérise par une faible surcharge de transport et une capacité à fonctionner avec des appareils disposant de ressources limitées ou dans des environnements à bande passante restreinte. Ces caractéristiques en font un protocole adapté aux besoins des JN, qui reposent sur la mise à jour en temps réel et la synchronisation bidirectionnelle avec le système physique

[9]. Dans ce cadre, MQTT permet l'échange de données entre le système physique et son JN, tout en respectant les exigences de communication sécurisée [24, 25]. MQTT requiert peu de bande passante, présente une latence réduite et offre une fiabilité accrue [9, 24]. La communication repose sur un courtier de messages (broker), qui gère l'authentification des clients, le traitement des abonnements et la transmission des messages publiés aux abonnés. Les clients publient des messages vers des sujets (topics) définis et peuvent s'abonner à un ou plusieurs topics pour recevoir les données correspondantes.

MQTT définit trois niveaux de qualité de service (Quality of Service, ou QoS). Le niveau QoS 0, dit au plus une fois, correspond à un envoi sans accusé de réception, ce qui implique un risque de perte en cas de défaillance du réseau de communication entre le client et le broker MQTT. Le niveau QoS 1, appelé au moins une fois, prévoit un accusé de réception et la republication du message en l'absence de confirmation, garantissant ainsi sa réception mais avec la possibilité de doublons. Enfin, le niveau QoS 2, dit exactement une fois, repose sur un mécanisme d'échanges croisés permettant d'éviter toute perte ou duplication, mais au prix d'une charge plus élevée en termes de calcul, de mémoire et de bande passante, ce qui peut ralentir le traitement global des messages [9, 24, 26].

Le protocole OPC-UA, quant à lui, est une norme de communication industrielle développée par l'OPC Foundation. Il repose sur une architecture orientée services permettant l'échange de données de manière sécurisée et indépendante de la plateforme utilisée [27]. OPC-UA prend en charge la modélisation des données et l'interopérabilité entre différents systèmes, ce qui en fait un protocole adopté pour connecter et intégrer des équipements, capteurs, API et applications logicielles dans des environnements industriels. Dans le contexte des JN, OPC-UA permet la transmission structurée et normalisée des données entre le système physique et sa représentation virtuelle [8].

D'autres protocoles sont également présents dans la littérature. Kafka, conçu pour le traitement de données massives, repose sur un modèle publier/s'abonner distribué où les producteurs diffusent des messages pouvant être lus par plusieurs consommateurs [7]. WebSocket permet, quant à lui, d'établir une communication bidirectionnelle persistante entre un client et un serveur sur une seule connexion, assurant un échange asynchrone où chaque partie peut envoyer des messages à tout moment tant que la liaison est ouverte [3, 26, 28].

Ces protocoles s'appuient sur des solutions de stockage adaptées à la nature et au format des données transmises. InfluxDB est une base de données orientée séries temporelles, adaptée au stockage et à l'interrogation de données temporelles, notamment pour la surveillance et l'analyse en continu [2, 29]. MongoDB est une base orientée documents, permettant de gérer

des données non structurées ou semi-structurées issues de capteurs, de flux de l'Internet des objets (Internet of Things, ou IoT) ou de simulations [7, 30]. Les bases de données relationnelles utilisant le langage de requête structuré (Structured Query Language, ou SQL) restent largement employées pour la gestion de données structurées [3, 4, 14, 22].

Dans le cadre de ce projet, plusieurs protocoles et outils sont mobilisés. Le portail d'automatisation totalement intégrée de Siemens (Totally Integrated Automation Portal, ou TIA Portal), est utilisé pour créer et gérer les blocs de données de l'API SIMATIC S7-1500. Ces blocs constituent des zones mémoire structurées servant au stockage et à l'échange de variables entre les capteurs, les actionneurs et les applications externes [31]. PROFINET, protocole Ethernet industriel (Process Field Network, ou PROFINET), assure la communication entre l'API et les dispositifs connectés tels que les modules d'entrées/sorties, lecteurs à identification par radiofréquence (Radio Frequency Identification, ou RFID), capteurs ou autres contrôleurs [32]. Le système d'exploitation pour la robotique (Robot Operating System 2, ou ROS2) est un ensemble de bibliothèques et d'outils libres (*open source*) destiné au développement d'applications robotiques. Il repose sur le protocole DDS service de distribution de données (Data Distribution Service, ou DDS) et un modèle publier/s'abonner pour l'échange de données entre nœuds logiciels répartis sur un ou plusieurs systèmes [33]. Enfin, le protocole de réseau de contrôleurs (Controller Area Network, ou CAN bus), utilisé dans les systèmes embarqués, permet des échanges robustes jusqu'à 1 Mbit/s, avec une forte immunité aux interférences et des capacités d'auto-diagnostic [34].

2.3.2 Synchronisation et gestion du temps réel

La synchronisation constitue une capacité essentielle au fonctionnement du JN, car elle garantit l'alignement temporel entre le système physique et sa représentation virtuelle [10]. Elle peut être définie comme le mécanisme assurant une correspondance continue des états entre les modèles virtuels et les entités physiques [6]. En l'absence d'un tel mécanisme, des délais de mise à jour apparaissent entre le réel et le virtuel, compromettant la fidélité temporelle et la qualité de la représentation du processus [16].

Parmi les paramètres mesurables de la performance de la synchronisation en temps réel figure la latence, définie comme le temps nécessaire pour transmettre une donnée entre une source et une destination [26]. Elle peut aussi être considérée comme le délai entre l'envoi et la réception d'un message, dont la valeur dépend du protocole de communication, de la configuration du réseau et de la charge de traitement [35]. Selon [6], une latence élevée réduit la capacité du JN à maintenir l'alignement avec l'état du système physique .

Ainsi, au-delà de ces mesures temporelles , il convient de préciser le fonctionnement pratique

de la synchronisation afin de comprendre comment elle s'opère entre le système réel et sa représentation virtuelle.

En effet, la synchronisation en temps réel correspond au processus par lequel le JN s'aligne continuellement sur le système physique afin d'éviter toute déviation ou erreur. Lorsqu'un événement survient dans le monde réel, il déclenche instantanément un événement correspondant dans le modèle virtuel, qui ajuste alors son état en conséquence. Deux cas de figure sont généralement observés : (i) si le système physique prend de l'avance, le modèle virtuel doit rattraper cet écart en s'actualisant immédiatement ; (ii) si le système physique est en retard, le processus virtuel est suspendu jusqu'à réception de l'information de fin de tâche, ce qui permet de maintenir la correspondance temporelle et l'exactitude de la représentation entre le réel et le virtuel [6].

Sur la base de ce mécanisme, [10] distingue deux approches principales de synchronisation : le temps réel (Real-Time, ou RT) et le quasi-temps réel (Near Real-Time, ou NRT). Le RT correspond à une mise à jour immédiate, déclenchée par les événements physiques eux-mêmes, ce qui permet un suivi continu du système physique. Cette approche est requise lorsque le JN doit participer à la prise de décision instantanée ou au contrôle autonome des opérations. Le NRT, quant à lui, repose sur des mises à jour périodiques planifiées à intervalles réguliers. Cette méthode, largement adoptée dans l'industrie, réduit les exigences en termes de fréquence d'acquisition, de bande passante et de puissance de calcul, tout en restant adaptée à de nombreuses tâches de supervision et d'optimisation. Selon [10], 72 % des modèles étudiés fonctionnent en NRT plutôt qu'en RT, principalement en raison de contraintes liées aux protocoles de communication, aux capacités des infrastructures IoT et aux limites de traitement des systèmes.

Par ailleurs, certains auteurs [22, 36] qualifient leurs systèmes de temps réel alors qu'ils décrivent en réalité des mécanismes assimilables à des initialisations. Dans ces cas, la mise à jour du modèle virtuel est effectuée uniquement à des moments prédéfinis, par exemple au début ou à la fin d'un quart de travail (Shift). Il s'agit alors d'une logique périodique où la cohérence entre le système physique et sa réplique virtuelle n'est vérifiée qu'à des intervalles réguliers, plutôt que maintenue en continu.

D'autres travaux signalent une approche encore plus restrictive, dans laquelle l'alignement entre le système réel et le JN se limite à une phase initiale : le modèle virtuel est synchronisé uniquement au démarrage, sans mécanisme de mise à jour dynamique durant le processus [14, 22]. Dans ces cas, le JN correspond davantage à une initialisation ponctuelle plutôt qu'une synchronisation en temps réel, le modèle conservant un état statique ou actualisé de manière limitée, sans suivi permanent des évolutions du système physique.

Comme l'illustre le **Tableau 2.2**, certains auteurs qualifient leurs systèmes de « temps réel » sans toutefois fournir de mesures temporelles quantitatives entre les mondes physique et virtuel [2, 6]. Seuls quelques travaux présentent des valeurs chiffrées : [16] rapporte un temps d'actualisation d'environ 200 ms, c'est-à-dire l'intervalle entre deux mises à jour successives du modèle virtuel. [17] mentionne des latences compris entre 11 et 541 ms, avec une moyenne de 54 ms ; [25] rapporte une exécution périodique de la détection d'obstacles toutes les 1000 ms ; et [4] décrit un intervalle de mise à jour de 100 ms.

Cette observation est confirmée par la revue systématique [10], qui indique que la quantification des délais de communication et de mise à jour demeure rarement abordée.

L'état de la littérature révèle ainsi que la synchronisation en temps réel recouvre en pratique une variété d'approches, allant de l'initialisation ponctuelle à des mises à jour périodiques, notamment celles liées aux quarts de travail, tandis que les preuves quantitatives de performance temporelle restent limitées. Cette situation met en évidence la nécessité de préciser et de mieux définir la synchronisation en temps réel pour les JN.

2.3.3 Représentation numérique des jumeaux numériques : Système Virtuel

La représentation numérique constitue l'un des fondements d'un JN niveau R1 selon [15], elle correspond à la création d'un modèle virtuel du système physique, défini à partir de données et d'algorithmes. Dans la littérature, différentes catégories d'environnements sont mobilisées pour représenter le système virtuel variant selon leurs objectifs visualisation immersive, conception, simulation décisionnelle et leur niveau d'intégration avec les données opérationnelles, comme l'illustre le **Tableau 2.2**.

Les logiciels de simulation de processus occupent une place importante dans la représentation des flux et des séquences opérationnelles. Parmi ces logiciels, Plant Simulation est employé, parfois associé à des moteurs 3D pour la visualisation [3, 4, 16, 17, 25]. FlexSim est également présent dans les applications de SED, en configurations 2D et 3D [6, 11, 22]. SIMIO est mobilisé pour la modélisation de processus, notamment dans des architectures orientées données [7]. AnyLogic se distingue par sa capacité à combiner plusieurs paradigmes de modélisation comme la SED, SBA et 3D, au sein d'un même environnement, ce qui permet de traiter des besoins variés dans un seul outil [14]. Dans le cadre des JN, il peut également intégrer des tableaux de bord interactifs permettant la visualisation en temps réel des indicateurs de performance, la supervision des entités simulées et le pilotage direct des scénarios, offrant ainsi une composante essentielle pour l'interface utilisateur et la prise de décision [37].

En parallèle de ces outils, des environnements immersifs ou d'émulation sont employés afin

de reproduire visuellement le comportement du système et de permettre une interaction directe avec l’opérateur. Emulate3D est cité comme solution d’émulation 3D orientée services [8]. Unreal Engine 4 est exploité pour offrir une immersion visuelle, souvent reliée à des bases de données [28]. Visual Components 4.2 est également recensé pour des représentations 3D [38], renforçant l’idée que ces environnements visent principalement la démonstration et l’illustration des systèmes.

Dans une logique plus proche de la conception et de la préparation hors ligne, RobotStudio et Siemens NX sont mobilisés pour la modélisation géométrique et la planification de tâches [2].

Enfin, certaines configurations hybrides associent simulation de processus et environnements immersifs, intégrant ainsi des capacités de visualisation à la modélisation décisionnelle.

Au regard de l’état de l’art, le présent travail adopte une représentation numérique fondée sur la SED avec AnyLogic, complétée par une visualisation 3D du modèle. Ce choix s’explique par la capacité de la SED à représenter les flux, événements, files d’attente et processus, éléments essentiels pour l’évaluation opérationnelle et la mise en œuvre de règles de décision. Par ailleurs, AnyLogic permet l’intégration de différents types de modèles et de données, l’accès au code de simulation pour des adaptations spécifiques, l’utilisation de bibliothèques spécialisées en manutention (*material handling*) et le développement de tableaux de bord pour le suivi et l’analyse des indicateurs de performance du JN [37].

Le présent projet ne se limite pas à la mise en place d’un modèle virtuel, mais s’appuie sur des mesures temporelles pour assurer la synchronisation en temps réel. La latence des échanges de données est analysée et la fréquence d’envoi des messages est ajustée afin de maintenir l’harmonisation entre les mondes physique et virtuel. L’architecture de communication repose sur une combinaison de protocoles industriels et IoT (Ethernet, CAN bus, PROFINET, TIA Portal Data Block, ROS 2, Snap7, CSV) qui servent à capter et transférer l’information issue des équipements physiques. Toutefois, le logiciel de SED AnyLogic n’intègre pas nativement ces protocoles : les échanges de données entre le système réel et le modèle virtuel sont assurés exclusivement par le protocole MQTT, dont l’implémentation a été développée dans le cadre de ce projet. L’évaluation menée dans ce travail vise donc à analyser les capacités et les limites de ce mécanisme de synchronisation.

Le **Tableau 2.2** suivant met en perspective ces choix avec les protocoles et représentations virtuelles rapportés dans la littérature. Il synthétise les protocoles de communication, les mesures temporelles mesurées et les types de représentations virtuelles utilisés dans les travaux analysés. Les références présentées indiquent une diversité de protocoles, incluant notamment MQTT, OPC-UA, Kafka et WebSocket, ainsi que des combinaisons spécifiques en fonction des besoins. Les mesures temporelles varient d’un système à un autre, allant de moins

de 100 ms à plusieurs secondes, ce qui reflète des différences dans les architectures réseau et les modes de transmission adoptés. La représentation virtuelle couvre plusieurs catégories, telles que les logiciels de SED, les environnements immersifs, les systèmes d’émulation et les outils de modélisation géométrique. Ces éléments montrent que les protocoles de communication et les représentations virtuelles sont des composantes interdépendantes, dont la sélection est déterminée par les objectifs du projet et les contraintes techniques identifiées.

L’examen du **Tableau 2.2** montre que peu de travaux intègrent des mesures temporelles permettant de quantifier la performance de synchronisation entre le système physique et son jumeau numérique. Parmi les études recensées, seules quelques-unes ont évalué la latence de communication c’est-à-dire le délai écoulé entre la transmission d’une donnée par le système physique et sa réception par le modèle virtuel [8,17], ou l’intervalle de mise à jour correspondant à la fréquence à laquelle les données physiques sont actualisées dans le JN [4, 16,25]. La majorité des travaux se limite à décrire l’architecture fonctionnelle et les protocoles employés, sans fournir d’indicateurs permettant d’évaluer la capacité de synchronisation ou de comparer objectivement les performances entre différentes implémentations. Ce constat met en évidence un écart récurrent dans la littérature et souligne la nécessité, dans les travaux futurs, d’intégrer des mesures temporelles pour quantifier la réactivité et la synchronisation en temps réel des échanges entre le système physique et sa réplique virtuelle. Le présent projet s’inscrit dans cette perspective en étudiant conjointement la latence et l’intervalle de mise à jour afin de quantifier la performance de synchronisation du jumeau numérique.

TABLEAU 2.2 État de l’art des protocoles, systèmes virtuels et mesures temporelles des jumeaux numériques

Auteurs	Protocoles de communication et d’échange de données	Système virtuel	Mesures temporelles	
			Latence	Intervalle de mise à jour
[16]	MQTT ; Node-RED	Plant Simulation (SED) ; Unity	—	200 ms
[2]	OPC UA ; MQTT ; InfluxDB	RobotStudio (3D) ; Siemens NX (CAO)	—	—
[6]	OPC UA ; ADS	FlexSim (SED, 3D)	—	—
[11]	OPC UA ; Cloud	FlexSim (SED, 3D)	—	—
[7]	Kafka ; MongoDB ; Excel wrapper	SIMIO (SED, 3D)	—	—
[8]	OPC UA ; Kepware ; ThingWorx	Emulate3D (3D)	1000 ms	—
[22]	SQL ; OPC/ODBC	FlexSim (SED, 3D)	—	—
[14]	SQL	AnyLogic (SED, SBA, 3D)	—	—
[28]	WebSocket ; MySQL ; ODBC	Unreal Engine 4 (Émulation)	—	—
[17]	MQTT ; OPC UA ; ROS	Plant Simulation (SED) ; Tecnomatix (3D)	11–541 ms	—
[3]	WebSocket ; SQL/ODBC	Plant Simulation (SED) ; Tecnomatix (3D)	—	—
[25]	MQTT ; JADE MAS	Plant Simulation (SED) ; Tecnomatix (3D)	—	1000 ms
[4]	OPC UA ; Modbus ; TCP/UDP ; MySQL	Plant Simulation (SED) ; Tecnomatix (3D)	—	100 ms
[38]	—	Visual Components 4.2 (3D)	—	—
Le présent projet	MQTT ; Ethernet ; CAN bus ; PROFINET ; TIA PORTAL ; ROS 2 ; Snap7 ; CSV	AnyLogic (SED, 3D)	20–225 ms	100 ms

2.4 Cadres normatifs des jumeaux numériques : ISO 23247

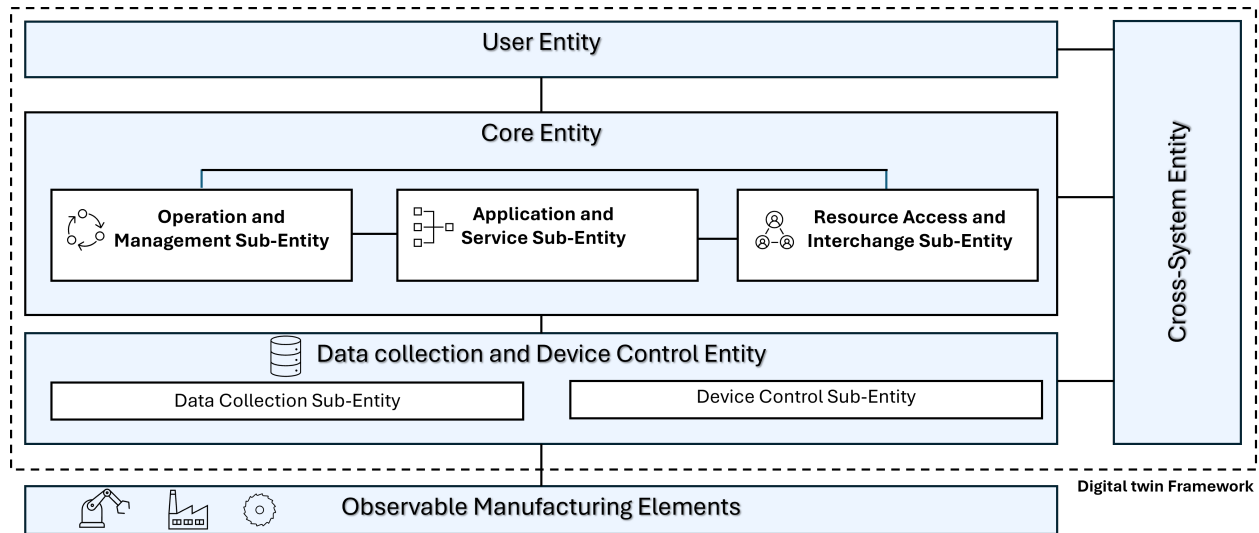
Le déploiement des JN dans les systèmes industriels demeure confronté à un manque de standardisation, aussi bien dans l'industrie que dans la recherche [2]. Les solutions sont souvent développées au cas par cas, selon des architectures spécifiques, ce qui limite leur interopérabilité et leur intégration entre systèmes distincts [11]. Cette absence de cadre commun restreint la mise en œuvre de mécanismes de connectivité, de synchronisation et de sécurité des données [6]. Dans la recherche scientifique, elle réduit la comparabilité des résultats, compromet la reproductibilité des expérimentations et freine le transfert des solutions vers des applications industrielles à grande échelle [25,39].

Comme les JN reposent sur un flux continu de données en temps réel, une manipulation non autorisée peut entraîner des risques opérationnels pour les entreprises. Certains auteurs recommandent ainsi l'instauration de standards pour encadrer le partage et la protection des données [15]. De plus, les informations liées au cycle de vie des systèmes devraient être stockées dans des formats de données standardisés, tels que XML (eXtensible Markup Language), YAML (Yet Another Markup Language) ou JSON (JavaScript Object Notation), afin d'en faciliter le traitement, la traçabilité et la réutilisation [2].

Face à ce défi, la norme ISO 23247 : Systèmes d'automatisation industrielle et intégration — Cadre technique de JN dans un contexte de fabrication, propose un cadre de référence pour le développement de JN manufacturiers. Elle établit une base commune destinée à assurer l'interopérabilité et la modularité des solutions. La première partie définit les concepts, le périmètre et un vocabulaire partagé, constituant ainsi une approche méthodologique pour l'industrie et la recherche [8,40]. La deuxième partie décrit une architecture de référence en organisant les domaines fonctionnels et leurs interactions [8,41]. La troisième partie précise la modélisation des entités manufacturières observables, en distinguant leurs attributs statiques et dynamiques [8,42]. La quatrième partie définit enfin les mécanismes d'échange d'informations, incluant les formats et protocoles nécessaires pour une communication cohérente et sécurisée [8,43]. Les parties 5 et 6 sont cours d'élaboration [44].

La figure 2.4, ci-dessous, illustre la structure fonctionnelle du cadre ISO 23247. Elle organise le JN en cinq entités principales : (i) l'entité utilisateur (User Entity), qui regroupe les interfaces d'interaction avec les opérateurs, techniciens et systèmes externes ; (ii) l'entité centrale (Core Entity), véritable cœur du JN, qui comprend trois sous-entités dédiées respectivement à l'exploitation et à la gestion, aux applications et services, et à l'accès et à l'échange de ressources ; (iii) l'entité de collecte et de contrôle des dispositifs (Data Collection and Device Control Entity), qui assure l'acquisition, le prétraitement et le pilotage des équipements ; (iv)

les éléments manufacturiers observables (Observable Manufacturing Elements), qui désignent les entités physiques surveillées et pilotées ; et (v) l'entité transversale (Cross-System Entity), qui fournit des services de support tels que la traduction de données, l'assurance qualité, la sécurité ainsi que la gestion des composants de communication [8, 41].



Source : ISO 23247-2 :2021

FIGURE 2.4 Cadre technique de jumeau numérique proposé par la norme ISO 23247

Ainsi, l'ISO 23247 constitue aujourd'hui un cadre de référence pour soutenir le développement de JN interopérables, modulaires et connectés en temps réel. Toutefois, comme le montre le **Tableau 2.3**, seules quelques études mentionnent cette norme ou d'autres cadres (ISO 14306 [16], RAMI 4.0 [11, 17]). L'adoption de l'ISO 23247 reste limitée, soulignant le besoin de normalisation et de convergence entre recherche et industrie.

2.5 Lacunes identifiées et positionnement de cette étude

L'examen de la littérature met en évidence un intérêt croissant pour les JN, ainsi qu'une diversité d'approches selon les secteurs et les contextes étudiés. Toutefois, cette diversité de travaux ne se traduit pas par une mise en œuvre complète des capacités fondamentales qui distinguent un JN d'une simulation, à savoir la synchronisation, le fonctionnement en temps réel et la bidirectionnalité, rarement intégrées de manière conjointe.

L'analyse du Tableau 2.3 montre que le temps réel est fréquemment évoqué, mais rarement appuyé par des mesures quantitatives telles que la latence ou les fréquences d'envoi de données, ce qui limite l'évaluation de la réactivité des architectures. La synchronisation, quant

à elle, est abordée de manière hétérogène : certaines études décrivent une mise à jour continue, d'autres une actualisation périodique ou ponctuelle limitée à la phase initiale [14, 22, 36]. Le mécanisme de synchronisation en temps réel des JN demeure ainsi une lacune importante dans la littérature.

La bidirectionnalité apparaît comme la capacité la moins développée. Alors que la définition même du JN suppose un échange d'informations dans les deux sens, la majorité des travaux se limitent à des flux unidirectionnels, réduisant le jumeau à une simple ombre numérique [12, 14]. La revue systématique [10] confirme que seulement 18,7 % des 75 articles analysés intègrent une interaction réellement bidirectionnelle.

Enfin, l'adoption des normes et cadres de référence reste marginale. Quelques travaux mentionnent l'ISO 14306 [16], l'ISO 23247 [2, 8], l'IEC 62832 ou le modèle RAMI 4.0 [11, 17], mais la plupart des contributions s'appuient sur des architectures développées pour des cas particuliers, sans alignement avec un cadre normalisé, ce qui limite l'interopérabilité, le transfert vers l'industrie et la reproductibilité scientifique [7, 13]. L'examen sectoriel du Tableau 2.3 met également en évidence une prédominance des travaux orientés vers la production manufacturière, tandis que les applications logistiques et entrepôts demeurent peu représentées [11, 14, 23, 25].

Le positionnement du présent travail repose sur trois apports : (i) démontrer expérimentalement l'intégration conjointe du temps réel, chiffré par des mesures, de la synchronisation et de la bidirectionnalité ; (ii) inscrire l'implémentation dans le cadre normatif ISO 23247 ; et (iii) étendre le champ d'application du JN aux entrepôts intelligents. Ces contributions répondent aux lacunes relevées dans la littérature et situent le JN comme outil de supervision et de pilotage en temps réel.

TABLEAU 2.3 Synthèse de la revue de littérature sur les caractéristiques fonctionnelles, les applications et les normes des jumeaux numériques

Auteurs	Caractéristiques fonctionnelles du JN		Application	Norme
	Synchronisation en temps réel	Bidirectionnalité		
[16]	✓	✓	Production & logistique	ISO 14306
[2]	✓	✗	Production	ISO 23247
[6]	✓	✗	Production	—
[11]	✓	✗	Entrepôt	IEC 62832, RAMI 4.0
[7]	✓	✗	Production	—
[8]	✓	✓	Production	ISO 23247
[22]	✗	✗	Production	—
[14]	✗	✗	Entrepôt	—
[13]	✗	✗	Production	—
[23]	✗	✗	Entrepôt	—
[28]	✓	✓	Production & logistique	—
[3]	✓	✗	Production	—
[17]	✓	✓	Production	RAMI 4.0
[25]	✓	✓	Entrepôt	—
[45]	✓	✗	Production	—
[5]	✗	✗	Production	—
[4]	✓	✓	Production & logistique	—
[38]	✓	✗	Production	—
Le présent projet	✓	✓	Logistique et maintenance	ISO 23247

✓ : Caractéristique couverte ✗ : Caractéristique non couverte — : Non spécifié

CHAPITRE 3 ARTICLE 1 :REAL-TIME DATA SYNCHRONIZATION MECHANISMS FOR DISCRETE EVENT SIMULATION-BASED DIGITAL TWINS

Dans ce chapitre, le contenu est présenté sous forme d'article afin d'exposer la contribution réalisée dans le cadre de ce travail. Cette contribution comprend la conception et le développement du JN fondé sur la SED intégrant les mécanismes de synchronisation en temps réel, l'expérimentation et l'analyse des résultats au laboratoire SCP-I, ainsi que la recherche bibliographique et la rédaction du manuscrit.

Authors

HEDIR OUKASSI^a, AMEL JAOUA^b, ALI AIDIBE^a, SOUMAYA YACOUT^a

^a*Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montreal, H3T 1J4, Canada*

^b*Industrial Engineering, LR-OASIS, National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia*

Journal

Computers & Industrial Engineering

Date of submission

December 2, 2025

Abstract

Digital Twin (DT) is a core concept in the industry 4.0 era, driving the shift toward real-time decision-making in industrial processes. While the integration of Discrete-Event Simulation (DES) models into the DT core is increasing, a structured real-time data synchronization framework for DES-based DTs remains a challenge. This lack of structure limits the DT's ability to maintain temporal alignment with its physical system and

causes communication latency due to high-rate data exchange. In this paper, we present a real-time data synchronization framework. The framework embeds the DES model as the DT core. It introduces two synchronization mechanisms : Event-Triggered (ET) and Time-Triggered (TT). This dual-mechanism approach achieves selective synchronization of physical components. The framework is implemented and validated experimentally for real-time dispatching of Autonomous Mobile Robots (AMR) to maintenance and charging stations in a smart warehouse environment at the Intelligent Cyber-Physical Systems (I-CPS) Laboratory of Polytechnique Montréal. Experimental results statistically confirm that this selective synchronization method significantly reduced the data exchange latency.

Keywords

Digital twin, Discrete-Event Simulation, Dispatching, Synchronization Mechanisms, Event-Triggered Synchronization, Time-Triggered Synchronization, Latency

3.1 Introduction

The digital transformation of industrial systems is reshaping traditional processes and driving the need for intelligent framework that integrate real-time data acquisition, analysis, and control. The integration of the Internet of Things (IoT), Cyber-Physical Systems (CPS), and embedded analytics enables continuous monitoring and adaptive decision-making across connected environments. Within this paradigm, the DT represents a new concept that enables bidirectional communication between physical systems and their virtual counterparts to support predictive and prescriptive decision-making. For Production Planning and Control (PPC), DT improves operational efficiency by synchronizing digital models with physical processes in real time, which supports scheduling, resource allocation, and maintenance operations [46].

Initially developed within the field of Operations Research (OR), Discrete-Event Simulation DES was employed as an offline analytical tool for experimentation and performance evaluation [47]. With the advent of Industry 4.0 and the increasing integration of real-time industrial data, this evolution has given rise to the DES-based DT, a framework in which simulation extends beyond representational modeling to function as an operational core for real-time monitoring and control [10, 12]. Despite the growing adoption of DES-based DT, there is still a lack of a systematic methodology to define and implement synchronization mechanisms that specify when and how data are exchanged between the physical system and its digital counterpart [48, 49]. This lack of formalization constrains the ability of

DT to maintain real-time synchronization with their physical systems, thereby reducing their capacity to support time-sensitive decision-making. As highlighted in [6], achieving synchronization remains a challenge, as increasing the frequency of data exchange between the DES and the physical system, which is required for high-fidelity synchronization, introduces communication latency. Thus, the objective of the present work is to provide a synchronization framework that embeds DES models as the kernel of a DT by defining and integrating synchronization points within its architecture. The proposed framework aims to maintain temporal alignment by establishing a structured approach to data exchange between the physical and the digital counterparts. It introduces two synchronization mechanisms : an Event-Triggered (ET) synchronization, where updates are initiated by specific changes in system state, and a Time-Triggered (TT) synchronization, where updates occur periodically to support continuous control and monitoring. This approach enables selective synchronization across physical systems to balance the need for responsiveness and computational efficiency. To validate this approach, the proposed synchronization framework was developed and implemented for the real-time dispatching of Autonomous Mobile Robots (AMRs) for maintenance and charging operations within the I-CPS laboratory at Polytechnique Montréal. The laboratory operates as a smart warehouse since it integrates AMRs, Programmable Logic Controller (PLC), and industrial sensors [50]. Experiments were conducted to evaluate how variations in the data transmission rate and the number of synchronization points considered in the DES-based DT affect message latency, which can delay the execution of dispatch commands and prevent AMR from reaching maintenance or charging stations at the required time. The paper is organized as follows. Section 2 presents a review of the existing studies related to DT synchronization and the integration of DES as the modeling core within DT architectures. Section 3 presents the conceptual framework of the proposed synchronization mechanism. Section 4 describes the implementation and experimental setup in the I-CPS Laboratory. Section 5 reports the results and analysis of the latency experiments. Finally, Section 6 concludes the paper by summarizing the main contributions, limitations, and directions for future research.

3.2 Related work

The integration of embedded sensors, the IoT, and CPS has reshaped manufacturing and logistics by enabling real-time monitoring and adaptive decision-making across dynamic environments. Embedded sensors act as primary data sources, while IoT infrastructures ensure their transmission across heterogeneous networks. CPS builds upon this foundation by integrating computational models and control mechanisms with physical processes,

forming a closed loop in which data are captured, transmitted, analysed, and translated into control actions [18]. Within this framework, a DT is conceived as a component of the CPS that establishes a bidirectional data flow between the physical system and its digital counterpart, enabling synchronization, feedback, and command exchange that supports real-time decision-making [51]. DT have been adopted across a wide range of industrial domains, including manufacturing and logistics. They directly support PPC by enhancing responsiveness in scheduling, dispatching, resource allocation, maintenance planning, and real-time decision-making. For example, studies such as [51, 52] demonstrated how the implementation of DT in maintenance operations enables condition monitoring and predictive interventions, thereby reducing unplanned downtime. Also, in logistics and warehousing, DT enhance the coordination of Automated Guided Vehicles (AGVs) and AMRs by enabling real-time monitoring and adaptive scheduling of charging and maintenance, thus improving overall efficiency and responsiveness [4, 48]. As DT applications expand, DES-based DT have emerged as an evolution of traditional DES models, extending conventional offline simulations by incorporating real-time data connectivity and adaptive decision mechanisms [10, 12].

3.2.1 DES-based DT approach

A DES-based DT is a digital counterpart of a physical system in which a DES model serves as the core component, synchronized with real-time data from the physical process [10]. In fact, DT developed for PPC commonly rely on DES, which is an approach that represents a system as a set of interacting entities whose states evolve through discrete events over time [47]. DES has been widely adopted in DT frameworks [14, 48, 53, 54] as it represents the dynamic behaviour of manufacturing activities with a level of detail suitable for operational analysis. Early developments of DT implementations based on DES mainly focused on virtual replication and scenario analysis rather than on bidirectional and real-time synchronization. [55] proposed a DES-based DT for real-time monitoring and performance prediction; however, the data exchange remained one-directional, limited to updating the simulation from the physical system without feedback or control. [56] developed a DES-based decision support tool for in-house logistics, marking an early step toward DES-based DT implementation. The model leveraged real operational data to support decision-making but operated unidirectionally, without real-time synchronization or feedback from the physical system. Similarly, [14] developed a hybrid DES model for an automated warehouse but explicitly defined it as a Digital Shadow, acknowledging the absence of continuous bidirectional communication. [13] further presented a conceptual DES-based framework, yet its application was conducted offline, with real-time alignment identified as a future goal dependent on 5G connectivity. A classification presented in [15] distinguishes

simulation models from DT according to their functional capability levels. Simulation is characterized by four capability levels, referred to as the 4S framework : (S1) Modeling, (S2) Analyzing, (S3) Predicting, and (S4) Prescribing. (S1) Modeling defines the ability to create a virtual representation of a physical system from historical or experimental data. (S2) Analyzing involves assessing system performance through experimentation and parameter variation. (S3) Predicting refers to estimating system behavior under new or alternative conditions. (S4) Prescribing corresponds to identifying optimal configurations or actions to improve system performance. Similarly, the DT evolves through four maturity levels, known as the 4R framework : (R1) Representation, (R2) Replication, (R3) Reality, and (R4) Relational. (R1) Representation consists in defining and collecting data from the physical system to construct its digital counterpart. (R2) Replication refers to reproducing the physical system's behavior within the virtual environment. (R3) Reality is achieved when insights generated in the virtual model are applied to the physical system for adaptive decision-making. Finally, (R4) Relational defines the stage where the physical and the digital counterpart remain synchronized through bidirectional data exchange. This classification highlights that, although simulation and DT share common modeling principles, only the DT establishes a continuous feedback loop with its physical counterpart, enabling real-time synchronization and closed-loop control. Despite the potential of DES to serve as the modelling kernel of DT, current DES software still faces limitations in maintaining continuous data exchange and real-time responsiveness. As noted in [57], DES software often requires integration with external programming environments to compensate for these shortcomings. Similarly, [58] observed that while DES tools are increasingly incorporating data-driven components to enable real-time communication and adaptive synchronization, such features are still in their early stages and fall short of fully supporting DT functionality. More recently, [59] proposed a structured methodology for transforming DES models into DT, identifying persistent gaps in real-time data integration, bidirectional communication, and autonomous decision-making. These developments indicate that DES tools are progressively extending their functionalities to incorporate DT capabilities, particularly those related to data synchronization and connectivity. Also, [11] noted that most DT implementations were developed for specific use cases without structural generalization, motivating recent standardization efforts such as the [41] framework to establish a common reference for integration across manufacturing environments.

3.2.2 Standardization efforts for DT integration

To address the lack of methodological standardization observed in DT implementations, international organizations have introduced several frameworks to structure DT integration

[8]. While other standards have been developed in the broader context of Industry 4.0, the (ISO 23247-2, 2021) framework specifically targets the integration of DT in manufacturing systems, as illustrated in [39]. The (ISO 23247-2, 2021) establishes a reference framework for implementing and integrating DT in manufacturing. It defines the roles of each entity and the required information exchange to support data flow, model execution, and control. The standard also recommends the use of modeling languages such as Unified Modeling Language (UML) to describe entity structures and interactions, and structured formats such as JavaScript Object Notation (JSON) or Extensible Markup Language (XML) for data representation, promoting interoperability. Several studies have applied ISO 23247-2 (2021) in manufacturing contexts, such as in manufacturing cell reconfiguration [2] and AMR dispatching with real-time data exchange [39]. However, these works did not assess synchronization mechanisms or scalability. Overall, studies adopting ISO 23247-2 (2021) [2, 8, 39] confirm its relevance for structuring DT architectures. Nevertheless, the standard remains descriptive, providing guidance at a conceptual and architectural level but lacking concrete instructions for applying synchronization mechanisms. In fact, as DT synchronization is still an active research domain, its integration within the (ISO 23247-2, 2021) framework has yet to be clearly established.

3.2.3 DT synchronization

The main capability of a DT is to maintain alignment between the physical system and its digital counterpart through bidirectional data exchange. This alignment, referred to as synchronization, enables the continuous transfer of data and control commands that keep both representations updated over time [6]. The synchronization ensures that the DT receives and processes information within the required time window, so that decisions computed in the digital counterpart can be transmitted and executed on the physical system without delay [49]. It is achieved through data communication infrastructures that transmit sensor information, system states, and control actions with defined temporal accuracy. In practice, this function relies on industrial communication protocols that coordinate data flow between physical assets, control systems, and computational services [8]. Among these protocols, Message Queuing Telemetry Transport (MQTT) offers a lightweight publish/subscribe mechanism managed by a broker [26]. It is widely used in DT implementations to connect physical assets with their digital counterpart, supporting data pipelines, machining cell simulations, and smart warehouses for real-time AMR tracking [9, 16, 25]. Similarly, Open Platform Communications-Unified Architecture (OPC UA), a secure client-server protocol ensuring interoperable data exchange across heterogeneous devices [8]. Kafka has also been adopted as a distributed streaming platform to integrate real-time shop-floor data

with simulation models, enabling continuous synchronization and decision support in DT environments [7]. The implementation of a DT is not limited to a single communication protocol; it may incorporate multiple protocols selected according to the operational requirements and the system context. Different protocols can be adopted separately or combined to balance lightweight messaging, interoperability, and scalability [17]. According to [60], the performance of these protocols depends on meeting strict temporal requirements, defined as the need to ensure that computations and communications are completed within application-specific deadlines. In high-speed control loops, such as those used in industrial processes, even small delays can compromise stability, whereas interactions involving human operators tolerate longer delays on the order of tens of milliseconds. Given these temporal requirements, latency is defined as the elapsed time between the transmission of data by a sender and its reception by the intended receiver [26]. In DT implementations, latency is a critical temporal indicator, as any delay between physical data transmission and the digital counterpart update directly affects the synchronization, monitoring, analysis, and control functions. According to [35] achieving low latency is therefore essential, since high latency leads to outdated system states and reduces prediction and control accuracy. In the context of real-time systems, latency is influenced by the choice of communication mechanism. Two main synchronization mechanisms are distinguished : Event-Triggered (ET) and Time-Triggered (TT) [60]. In ET, updates occur only in response to significant events, providing resource efficiency but introducing variability in response times, whereas in TT, updates occur at fixed, predefined intervals under a synchronized global clock, ensuring deterministic timing at the cost of higher communication and processing load. This distinction is essential for DT, as it directly affects how and when the digital counterpart is synchronized with its physical counterpart, determining DT's ability to support real-time decision-making and control. Some studies have investigated synchronization in DES-based DT systems from different perspectives. [10] classified synchronization into real-time (RT) and near-real-time (NRT) approaches, emphasizing their respective relevance to control-oriented and decision-support applications. They observed that NRT synchronization is the most commonly adopted approach, as RT implementations remain constrained by communication delays and integration challenges. [6] proposed a DES-based synchronization mechanism for DT using Petri nets to coordinate information flow between physical and digital counterparts. The study included an experimental implementation that demonstrated the feasibility of the approach but did not introduce quantitative metrics to assess synchronization performance. Latency and update frequency were discussed qualitatively, without measurements or defined thresholds, which limited the capacity to evaluate the real-time responsiveness of the proposed mechanism in industrial settings. Furthermore, [61] introduced a catalog of

eleven synchronization patterns for DT systems, encompassing event-driven, time-driven, hybrid, adaptive, and context-aware mechanisms. The authors demonstrated their approach through simulation, notably in the Automated Guided Vehicle (AGV) case, where hybrid synchronization combined ET updates with periodic checks to maintain alignment during both active and idle phases. Although the work provided a structured classification of synchronization methods, it remained conceptual and did not include quantitative indicators such as latency, update rate, or data loss. [49] further formalized synchronization as a stochastic control problem that determines optimal update timing to balance synchronization cost and prediction accuracy. They compared state-independent, state-dependent, and full-information policies, showing that state-dependent synchronization reduces unnecessary updates while preserving model accuracy. Nonetheless, latency was expressed only in qualitative terms like high, low, or near-zero, without quantitative thresholds or empirical validation, leaving its industrial applicability undetermined. [62] proposed a DES-based DT to support tactical, strategic, and operational decisions in manufacturing. The DES model operated with daily or weekly updates rather than continuous synchronization, functioning in a near-real-time mode that limited responsiveness and excluded real-time control. This reflects a recurring limitation of DES-based DT, which often remain intermittent decision-support tools rather than continuously synchronized systems. While these recent studies have examined synchronization in DES-based DT, their contributions remain conceptual, with limited empirical or quantitative assessment. Most studies describe synchronization using qualitative terms such as low latency or instantaneous but rarely establish measurable performance indicators. Consequently, the absence of evaluation criteria limits understanding of how synchronization mechanisms operate under industrial real-time conditions, where temporal parameters such as latency and update frequency directly influence the decision-making process.

3.2.4 Key insights from this literature review

This literature review highlights three main areas where research on DES-based DT remains limited. The first gap, as reported in [59], is that most existing implementations rely mainly on one-directional rather than bidirectional data exchange, which prevents full bidirectional alignment between the physical and the digital counterparts. Secondly, standardization plays a central role in structuring and formalizing DT developments. As demonstrated in [8], adopting such frameworks enables consistent integration of IoT platforms and simulation models, ensuring traceable data exchange and system interoperability. Nevertheless, the (ISO 23247-2, 2021) standard remains descriptive, offering conceptual guidance without specifying how synchronization mechanisms should be applied. It provides a conceptual and

architectural reference rather than operational specifications, which may explain the limited quantitative consideration of synchronization and latency, a challenge that remains open in current research. Finally, the main gap in the literature is that DES-based DT synchronization lacks the fundamental characteristics of a real-time systems application, including guaranteed bounded latency, and strict temporal consistency. As noted in [49], most existing studies leverage DT primarily for production planning, often neglecting the issue of synchronization required for the real-time production control. In particular, the relationship between synchronization points and triggering mechanisms remains unexplored, even though these mechanisms are essential for ensuring that the DT reflects changes in the physical system, thereby directly affecting real-time responsiveness. Consequently, quantitative metrics are needed for the evaluation of synchronization mechanisms in industrial real-time settings, where latency, update intervals, and other temporal parameters influence decision-making. Thus, the objective of the present work is to address these gaps by developing and validating a real-time synchronization mechanism for DES-based DT applications, using a warehouse system as proof of concept and following a formal methodology based on the (ISO 23247-2, 2021) framework.

3.3 Conceptual solution design

The objective of this work is to formalize a synchronization mechanism extending the (ISO 23247-2, 2021) reference architecture by defining how data are exchanged between the DES-based DT and the physical system. Continuously synchronizing all physical components over time increases network load and induces latency, which can affect the alignment between the physical and digital counterparts. To address this issue, the proposed framework introduces a selective synchronization mechanism that distinguishes data exchange according to its function in the system. The framework differentiates between data used for supervision and monitoring, and data required for decision and control. TT synchronization is assigned to control-related OMEs that require periodic updates, while ET synchronization is associated to monitoring sensors that transmit data only when a state change occurs. The following section describes the architecture of the proposed synchronization framework and the design principle of ET and TT synchronization points.

3.3.1 Architecture of the Proposed Synchronization Framework

The proposed framework is designed to coordinate real-time data exchange between the physical and the digital counterparts of a manufacturing system. As illustrated in Figure 3.1, the framework establishes a bidirectional communication loop between the physical system

and the DES-based DT. Data generated by sensors, actuators, and robots are transmitted to the DES model, where state synchronization enables prediction, decision generation, and visualization. The resulting decisions are then communicated back to the physical system to adjust its operational state. The interaction between both layers is governed by synchronization mechanism that ensure temporal coherence : ET synchronization is activated by system state change, while TT synchronization operate at predefined time intervals to enable continuous monitoring and control.

The proposed synchronization framework is developed in accordance with the (ISO 23247-2, 2021) architecture, which structures DT implementations through four entities : the Observable Manufacturing Element (OME), the Data Collection and Device Control Entity (DCDCE), the Core Entity, and the User Entity. In this configuration, the OME represents the physical element to be mirrored, such as equipment, material, process, or personnel, and serves as the primary data source for the DT. The DCDCE acquires and transmits OME data while enabling bidirectional command exchange between the physical and the digital counterparts. The Core Entity hosts the digital representation, integrating simulation, analytics, and synchronization mechanisms to maintain alignment with the OME. Finally, the User Entity provides the interface for monitoring and control actions.

Within this reference structure, the present work focuses on the design of a synchronization mechanism that governs data exchange between the DES-based DT and the OMEs, as illustrated in Figure 3.1 Each $\{OME_i\}$ represents a physical component of the system, such as a sensor, actuator, or robot, and is associated with a synchronizer. Each synchronizer defines how data from $\{OME_i\}$ are exchanged with the digital model. The synchronizer may include event-based synchronization channels $\{E_{i,k}\}$ and/or time-based synchronization channels $\{T_{i,j}\}$, where the indices k and j refer to specific event or time triggers associated with $\{OME_i\}$. Event-based synchronizer is activated when a state change is detected, whereas time-based channels operate at fixed intervals defined by the update rate of the monitored variables and enable decision feedback. For TT, the Core Entity transmits a decision to the DCDCE, which then sends the corresponding command to the $\{OME_i\}$ to update its operation. This structure enables each OME to manage multiple synchronization mechanisms according to its operational role, ensuring that both ET and periodic TT are synchronized within the DES-based DT. For example, OME_1 may include $\{E_{1,1}, E_{1,2}, T_{1,1}\}$, OME_2 may include $\{E_{2,1}\}$, and OME_3 may include $\{T_{3,1}\}$.

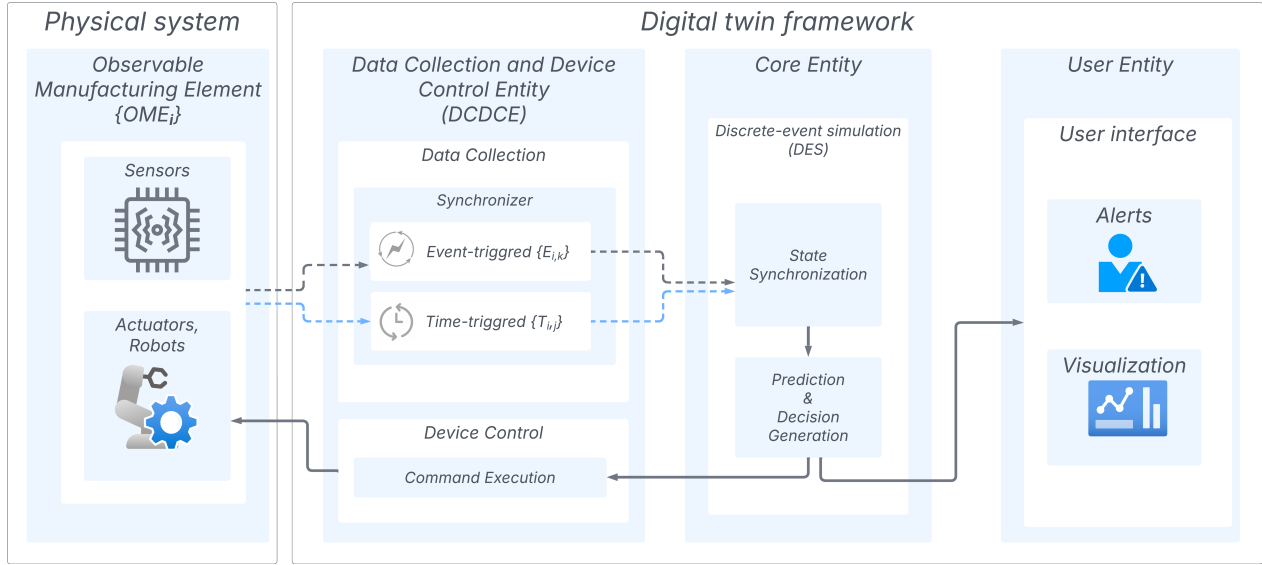


FIGURE 3.1 The conceptual design of the proposed synchronization mechanism for the DES-based DT

3.3.2 Design Logic of Event and Time Triggered Synchronization Points

The Unified Modeling Language (UML) sequence diagram in Figure 3.2 illustrates the design principle of the synchronization mechanism implemented within the proposed framework. Each synchronization point defines how the DCDCE manages the acquisition, transmission, and feedback of data between the OME_i and the DES. Two synchronization mechanisms are implemented ET and TT, each governed by a distinct triggering condition. The diagram provides an overview of the interactions among the four (ISO 23247-2, 2021) entities (OME_i , DCDCE, Core Entity, and User Entity). It shows how the DCDCE coordinates information flow between the OME and the DES model through these two mechanisms : the upper part depicts ET exchanges activated by state change, while the lower part illustrates TT synchronization executed at fixed intervals. Together, these interactions demonstrate how the framework enables bidirectional data flow that supports monitoring, decision generation, and feedback within the DT environment.

In the event-based synchronization mode ($E_{i,k}$), the state change S represents the state of the monitored sensor associated with each OME_i . The DCDCE monitors S and evaluates whether a change has occurred or not. The logic follows an alternative (alt) condition : when no state change is detected ($S = 0$), data transmission remains inactive; when a change is detected ($S = 1$), the DCDCE triggers data transmission to the Core Entity. In practice, ($S = 1$) may correspond to a change in the sensor state, for example when a proximity sensor

detects an object. The DES model then receives the data, synchronizes the system state, and notifies the User Entity through the user interface. This synchronization mode supports supervision and monitoring functions, where updates are event-driven and do not generate control commands toward the OME_i .

In the time-based synchronization mode ($T_{i,j}$), data transmission occurs periodically (loop) at fixed time intervals $\Delta T_{i,j}$, independent of state change. The DCDCE transmits operational data from actuators and robots to the Core Entity, where the DES model updates the system state and generates the corresponding decision or prediction. The resulting output is then sent back to the DCDCE that translates it into a control command directed to the respective OME_i , while the User Entity is updated with the latest system state and feedback.

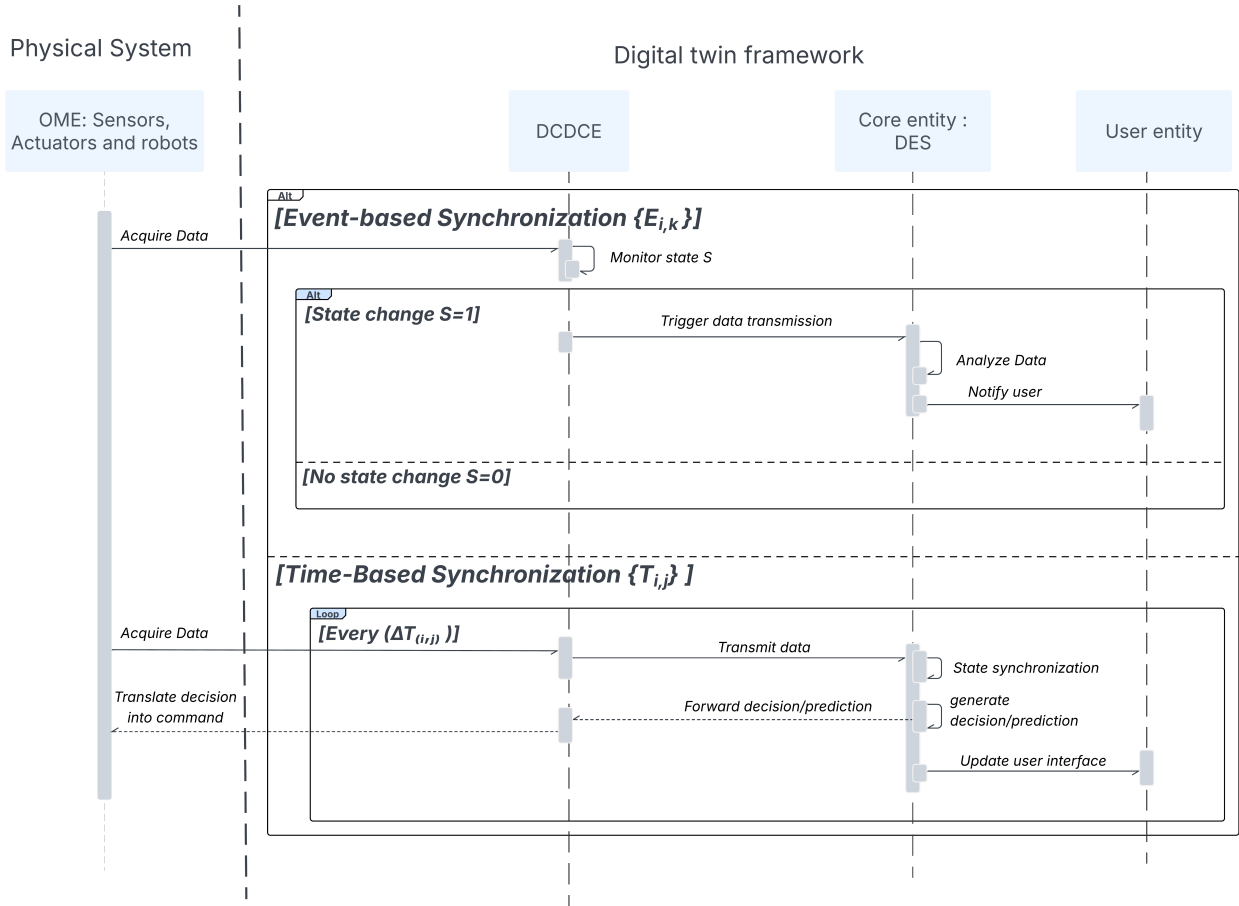


FIGURE 3.2 The UML sequence diagram illustrating the design logic of the ET and TT synchronization mechanisms within the proposed DES-based DT framework

In summary, the proposed synchronization mechanism integrates both ET and TT exchanges within a unified framework. Through this configuration, the framework adjusts the synchronization mechanism to match operational demands, maintaining real-time coherence

between the OME and the DES-based DT.

3.4 Use case of the proposed DES-Based DT Framework

3.4.1 Use case at the I-CPS laboratory

The proposed synchronization mechanism for the DES-based DT framework was implemented in the I-CPS laboratory at Polytechnique Montréal. The I-CPS laboratory operates as a smart warehouse equipped with three AMR, multiple sensors, and diverse communication protocols that enable real-time data exchange and synchronization. In this implementation, the DES-based DT performs the real-time dispatch of AMRs for maintenance and charging activities. It continuously monitors the operational state of each robot, particularly motor condition and battery level, to issue dispatch commands that redirect the corresponding AMR to the maintenance or charging station when required. As illustrated in Figure 3.3, the I-CPS Laboratory integrates both digital and physical components that support the implementation of the proposed DES-based DT synchronization framework. Figure 3.3 (a) presents the DT user interface entity visualizing in real time the state of charge and motor status of the AMR. Figure 3.3 (b) shows the I-CPS laboratory testbed at Polytechnique Montréal (Québec, Canada), including the transporter and Dual-Arm Manipulator AMRs, charging, and maintenance stations. Figure 3.3 (c) displays the DT user interface of the reception and shipping storage areas with a real-time stock-level dashboard, while Figure 3.3 (d) depicts the corresponding physical racks equipped with proximity sensors for detecting product presence in both storage zones. Together, these digital and physical components form the experimental foundation for evaluating real-time data acquisition within a warehouse operation. In the I-CPS laboratory testbed, the Dual-Arm Manipulator AMR transfers the product from the Computer Numerical Control (CNC) machine, which serves as the supplier, to the receiving storage area. It then retrieves the product and places it onto the transporter AMR, which subsequently carries it through the identification and inspection sequence. This sequence starts with the Radio Frequency Identification (RFID) writer station and continues with the camera vision system for inspection. If the product fails inspection, it is directed to the quarantine area; otherwise, it proceeds to the RFID reader and Quick Response (QR) code stations before reaching the waiting area. The validated product is finally stored in the shipping storage area. Throughout these operations, the DT receives real-time data from the connected equipment and sensors. The DES-based DT analyzes these data to detect the anomalies of the motors of the transporter AMR. When an anomaly is identified, a command is issued to redirect the affected AMR to the maintenance station. Similarly, when the State of Charge (SoC) of an AMR battery reaches

20%, the DES-based DT model sends a dispatch command assigning the AMR to move to the charging station. The I-CPS Laboratory architecture supports these operations through an interconnected network integrating sensors, controllers, and industrial communication protocols. The following section presents this architecture, detailing the communication layers, data flow, and protocols that enable real-time connectivity between the physical and the digital counterpart

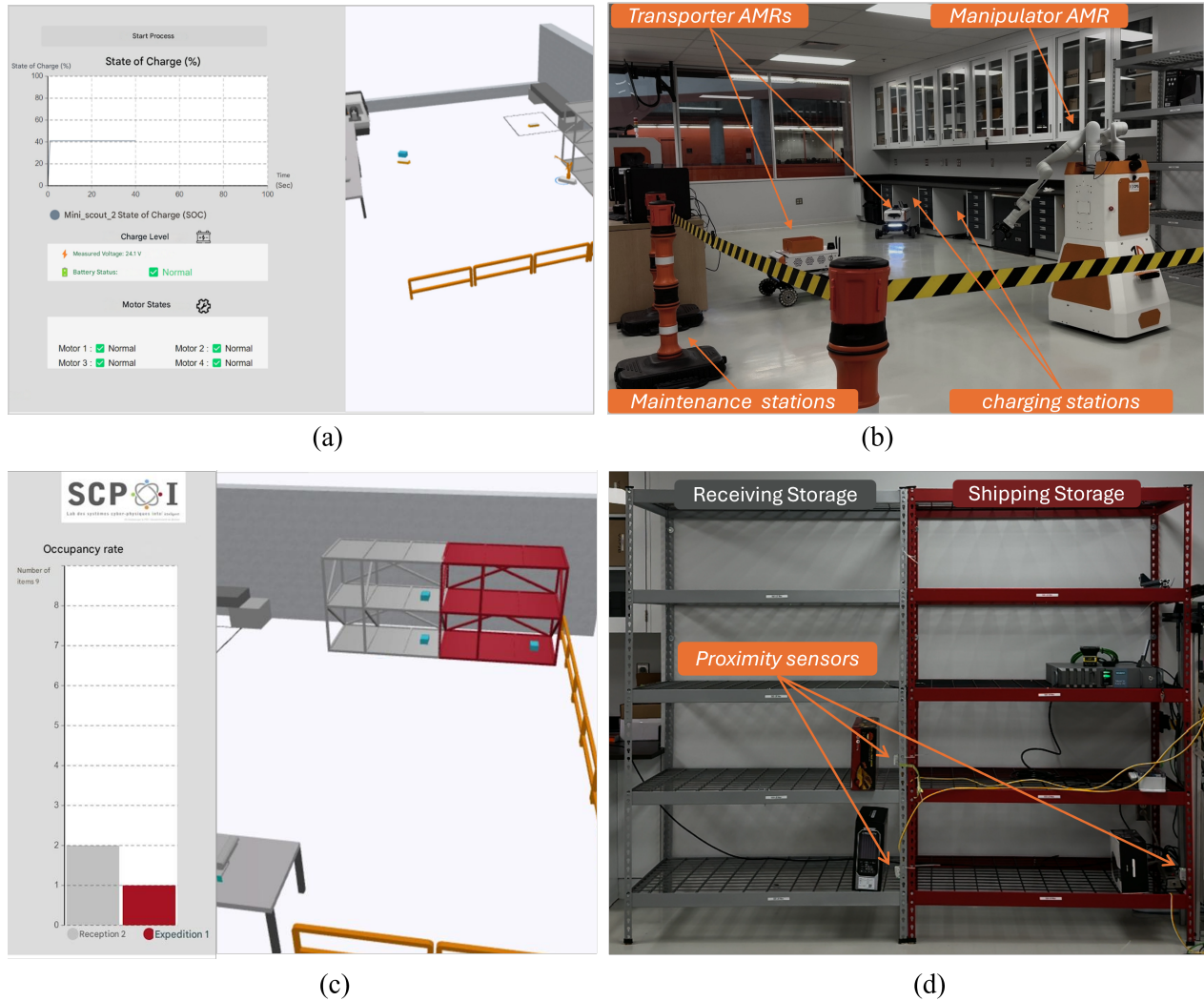


FIGURE 3.3 (a) The DT user interface showing real-time charge level and motor status of the AMR; (b) The I-CPS laboratory testbed at Polytechnique Montréal (Québec, Canada) with transporter and Dual-Arm Manipulator AMRs, charging, and maintenance stations; (c) The DT user interface of the reception and shipping storage area with a real-time stock-level dashboard; (d) The Physical reception and shipping storage area equipped with proximity sensors for real-time stock monitoring

3.4.2 System Architecture

Within the I-CPS Laboratory, a DES-based DT architecture was implemented. Figure 3.4 presents the overall configuration of the physical and the digital counterparts, highlighting the communication pathways and data flow between entities. At the physical level, the OME include different components distributed across the warehouse stations : two transporter AMR, one dual-arm manipulator AMR, two RFID reader/writer modules, a QR-code reader, a camera vision system, and proximity sensors. These components provide real-time operational data related to motion, identification, inspection, and storage activities. The AMRs communicate wirelessly with the Edge PC using ROS 2 and MQTT protocols, enabling bidirectional data transmission for navigation, task execution, and state feedback. The Edge PC acts as a communication hub bridging the ROS 2/MQTT environment of the AMRs and is linked via wired connections to the Industrial Ethernet (IE) switch and the PLC. The stationary devices, including the RFID modules, QR-code reader, and camera vision system, communicate with the PLC through wired access points (AP1–AP3) connected to the IE switch. The proximity sensors are connected to the PLC through IO-Link modules, also routed via the IE switch, providing real-time data for product detection in the receiving and shipping areas. The cross-system network therefore integrates the IE switch, access points, and IO-Link modules that interconnect the sensing and control infrastructure. The DCDCE comprises the PLC, the Edge PC, and Workstation 2 (WS2), which collectively manage data aggregation and communication between the physical and the digital counterparts. Operational data generated by the OME are consolidated in the PLC. WS2 hosts the Python-Snap7 interface [63] and the Mosquitto MQTT broker [64], which extract the aggregated data from the PLC and transmit them wirelessly to Workstation 1 (WS1), where the Core Entity executes the DES-based DT model. To achieve bidirectional data exchange, the DCDCE also receives decisions from the Core Entity through MQTT and translates this feedback into executable control commands transmitted to the AMRs. Specific data, such as battery and motor states, are transmitted directly from the AMRs to the Edge PC and then to the Core Entity, bypassing the PLC. The Core Entity, hosted on WS1, runs the DES-based DT model in AnyLogic, as illustrated in Figure 3.4, it replicates in real time the operational state of each OME. The model receives data from the DCDCE, updates the virtual state of the system, and generates maintenance or charging decisions that are sent back to the physical components. The Core Entity also records operational logs in CSV files for traceability and further analysis. The User Entity provides the human machine interface of the DT, displaying in real time the AMRs' SoC, motor condition, and storage-rack occupancy while visualizing the 3D process model, thereby enabling supervision, monitoring, and control.

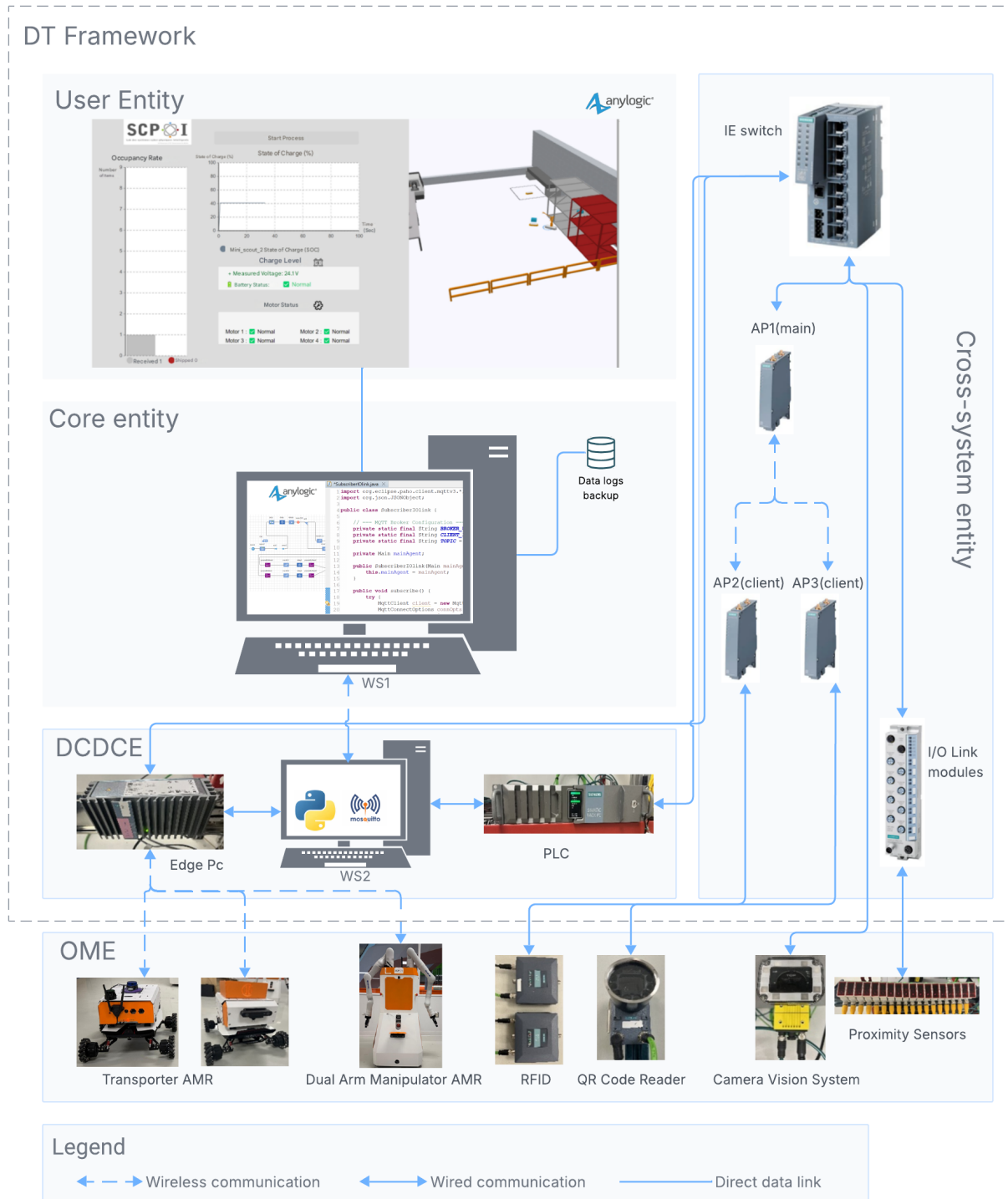


FIGURE 3.4 The communication architecture of the DES-based DT implemented in the I-CPS Laboratory

3.4.3 Data Representation and Modeling of Observable Manufacturing Elements

This section defines the formal data model used to represent the OME of the I-CPS warehouse within the DES-based DT. The modeling approach is formalized through the entity–relationship diagram, which specifies the entities, their relationships, and the cardinalities governing data organization and flow. Figure 3.5 illustrates the diagram that represents the OMEs and their corresponding digital entities. In this model, the DES-based DT maintains a one-to-one association with the PLC, which consolidates data from multiple connected devices, including RFID modules, QR-code readers, inspection cameras, and proximity sensors. This configuration establishes a one-to-many relationship between the PLC and the connected devices, while each device is linked to only one PLC, ensuring a unique data correspondence. The Edge PC maintains a one-to-many relationship with the AMRs, each of which is structured as a composition of two entities, AMR Health and AMR Status, connected through one-to-one cardinalities describing the AMR’s condition and operational state. At the instance level, the I-CPS implementation includes three AMRs (two Transporter AMRs and one dual-arm Manipulator AMR). Two RFID modules are instantiated for product identification at different workstations, while 18 proximity sensors are deployed, 9 assigned to the receiving storage area and 9 to the shipping storage area. One QR-code reader and one camera vision system are also instantiated. This data representation and modelling structure defines the correspondence between the physical OMEs and their virtual entities, establishing the structural foundation for the synchronization mechanisms detailed in Sections 3.4.4 and 3.4.5.

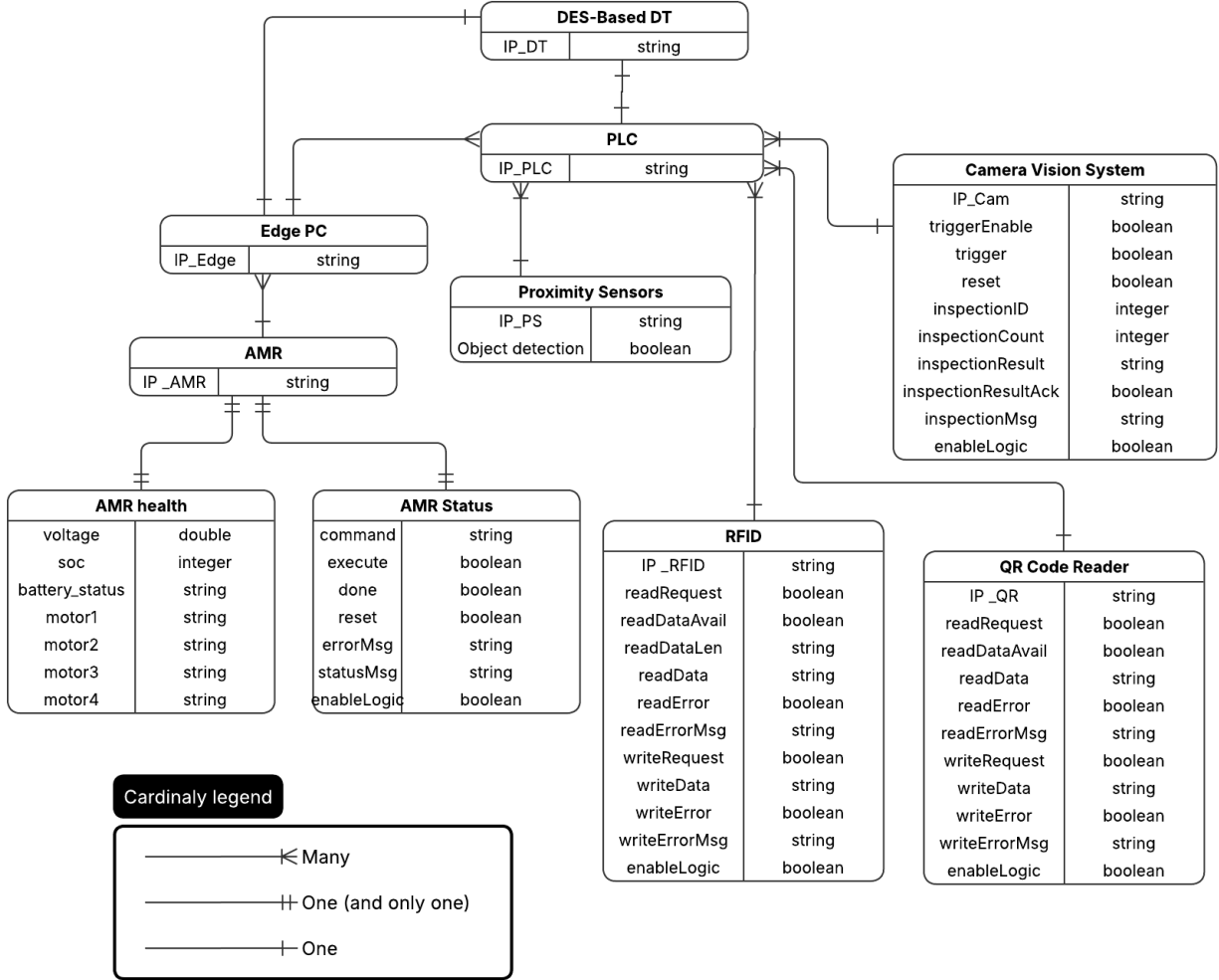


FIGURE 3.5 The Entity-Relationship Diagram illustrating the data representation and modeling of the OME within the DES-based DT framework implemented in the I-CPS Laboratory

3.4.4 Implementation of Event-Triggered Synchronization

The ET synchronization was implemented to manage data acquired from the monitoring devices of the I-CPS laboratory which include proximity, identification, and inspection sensors such as the RFID reader/writer, QR-code reader, and camera vision system. Figure 3.6 presents the UML sequence diagram describing the data flow from these devices to the DES-based DT. ET synchronization triggered by sensors initiates the data-exchange process through two monitoring paths. Proximity sensors on the receiving and shipping racks detect object presence and transmit discrete signals through the IO-Link modules and Industrial Ethernet switch to the PLC. In parallel, identification and inspection sensors send their data

via the AP client, AP main, and IE switch to the PLC. The MQTT publisher service running on WS2 establishes a Snap7 connection to the PLC, reads the corresponding data blocks, and continuously monitors the state of each sensor variable. A conditional alternative (alt) block, defined in the UML sequence logic, ensures that data are transmitted only when a state change is detected. When the state change occurs ($S = 1$), the data are parsed into a JSON structure and published to the MQTT broker. If no state change is observed ($S = 0$), the cycle ends without publication, thereby avoiding redundant communication and reducing network load. In this context, a state change corresponds to a Boolean transition from FALSE to TRUE or from TRUE to FALSE triggered by the sensor. Each topic in the MQTT broker corresponds to a specific group of the OME. The topic “plc/sensors” carries the Boolean states of the proximity sensors installed on the racks, with assigned to the receiving and the shipping area. The topics “icps/rfid”, “icps/inspection”, and “icps/qr code” respectively represent the RFID reader/writer modules, the camera vision system, and the QR-code reader. On the subscriber side, the DES-based DT hosted on WS1 decodes the incoming JSON payloads and updates the DES model variables in real time. The messages received from “plc/sensors” update the variables $Sensor_1$ to $Sensor_{18}$, which control the logical status of the storage blocks in the DES-based DT model.

Data received via the topics “icps/rfid,” “icps/inspection,” and “icps/qr code” respectively update the variables `lastRFID`, `lastInspect`, and `lastqr`, which store the latest tag, inspection, and identification data. Each message is automatically logged and periodically exported for traceability. Error handling is integrated into the MQTT loop to manage publisher–broker connection failures. This ET synchronization ensures that updates in the DES-based DT occur upon state change detected in the physical system, preserving real-time consistency. Consequently, the DES-based DT model remains temporally aligned with the operational state of each OME.

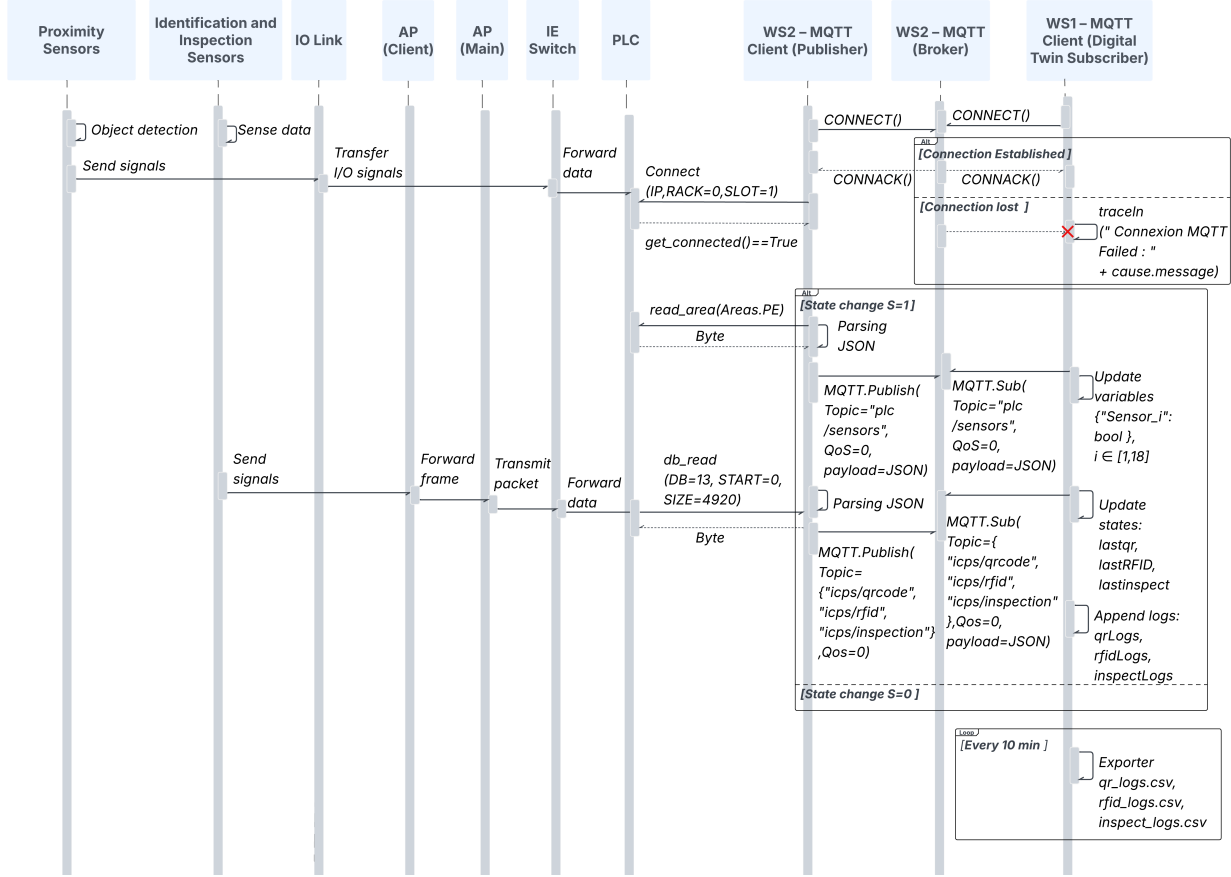


FIGURE 3.6 The UML sequence diagram illustrating the implementation logic of ET synchronization within the proposed DES-based DT framework

3.4.5 Implementation of Time-Triggered Synchronization

The TT synchronization defines a bidirectional communication loop between the AMRs and the DES-based DT, enabling continuous data exchange in both directions. Figures 7 and 8 illustrate this bidirectional synchronization mechanism, showing the continuous feedback flow from the AMRs to the DES-based DT and, conversely, the control flow from the DES-based DT to the AMRs.

Figure 3.7 shows the data flow from the physical system to the DES-based DT where the transporter and manipulator AMRs continuously publish feedback containing status and pose, motors condition, and battery state of charge. The status and pose data are transmitted from the AMRs to the Edge PC, then through the IE switch and the PLC, before being published by WS2 to the DES-based DT under the topic “icps/robots”. The motors condition and battery SoC follow a direct transmission path from the AMRs to the Edge PC to the DES-based DT under the topic “Robots/batt/motor”. All feedback data are extracted on

the Edge PC, where voltage readings are processed to estimate the SoC and the data are encoded in JSON format. The MQTT publisher on WS2 retrieves the PLC data and sends the payloads to the MQTT broker. The DES-based DT subscriber on WS1 updates the AMR status variables in real time and periodically exports the recorded logs to `Robot_logs.csv` for traceability and analysis.

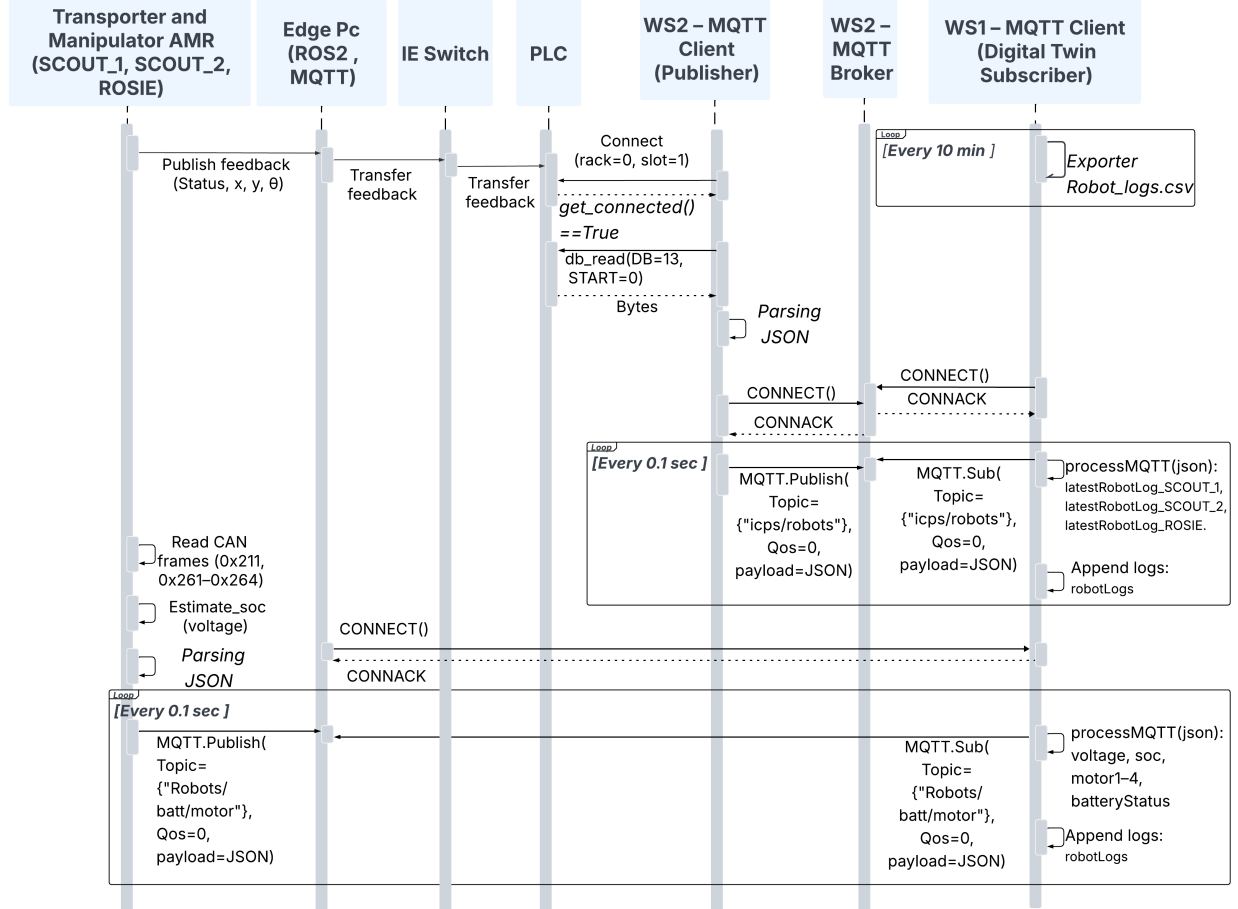


FIGURE 3.7 The UML sequence diagram of TT synchronization in the proposed DES-based DT framework.

Figure 3.8 depicts the reverse data flow, from the digital counterpart to the physical system, achieving the bidirectional loop. The DES-based DT controls the operational variables related to voltage, motor status, and battery state. The DES model, acting as an MQTT publisher, analyzes these parameters and determines the corresponding control action. When the battery state of charge drops below 20%, the model publishes commands through the topic “robot_command”, dispatching the robot to the charging station. When a motor anomaly is detected, related to temperature or current conditions exceeding the predefined thresholds established by the manufacturer, the DES-based DT model sends a command to dispatch

the robot to the maintenance station. These control messages are received by the MQTT subscriber running on WS2, which writes the payloads into the PLC memory via the Snap7 interface. The PLC forwards the command to the Edge PC, which triggers ROS 2 navigation ; once executed, the AMR sends updated feedback via icps/robots, closing the synchronization loop.

This synchronization mechanism follows a TT approach, where the DES-based DT evaluates the operational state of AMRs at fixed time intervals. Continuous monitoring ensures that the DES-based DT maintains active control over the OMEs, enabling real-time decision-making and adaptive responses. This configuration achieves a bidirectional data flow, allowing the DES-based DT both to receive real-time feedback and to issue control commands that directly influence the behavior of the physical system.

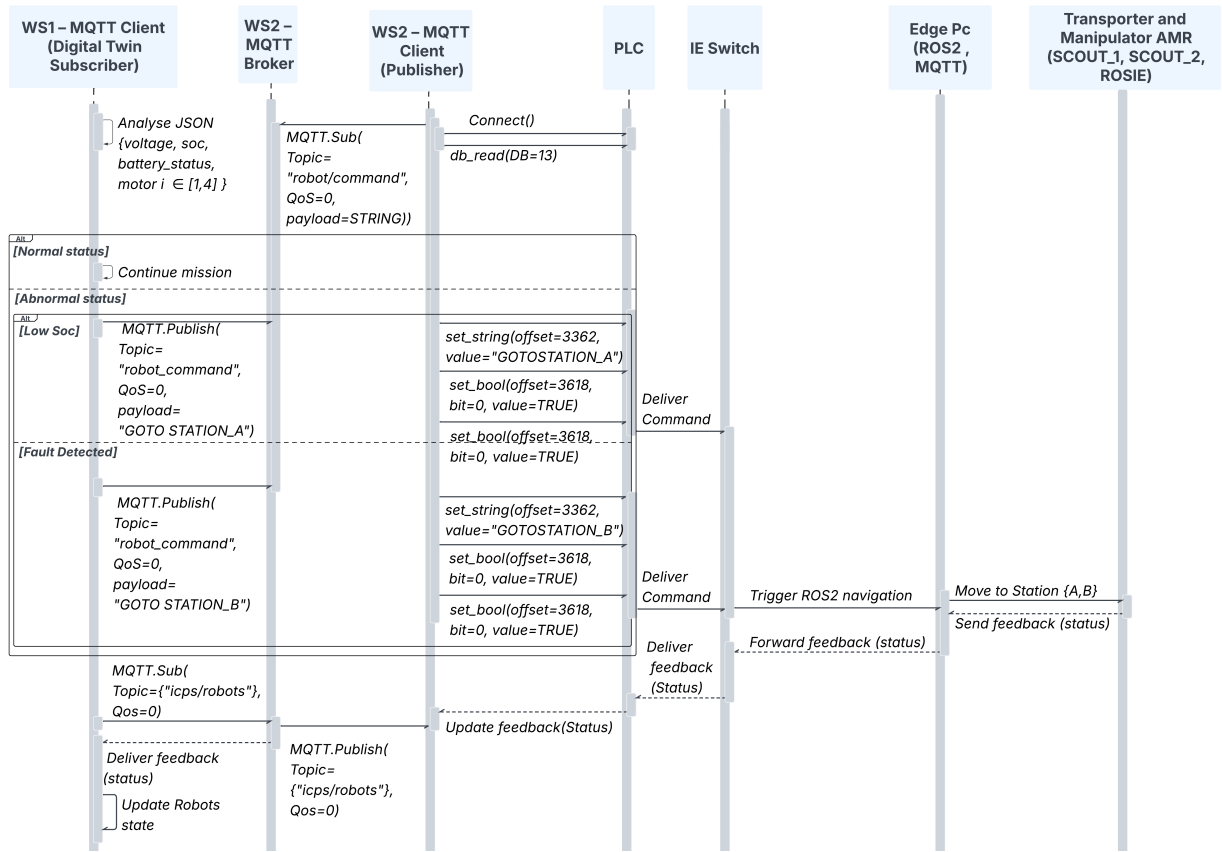


FIGURE 3.8 The UML sequence diagram illustrating the control response and bidirectional data flow between the DT and the physical system.

3.5 Experimentation and Result

This section presents the experimental evaluation to validate the proposed synchronization framework integrating TT and ET mechanisms within the DES-based DT. The framework is developed to enable the real-time dispatch of AMRs for maintenance and charging operations, where real-time decision feedback is required to avoid downtime. The experiments aim to assess the effect of the number of MQTT topics and the update interval (ΔT) on communication latency. The number of topics, representing the synchronization points of the DES-based DT, defines the active data streams between the physical system and the digital counterparts, while the update interval (ΔT) denotes the time between two consecutive message publications. The experiments analyze how these factors influence real-time synchronization and highlight the trade-off between communication load and system response time. A factorial Design of Experiment (DoE) was used to evaluate the effects of both factors on latency.

The following subsection presents the preliminary assessment of synchronization performance, focusing on the measurement of message latency between the DES and the physical system.

3.5.1 Experimental Assessment of Synchronization Performance

Preliminary latency tests were conducted to evaluate the synchronization performance between the DES-based DT and the OME, which aims to ensure the timely dispatch of the AMR for maintenance and charging operations. In this context, latency (L) refers to the elapsed time between message publication in the OME and its reception in the DES-based DT, i.e., between WS2 and WS1, defined as :

$$L = t_{\text{reception}} - t_{\text{emission}}$$

The latency quantifies the time required for data transmission between the OME and the digital counterparts. High latency affects the update of system states in the DES-based DT and the timing of command execution during AMR dispatch.

The experimental configuration included five MQTT topics summarized in Table 3.1, each representing a synchronization point between the OME and the digital counterparts. To simplify reference throughout this section, these topics are identified by their index from Topic 1 to Topic 5. In the preliminary tests, Topic 1 served as the baseline configuration. The remaining topics from Topics 2 to 5 were then incrementally activated to evaluate how the increasing number of synchronization points affects message latency.

TABLE 3.1 MQTT Topics Defining the Synchronization Points of the DES-based DT

Topics index	Name	Function
<i>Topic 1</i>	icps/rfid	Sends RFID tag data for product tracking.
<i>Topic 2</i>	plc/sensors	Reports object presence from 18 proximity sensors.
<i>Topic 3</i>	icps/qrcode	Transmits decoded QR code information.
<i>Topic 4</i>	icps/robots	Publishes AMR status, position, and errors.
<i>Topic 5</i>	icps/inspection	Provides inspection triggers and results.

As depicted in Figure 3.9, the latency of Topic 1 increases progressively as additional topics are activated from one to five. When Topic 1 operates alone, the average latency remains around 50 ms. With two active topics (Topics 1 and 2), the latency of Topic 1 rises to approximately 100 ms, indicating a moderate increase. However, once three or more topics are simultaneously active, latency begins to grow exponentially, reflecting congestion and queuing effects at the MQTT broker. At five active topics, the latency of Topic 1 exceeds 10^4 ms, demonstrating a significant degradation in synchronization performance. Each curve corresponds to 200 observations, representing 200 messages exchanged between the DES-based DT and the OME through the MQTT topics during the tests. The results show that as more OMEs are synchronized simultaneously, the update frequency of the DES-based DT decreases, leading to a delay in the representation of the physical system state. To quantitatively assess this effect, the next section presents a DoE evaluating the influence of the number of topics and the update interval (ΔT) on latency.

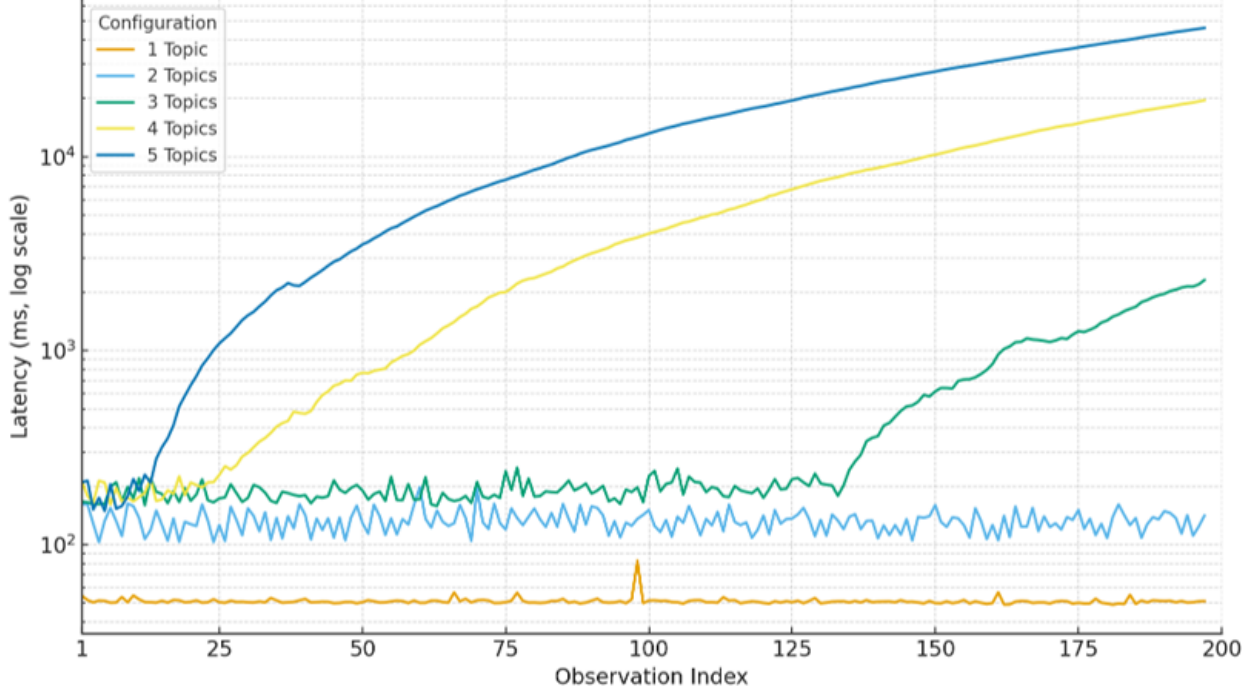


FIGURE 3.9 The message latency for Topic 1 as additional topics are incrementally activated

3.5.2 Experimental Evaluation of the Number of Topics and Update Frequency on Latency

In order to examine the effect of the number of MQTT topics and the update interval (ΔT) on latency, a factorial DoE was implemented. Latency (L) was selected as the response variable. Two independent factors were considered :

- **Factor 1 (Number of Topics, F_1)** : five levels corresponding to the incremental activation of Topics 1–5, as defined in Table 3.1
- **Factor 2 (Update Interval, F_2)** : comprised two levels defining the time between consecutive message publications, with ($\Delta T = 0.01$ s) and ($\Delta T = 0.1$ s).

Topic 1 served as the baseline for all experimental runs, while Topics 2–5 were progressively activated to increase the communication load and assess its influence on latency. Each experiment was executed with three independent replications, resulting in a total of $N = 5 \times 2 \times 3 = 30$ experimental runs. All tests were performed under identical network conditions using a local MQTT broker hosted on WS2 and a subscriber on WS1. The network speed during the tests was 93.6 Mbit/s download, 93.5 Mbit/s upload, and 1 ms latency, measured on the Montreal server. These network characteristics ensured that external communication delays were negligible and allowed the measured latency to reflect the internal communication

and processing times within the DES-based DT architecture.

A two-way Analysis of Variance (ANOVA) was performed to quantify the effects of the number of MQTT topics and the update interval (ΔT) on latency within the real-time DES-based DT framework. In addition to assessing the main effects, the test examined the interaction between both factors to determine whether the impact of topic count varied with the update interval. Two main factors were considered : the number of MQTT topics and the update interval (ΔT), as well as their interaction. The null hypothesis (H_0) state that variations in these factors have no significant impact on latency, whereas the alternative hypothesis (H_1) posit that each factor and their combined interaction affect latency. Testing these hypotheses allows identifying whether increasing the number of active topics, reducing the update interval, or the combination of both leads to statistically significant changes in message transmission delay.

Table 3.2 presents the results of the two-way ANOVA assessing the effects of the number of MQTT topics and the update interval (ΔT) on message latency. All p -values are below the 0.05 significance threshold, leading to the rejection of the null hypotheses. These results confirm that latency is significantly affected by the number of active topics and the update interval, and that their combined influence amplifies communication delays. Within the DES-based DT context, this indicates that simultaneous activation of multiple synchronization points, represented by MQTT topics, leads to increased network congestion and latency, which directly impacts synchronization between the physical and the digital counterparts. Consequently, maintaining all OME under continuous TT updates would degrade the real-time alignment of the DT. These findings validate the need for the proposed selective synchronization mechanism.

TABLE 3.2 Two-way ANOVA results for latency

Factor	p-value	Interpretation
<i>Number of Topics</i>	< 0.05	H_0 rejected – significant effect
<i>Update Interval (ΔT)</i>	< 0.05	H_0 rejected – significant effect
<i>Topics $\times$ ΔT (interaction)</i>	< 0.05	H_0 rejected – significant interaction effect

Figures 3.10 below illustrate the influence of the number of MQTT topics, the update interval (ΔT), and their interaction on latency within the proposed DES-based DT. Figures 3.10(a)

reveals an exponential increase in latency as the number of active topics grows. When a single topic is active, the average latency remains below 70 ms, while activating all five topics increases the latency to approximately 24.6 s at ($\Delta T = 0.01$ s). This escalation demonstrates that concurrent topic activation imposes processing and queuing loads on the MQTT broker, thereby extending message dispatch time and reducing communication efficiency. Figures 3.10(b) illustrates the boxplots of the effect of the update interval. Reducing (ΔT) from (0.1 s) to (0.01 s) causes the average latency to rise from around 16.2 s to 24.6 s at five topics, reflecting the significant impact of shorter intervals on communication stability. The higher publication frequency overloads the broker and increases packet queuing, leading to wider latency dispersion across trials.

Figures 3.10(c) presents the interaction effect between the number of topics and the update interval. For a limited number of topics (one or two), latency remains below 150 ms regardless of (ΔT). However, as the number of topics exceeds three, the difference between both update intervals becomes significant : at five topics, latency reaches 24.6 s for ($\Delta T = 0.01$ s), compared to 16.2 s for ($\Delta T = 0.1$ s).

These results quantitatively demonstrate that simultaneous activation of all MQTT topics degrades synchronization performance and temporal consistency within the DES-based DT. Continuous TT updates for all the OME would therefore compromise real-time operation due to high latency. For this reason, the proposed synchronization mechanism assigns TT synchronization to Topic 4, corresponding to data from the AMRs, which requires frequent data exchange to enable the DES-based DT to generate maintenance and charging decisions that are then transmitted as feedback or commands to the corresponding OME. In contrast, topics associated with monitoring and identification, such as Topic 1, Topic 2, Topic 3, and Topic 5, operate under an ET mode, as their role is to supervise and their updates occur less frequently.

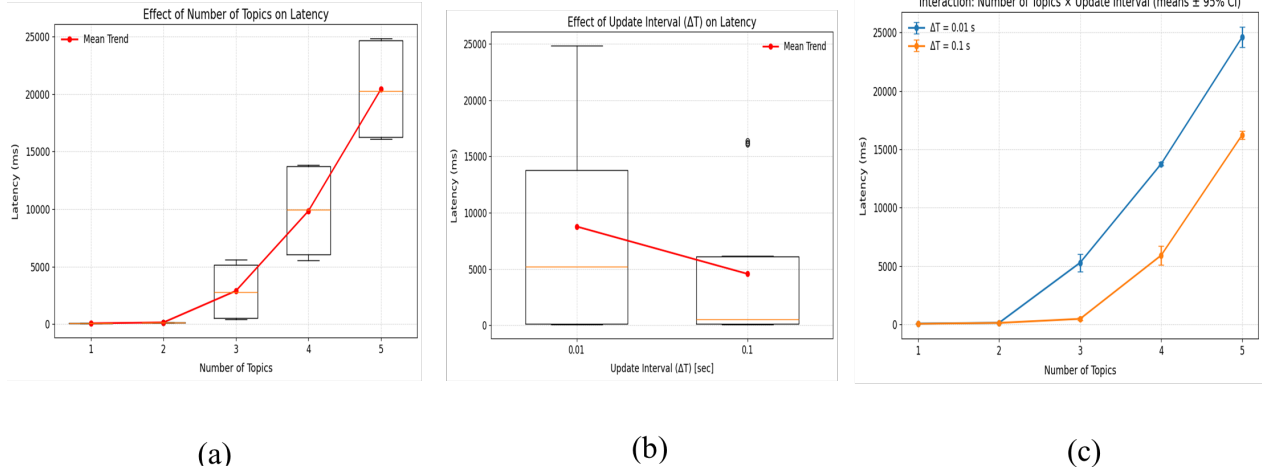


FIGURE 3.10 The effect of number of MQTT topics on message latency (mean trend across five topic levels); (b) The effect of update interval (T) on message latency (comparison of $T = 0.01$ s and $T = 0.1$ s); (c) The interaction effect between number of topics and update interval.

The following section presents the experimental results obtained from the implementation of the proposed synchronization mechanism. A selective synchronization approach was applied to mitigate the latency increase observed when all OMEs operate under continuous updates. The next subsection evaluates this configuration by combining TT and ET mechanisms and analyzing their impact on synchronization performance.

3.5.3 Experiments under selective TT and ET mechanism

The experimental validation of the proposed synchronization mechanism aimed to evaluate its effect on latency for data exchange within the DES-based DT. Two configurations were assessed : a baseline configuration in which all OME were synchronized through a TT mechanism, and the proposed configuration combining TT and ET synchronization mechanisms according to the operational role of each OME.

Figure 3.11 presents the latency evolution for the baseline configuration in which all OME were synchronized through a TT mechanism with a fixed publication interval of ($\Delta T = 0.1$ s). A total of 2000 observations were recorded, corresponding to 2000 MQTT messages published and received per topic. During the first 250 transmissions, the latency remained within a few hundred milliseconds for the five topics. Beyond this point, latency increased exponentially, rising from approximately 10^2 ms to 10^4 ms (≈ 10 s) as message queuing accumulated in the MQTT broker. This represents an increase of nearly two orders of magnitude in latency, indicating that the network becomes saturated when all OMEs transmit data continuously,

causing delayed message delivery and a loss of temporal consistency between the physical and the digital counterparts. The figure therefore highlights the limitation of a fully TT synchronization mechanism, where constant and simultaneous updates from multiple devices introduce cumulative delays that impair the DES-based DT ability to maintain real-time synchronization.

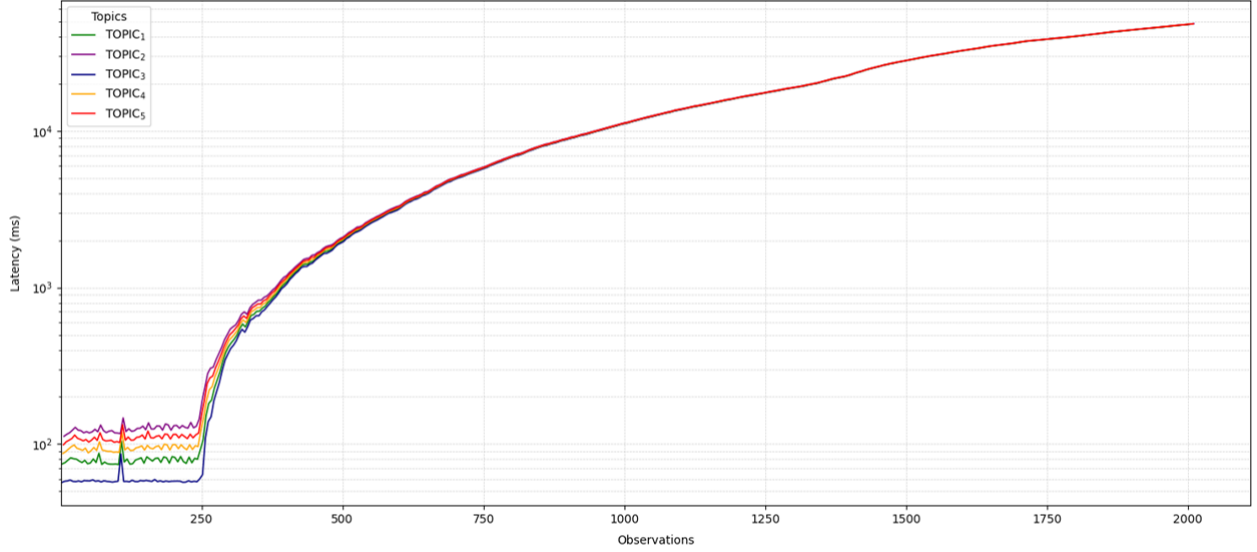


FIGURE 3.11 The baseline configuration for all OMEs synchronized through a TT mechanism ($\Delta T = 0.1$ s), latency shown on a logarithmic scale.

Figure 3.12 presents the latency results obtained after applying the proposed synchronization mechanism. In this configuration, Topic 4 is synchronized through a TT mechanism with a fixed update interval of ($\Delta T = 0.1$ s), while Topic 1, Topic 2, Topic 3, and Topic 5 are synchronized through an ET mechanism, in which data are transmitted only when a change in the sensor state is detected. The total number of recorded transmissions remained identical to the baseline configuration, equal to 2000 observations. Under this configuration, the latency of Topic 4 remains below 175 ms, while the maximum latency among the ET topics reaches 225 ms for Topic 5. Compared with the baseline latency of approximately 10^4 ms, this represents a reduction of two orders of magnitude, with latency values now on the order of 10^2 ms.

The results indicate that the proposed synchronization mechanism reduces latency. By associating the synchronization mechanism with the operational role of each OME, the DES-based DT maintains continuous updates for Topic 4, required for dispatch decision generation related to maintenance and charging, and transmits monitoring and identification data only upon state change. This configuration limits redundant transmissions, lowers

communication delays, and enables the DES-based DT to achieve real-time synchronization with the physical system.

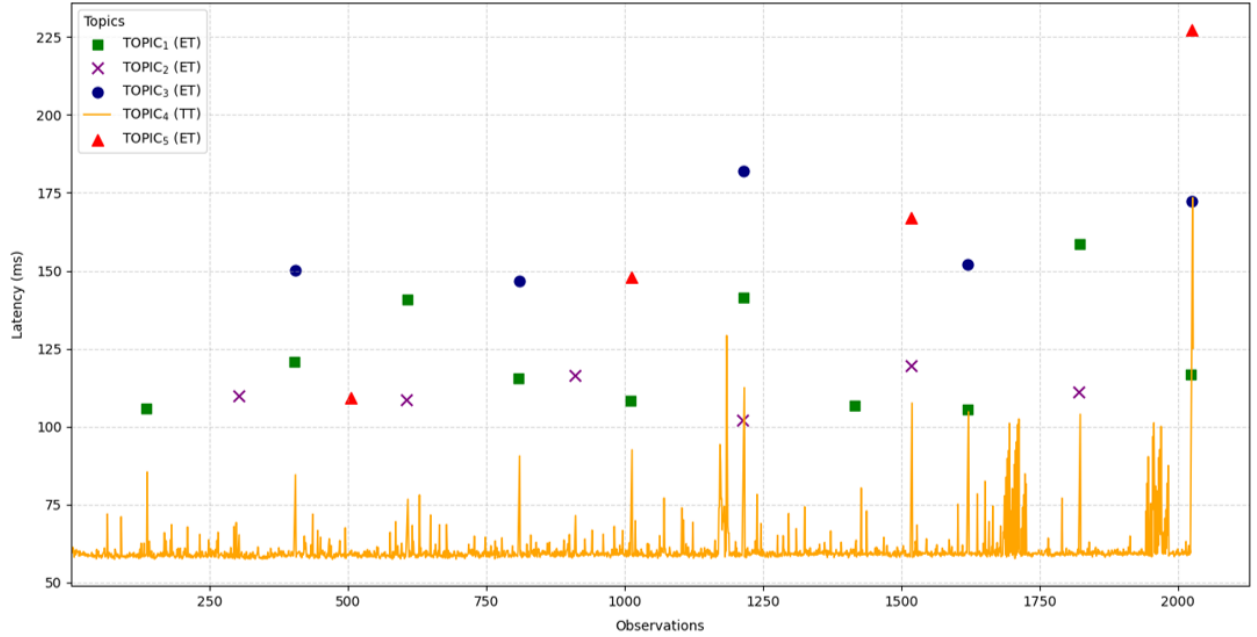


FIGURE 3.12 The proposed configuration for topic4 synchronized through a TT mechanism ($\Delta T = 0.1$ s); other OMEs synchronized through ET mechanisms.

3.6 Conclusion

This study proposes a synchronization framework that enables real-time dispatching of AMRs for maintenance and charging operations through a DES-based DT. The proposed framework formalizes two complementary synchronization mechanisms, the TT and ET, within a unified framework that links each data flow to its operational role. This configuration enables selective synchronization of the OMEs, ensuring temporal coherence while reducing communication latency.

The implementation carried out at the I-CPS Laboratory validated the operational relevance of the proposed approach in a smart warehouse environment. The experimental design demonstrated that both the number of MQTT topics and the update interval (ΔT) significantly affect communication latency, and that their combined variation amplifies this effect. Results showed that synchronizing all OMEs through a fully TT mechanism led to latency escalation of up to two orders of magnitude due to broker queuing and congestion. Conversely, selective synchronization reduced latency from approximately 10^4 ms to 10^2 ms while maintaining the temporal coherence of the DES-based DT and keeping the dispatch of

AMRs for maintenance and charging operations aligned.

These findings demonstrate that the proposed synchronization framework provides a structured foundation for latency-aware operation in DES-based DTs. By formalizing synchronization points and quantifying the relationship between communication parameters and temporal behavior, the study advances the design of real-time DES-based DT architectures capable of maintaining stable alignment between the physical system and the digital counterpart. Although the validation employed MQTT as the communication protocol, the framework remains applicable to other protocols, emphasizing that synchronization design is as critical as transmission speed for achieving real-time DT performance.

Future developments could focus on enhancing the decision-making capability of the DES-based DT by integrating intelligent algorithms, such as a reinforcement learning algorithm, to enable adaptive synchronization and control based on real-time operational data.

CHAPITRE 4 DISCUSSION GÉNÉRALE

La réalisation de ce travail a permis de démontrer la faisabilité et l'intérêt d'un jumeau numérique (JN) basé sur une simulation à événements discrets (SED) pour soutenir la prise de décision en temps réel dans un environnement logistique robotisé. L'intégration d'un mécanisme de synchronisation combinant des points Time-Triggered (TT) et Event-Triggered (ET) constitue l'une des principales contributions. Les résultats expérimentaux obtenus au laboratoire SCP-I montrent que la synchronisation sélective limitant l'usage du TT aux actions critiques et privilégiant l'ET pour le suivi des états réduit significativement la latence de communication entre le système physique et sa réplique virtuelle. Cette diminution de latence améliore la capacité du JN à refléter fidèlement l'état du système, condition essentielle pour une prise de décision adaptative et continue.

Les expérimentations ont également mis en lumière l'impact du nombre de topics MQTT et de leur fréquence d'actualisation sur les performances temporelles. L'augmentation du volume de données publiées simultanément provoque une croissance exponentielle de la latence, nuisant à l'alignement temporel avec le système réel. Ces observations confirment que la conception d'un JN ne dépend pas uniquement du modèle virtuel ou des capacités de simulation, mais aussi de l'architecture de communication et de la gestion des flux de données. Le JN doit donc intégrer des mécanismes de synchronisation adaptés pour préserver la cohérence temporelle entre le système physique et sa réplique virtuelle.

Sur le plan opérationnel, le JN développé montre sa capacité à supporter des décisions liées à la gestion des interruptions : en particulier les cycles de recharge et la détection des pannes. En recueillant les données de batterie, d'état moteur et de signaux de défaut, le JN permet de reconfigurer les missions des robots. Cette logique adaptative constitue une réponse aux limites des approches traditionnelles, souvent statiques ou centrées sur des hypothèses fixes. Ce résultat s'inscrit dans la littérature qui souligne que les JN à base de SED offrent un potentiel pour la gestion dynamique des ressources dans les entrepôts automatisés, tout en montrant la rareté de travaux intégrant une synchronisation mesurée et validée expérimentalement.

Au-delà de ces apports, certaines limites doivent être soulignées. Le mécanisme de synchronisation repose exclusivement sur MQTT, alors que d'autres protocoles industriels pourraient être explorés pour des niveaux de criticité plus élevés. De plus, le système physique du SCP-I constitue un environnement contrôlé : l'extension vers des systèmes industriels de grande échelle nécessitera une évaluation de la robustesse du modèle face à des charges de

données plus importantes, à une variabilité accrue des comportements et à des contraintes de sécurité informatique. Enfin, le modèle décisionnel intégré dans le JN demeure basé sur des règles déterministes ; l'intégration future de modèles prédictifs ou d'apprentissage pourrait accroître la capacité du système à anticiper les interruptions et à optimiser les trajectoires opérationnelles.

En somme, ce travail confirme que l'association d'un modèle SED, d'une architecture de communication conforme à ISO 23247 et d'un mécanisme de synchronisation sélectif constitue une voie prometteuse pour la conception de JN industriels capables d'agir en temps réel. Les résultats expérimentaux valident la pertinence de cette approche et ouvrent la voie à un déploiement progressif vers le partenaire industriel, où le JN pourra soutenir la gestion adaptative des opérations de maintenance et de recharge dans des conditions réelles.

CHAPITRE 5 CONCLUSION

Dans ce chapitre, nous présentons une synthèse des travaux réalisés ainsi que les principaux résultats obtenus. Nous y discutons également les limites de la solution proposée et les perspectives d’amélioration pour les recherches futures.

5.1 Synthèse des travaux

Ce mémoire présente le développement et la validation d’un JN fondé sur SED au sein du laboratoire des systèmes SCP-I de Polytechnique Montréal, en collaboration avec un partenaire industriel spécialisé dans la maintenance de véhicules électriques. L’objectif du travail était de réduire les temps d’arrêt liés aux opérations de maintenance et de recharge en exploitant la technologie des JN pour générer des décisions d’expédition en temps réel.

Le cadre proposé repose sur une architecture de synchronisation en temps réel assurant la cohérence des échanges de données entre le système physique et sa réplique numérique. Deux mécanismes complémentaires y sont intégrés : la synchronisation périodique TT pour le pilotage des opérations et la synchronisation déclenchée par événement ET pour la surveillance et la détection des changements d’état. Cette structure, adaptée à la norme (ISO 23247-2,2021), définit les interactions entre les entités physiques et numériques afin de garantir un échange bidirectionnel des données.

L’implémentation a été réalisée dans le laboratoire SCP-I, qui regroupe plusieurs robots mobiles, capteurs et automates connectés. Le modèle SED reçoit en continu les données de fonctionnement, notamment le niveau de charge et les signaux de défaillance, puis envoie les décisions correspondantes pour l’expédition des AMR vers les stations de maintenance ou de recharge selon leur état. Ces expérimentations ont permis d’évaluer l’effet du nombre de points de synchronisation et de l’intervalle de mise à jour sur la latence des messages entre le SED et le système physique, paramètre essentiel pour la réactivité du JN. Les résultats ont montré que la synchronisation sélective combinant les mécanismes TT et ET réduit significativement la latence tout en maintenant la cohérence temporelle entre le système physique et sa réplique numérique.

Ce travail a conduit à la mise en place d’un mécanisme de synchronisation formalisé pour les JN à base de SED et à sa validation expérimentale dans un environnement connecté représentatif des systèmes industriels. Les résultats confirment que le mécanisme développé permet de soutenir la génération de décisions d’expédition pour la maintenance et la recharge,

contribuant ainsi à la réduction des temps d'arrêt et à l'amélioration de la continuité opérationnelle.

5.2 Limitations de la solution proposée

Les expériences ont été menées dans un environnement contrôlé du laboratoire SCP-I, où les conditions de réseau et de traitement des données étaient locales. Bien que ce cadre expérimental ait permis de valider le fonctionnement du JN et du mécanisme de synchronisation, il ne reproduit pas entièrement la variabilité d'un contexte opérationnel. Dans une application avec des véhicules électriques en déplacement, l'échange de données s'effectuera à distance sur un réseau mobile, ce qui peut introduire des latences plus élevées ou des pertes de paquets. De plus, les flux d'information proviendront de sources hétérogènes et réparties, ce qui nécessitera de vérifier la capacité du cadre à maintenir la synchronisation en temps réel. Ces aspects ne remettent pas en cause la validité du modèle, mais constituent les conditions à examiner lors du déploiement dans un environnement où les conditions varient.

5.3 Améliorations futures

Ce travail a été mené dans un environnement expérimental contrôlé au sein du Laboratoire des systèmes cyber-physiques intelligents (SCP-I) de Polytechnique Montréal, afin de valider la faisabilité technologique du JN proposé et de mesurer ses performances en matière de synchronisation en temps réel. L'ensemble des développements réalisés, incluant le développement du JN et l'implémentation du mécanisme de synchronisation (TT/ET), a été partagé avec le partenaire industriel, en vue d'une application à des véhicules électriques.

Les prochaines étapes de recherche consisteront à implémenter le cadre de synchronisation proposé dans un environnement industriel réel, connecté via un réseau 5G, afin de piloter la maintenance et la recharge prédictive d'une flotte de véhicules électriques. Cette phase permettra d'évaluer la robustesse du JN en conditions opérationnelles et de vérifier sa capacité à soutenir la prise de décision en temps réel.

Ces travaux visent à assurer la continuité entre la validation expérimentale réalisée au SCP-I et l'application industrielle du JN pour la gestion adaptative des opérations.

RÉFÉRENCES

- [1] S. Masuch, J. R. Landwehr, C. M. Flath et F. Thiesse, “The faster, the better? the impact of short delivery times on product returns in online retailing,” *Journal of Retailing*, vol. 100, n^o. 3, p. 475–485, 2024.
- [2] B. Wallner, B. Zwölfer, T. Trautner et F. Bleicher, “Digital twin development and operation of a flexible manufacturing cell using iso 23247,” *Procedia CIRP*, vol. 120, p. 1149–1154, 2023.
- [3] I. Lichtenstern et F. Kerber, “Data-based digital twin of an automated guided vehicle system,” dans *2022 Winter Simulation Conference (WSC)*. IEEE, 2022, p. 2936–2946.
- [4] S. Wu, W. Xiang, W. Li, L. Chen et C. Wu, “Dynamic scheduling and optimization of agv in factory logistics systems based on digital twin,” *Applied Sciences*, vol. 13, n^o. 3, p. 1762, 2023.
- [5] E. Tappia, E. Moretti, U. Mohring et I. Adan, “Part feeding scheduling for mixed-model assembly lines with autonomous mobile robots : benefits of using real-time data,” *International Journal of Computer Integrated Manufacturing*, vol. 38, n^o. 4, p. 485–500, 2025.
- [6] F. Abdoune, O. Cardin, M. Nouri et P. Castagna, “Real-time field synchronization mechanism for digital twin manufacturing systems,” *IFAC-PapersOnLine*, vol. 56, n^o. 2, p. 5649–5654, 2023.
- [7] R. Santos, J. Basto, S. Alcalá, E. Frazzon et A. Azevedo, “Industrial iot integrated with simulation-a digital twin approach to support real-time decision making,” dans *Proceedings of the international conference on industrial engineering and operations management*, 2019, p. 23–26.
- [8] L. Erdal, A. Gubartalla, P. V. Lopes, H. Cao, G. Shao, P. Lonnehed, H. Putto, A. Ahmed, S. Ekered et B. Johansson, “Integrating dynamic digital twins : enabling real-time connectivity for iot and virtual reality,” dans *2024 Winter Simulation Conference (WSC)*. IEEE, 2024, p. 2987–2998.
- [9] C. Human, A. H. Basson et K. Kruger, “Digital twin data pipeline using mqtt in sladta,” dans *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*. Springer, 2020, p. 111–122.
- [10] C. H. dos Santos, J. A. B. Montevechi, J. A. De Queiroz, R. de Carvalho Miranda et F. Leal, “Decision support in productive processes through des and abs in the digital

- twin era : a systematic literature review,” *International Journal of Production Research*, vol. 60, n°. 8, p. 2662–2681, 2022.
- [11] Z. U. Rizqi, S.-Y. Chou et W. N. Cahyo, “A simulation-based digital twin for smart warehouse :bridging the gap between discrete event simulation and digital twin : A manufacturing case study,” *Decision Analytics Journal*, vol. 12, p. 100509, 2024.
 - [12] E. Aretoulaki, S. T. Ponis, G. Plakas et D. Tzanetou, “Discrete event simulation and digital twins in warehouse logistics : a bibliometric and content analysis-based systematic literature review,” *International Journal of Computer Integrated Manufacturing*, vol. 37, n°. 10-11, p. 1376–1403, 2024.
 - [13] M. Catalano, A. Chiurco, C. Fusto, L. Gazzaneo, F. Longo, G. Mirabelli, L. Nicoletti, V. Solina et S. Talarico, “A digital twin-driven and conceptual framework for enabling extended reality applications : A case study of a brake discs manufacturer,” *Procedia Computer Science*, vol. 200, p. 1885–1893, 2022.
 - [14] A. Ferrari, G. Zenezini, C. Rafele et A. Carlin, “A roadmap towards an automated warehouse digital twin : current implementations and future developments,” *IFAC-PapersOnLine*, vol. 55, n°. 10, p. 1899–1905, 2022.
 - [15] A. Wooley, D. F. Silva et J. Bitencourt, “When is a simulation a digital twin? a systematic literature review,” *Manufacturing Letters*, vol. 35, p. 940–951, 2023.
 - [16] M. Müller, J. Mielke, Y. Pavlovskiy, A. Pape, S. Masik, T. Reggelin et S. Häberer, “Real-time combination of material flow simulation, digital twins of manufacturing cells, an agv and a mixed-reality application,” *Procedia CIRP*, vol. 104, p. 1607–1612, 2021.
 - [17] J. B. Mathews, J. Rachner, L. Kaven, D. Grunert, A. Göppert et R. H. Schmitt, “Industrial applications of a modular software architecture for line-less assembly systems based on interoperable digital twins,” *Frontiers in Mechanical Engineering*, vol. 9, p. 1113933, 2023.
 - [18] F. Tao, Q. Qi, L. Wang et A. Nee, “Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0 : Correlation and comparison,” *Engineering*, vol. 5, n°. 4, p. 653–661, 2019.
 - [19] S. Yacout, “Simulation des systèmes à événements discrets,” Cours IND8204 - , Polytechnique Montréal, 2024.
 - [20] A. Hyre, G. Harris, J. Osho, M. Pantelidakis, K. Mykoniatis et J. Liu, “Digital twins : representation, replication, reality, and relational (4rs),” *Manufacturing Letters*, vol. 31, p. 20–23, 2022.
 - [21] J. Osho, A. Hyre, M. Pantelidakis, A. Ledford, G. Harris, J. Liu et K. Mykoniatis, “Four rs framework for the development of a digital twin : The implementation of

- representation with a fdm manufacturing machine,” *Journal of Manufacturing Systems*, vol. 63, p. 370–380, 2022.
- [22] A. Kampa, “Modeling and simulation of a digital twin of a production system for industry 4.0 with work-in-process synchronization,” *Applied Sciences*, vol. 13, n°. 22, p. 12261, 2023.
 - [23] N. Zhang, R. Bahsoon, N. Tziritas et G. Theodoropoulos, “A digital twin approach for adaptive compliance in cyber-physical systems : Case of smart warehouse logistics,” *arXiv preprint arXiv :2310.07116*, 2023.
 - [24] MQTT Org, “Mqtt,” <https://mqtt.org/>, 2025, consulté le 14 août 2025.
 - [25] H. Marah et M. Challenger, “Madtwin : a framework for multi-agent digital twin development : smart warehouse case study,” *Annals of Mathematics and Artificial Intelligence*, vol. 92, n°. 4, p. 975–1005, 2024.
 - [26] D. R. Silva, G. M. Oliveira, I. Silva, P. Ferrari et E. Sisinni, “Latency evaluation for mqtt and websocket protocols : an industry 4.0 perspective,” dans *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018, p. 01 233–01 238.
 - [27] OPC Foundation, “What is opc ?” <https://opcfoundation.org/about/what-is-opc/>, 2025, consulté le 14 août 2025.
 - [28] C. Guo, X. Li, L. Zhao et J. Yan, “Establishment and implement of digital twin model for an intelligent workshop,” dans *Industrial Engineering and Applications*. IOS Press, 2023, p. 480–488.
 - [29] InfluxData, “Influxdb - time series database platform,” <https://www.influxdata.com/>, 2025, consulté le 14 août 2025.
 - [30] MongoDB Inc., “Mongodb manual,” <https://www.mongodb.com/docs/manual/>, 2025, consulté le 14 août 2025.
 - [31] Siemens, *TIA Portal Data Block*, 2025, documentation technique. [En ligne]. Disponible : <https://support.industry.siemens.com>
 - [32] PROFIBUS & PROFINET International, “Profinet technology description,” <https://www.profinet.com/profinet-explained/technology-description>, consulté le 15 août 2025.
 - [33] Open Source Robotics Foundation, “Why ros ?” <https://www.ros.org/blog/why-ros/>, consulté le 15 août 2025.
 - [34] M. Bozdal, M. Samie et I. Jennions, “A survey on can bus protocol : Attacks, challenges, and potential solutions,” dans *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. IEEE, 2018, p. 201–205.

- [35] D. An, H. Joo et H. Kim, “Enabling low-latency digital twins for large-scale uav networks using mqtt-based communication framework,” *ICT Express*, vol. 11, n^o. 2, p. 264–269, 2025.
- [36] A. D. Elbouzidi, R. Pellerin, A. Ait El Cadi, S. Lamouri et S. Boubaker, “Digital twin modeling framework for manual warehouses,” dans *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2023, p. 3528–3533.
- [37] The AnyLogic Company, “Anylogic help – documentation officielle,” <https://anylogic.help/anylogic/index.html>, consulté le 15 août 2025.
- [38] T. Raamets, K. Karjust, A. Hermaste et K. Mahmood, “Planning and acquisition of real-time production data through the virtual factory in chemical industry,” dans *ASME International Mechanical Engineering Congress and Exposition*, vol. 85567. American Society of Mechanical Engineers, 2021, p. V02BT02A017.
- [39] H. Oukassi, A. Jaoua, S. Yacout, E. Negri et M. Jaoua, “Towards standardizing the integration of digital twins in manufacturing systems,” dans *2024 Winter Simulation Conference (WSC)*. IEEE, 2024, p. 2939–2950.
- [40] *Automation systems and integration — Digital twin framework for manufacturing — Part 1 : Overview and general principles*, International Organization for Standardization Standard ISO 23 247-1 :2021, 2021, systèmes d’automatisation industrielle et intégration — Cadre technique de jumeau numérique dans un contexte de fabrication — Partie 1 : Vue d’ensemble et principes généraux.
- [41] *Automation systems and integration — Digital twin framework for manufacturing — Part 2 : Reference architecture*, International Organization for Standardization Standard ISO 23 247-2 :2022, 2022.
- [42] *Automation systems and integration — Digital twin framework for manufacturing — Part 3 : Digital representation of manufacturing elements*, International Organization for Standardization Standard ISO 23 247-3 :2022, 2022.
- [43] *Automation systems and integration — Digital twin framework for manufacturing — Part 4 : Information exchange*, International Organization for Standardization Standard ISO 23 247-4 :2022, 2022.
- [44] *Automation systems and integration — Digital twin framework for manufacturing — Part 5 : Digital thread for digital twin*, International Organization for Standardization Draft International Standard ISO/DIS 23 247-5, 2025. [En ligne]. Disponible : <https://www.iso.org/standard/87425.html>
- [45] S. Wang, T. Fu, P. Luo et A. Wang, “Multi agv simulation system of intelligent workshop based on digital twin,” dans *2021 4th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*. IEEE, 2021, p. 325–329.

- [46] I. C. Fantozzi, A. Santolamazza, G. Loy et M. M. Schiraldi, “Digital twins : Strategic guide to utilize digital twins to improve operational efficiency in industry 4.0,” *Future Internet*, vol. 17, n^o. 1, p. 41, 2025.
- [47] I. J. Akpan et G. E. Etti, “Computer simulation everywhere : Mapping fifteen years evolutionary expansion of discrete-event simulation and integration with digital twin and generative artificial intelligence,” *Symmetry*, vol. 17, n^o. 8, p. 1272, 2025.
- [48] A. Jaoua, S. Masmoudi et E. Negri, “Digital twin-based reinforcement learning framework : application to autonomous mobile robot dispatching,” *International Journal of Computer Integrated Manufacturing*, vol. 37, n^o. 10-11, p. 1335–1358, 2024.
- [49] B. Tan et A. Matta, “The digital twin synchronization problem : Framework, formulations, and analysis,” *IISE Transactions*, vol. 56, n^o. 6, p. 652–665, 2024.
- [50] B. Wang, F. Tao, X. Fang, C. Liu, Y. Liu et T. Freiheit, “Smart manufacturing and intelligent manufacturing : A comparative review,” *Engineering*, vol. 7, n^o. 6, p. 738–757, 2021.
- [51] R. da Silva Mendonça, S. de Oliveira Lins, I. V. de Bessa, F. A. de Carvalho Ayres Jr, R. L. P. de Medeiros et V. F. de Lucena Jr, “Digital twin applications : A survey of recent advances and challenges,” *Processes*, vol. 10, n^o. 4, p. 744, 2022.
- [52] W. Kritzinger, M. Karner, G. Traar, J. Henjes et W. Sihn, “Digital twin in manufacturing : A categorical literature review and classification,” *Ifac-PapersOnline*, vol. 51, n^o. 11, p. 1016–1022, 2018.
- [53] R. Santos, H. Piqueiro, R. Dias et C. D. Rocha, “Transitioning trends into action : A simulation-based digital twin architecture for enhanced strategic and operational decision-making,” *Computers & Industrial Engineering*, vol. 198, p. 110616, 2024.
- [54] A. H. Sakr, A. Aboelhassan, S. Yacout et S. Bassetto, “Building discrete-event simulation for digital twin applications in production systems,” dans *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2021, p. 01–08.
- [55] L. Morabito, M. Ippolito, E. Pastore, A. Alfieri et F. Montagna, “A discrete event simulation based approach for digital twin implementation,” *IFAC-PapersOnLine*, vol. 54, n^o. 1, p. 414–419, 2021.
- [56] F. Coelho, S. Relvas et A. P. Barbosa-Póvoa, “Simulation-based decision support tool for in-house logistics : the basis for a digital twin,” *Computers & Industrial Engineering*, vol. 153, p. 107094, 2021.
- [57] M. Peyman, P. Copado, J. Panadero, A. A. Juan et M. Dehghanimohammadabadi, “A tutorial on how to connect python with different simulation software to develop rich

- simheuristics,” dans *2021 Winter Simulation Conference (WSC)*. IEEE, 2021, p. 1–12.
- [58] J. F. Leon, P. Marone, M. Peyman, Y. Li, L. Calvet, M. Dehghanimohammadabadi et A. A. Juan, “A tutorial on combining flexsim with python for developing discrete-event simheuristics,” dans *2022 Winter Simulation Conference (WSC)*. IEEE, 2022, p. 1386–1400.
- [59] A. Wooley, J. Bitencourt et D. Silva, “Bridging the gap between discrete event simulation and digital twin : A manufacturing case study,” *Manufacturing Letters*, vol. 44, p. 1274–1284, 2025.
- [60] H. Kopetz, *Real-time systems : design principles for distributed embedded applications*. Springer, 1997.
- [61] W. Alghamdi et E. Albassam, “Synchronization patterns for digital twin systems,” *Journal of Applied Data Sciences*, vol. 5, n^o. 3, p. 1026–1037, 2024.
- [62] B. C. Battissacco, W. Azzolini Júnior, C. H. dos Santos et A. L. Romano, “Discrete event simulation-based digital twin : an approach focused on tactical, strategic, and operational decisions,” *Journal of Simulation*, p. 1–19, 2025.
- [63] G. Molenaar et S. Preeker, “python-snap7 documentation,” 2023.
- [64] R. A. Light, “Mosquitto : server and client implementation of the mqtt protocol,” *Journal of Open Source Software*, vol. 2, n^o. 13, p. 265, 2017.

ANNEXE A CAS D'USAGE

TABLEAU A.1 Résumé du cas d'usage conforme à la norme ISO 23247.

ID	SCP-I
<i>Nom du cas d'usage</i>	JN pour la gestion adaptative de la recharge et de la maintenance des robots
<i>Domaine d'application</i>	Entrepôt intelligent
<i>Phase du cycle de vie couvertes</i>	Opération et maintenance
<i>Statut</i>	TRL4* : validation expérimentale en laboratoire
<i>Portée</i>	Gestion en temps réel des expéditions de recharge et de maintenance des AMR
<i>Objectifs</i>	Réduire les temps d'arrêt et améliorer la disponibilité des robots grâce à la synchronisation en temps réel entre le système physique et le JN
<i>Description courte</i>	Le JN surveille en temps réel l'état des robots. Lorsqu'un seuil de batterie est atteint ou qu'une anomalie est détectée, une instruction est générée pour rediriger automatiquement le robot vers la station de recharge ou de maintenance, sans interrompre les autres opérations. Ces actions sont modélisées dans AnyLogic et reproduites dans le système physique. L'échange bidirectionnel de données via MQTT assure la cohérence entre le modèle virtuel et le système physique.
<i>Parties prenantes</i>	Opérateurs d'entrepôt, ingénieurs de maintenance, intégrateurs systèmes, chercheurs
<i>Technologies clés</i>	Robots AMR (Mini-Scout 1 et 2, Rosie), MQTT, TIA Portal, Snap7, RFID, lecture QR, AnyLogic et Python
<i>Norme de référence</i>	ISO 23247 – Cadre de référence pour l'intégration des JN
<i>Besoin de normalisation</i>	Scalabilité et reproductibilité des JN basés sur SED

*Niveau de maturité technologique (Technology Readiness Level, ou TRL)

ANNEXE B PHASE 1 : TRANSFERT DU PRODUIT VERS LA ZONE DE RÉCEPTION

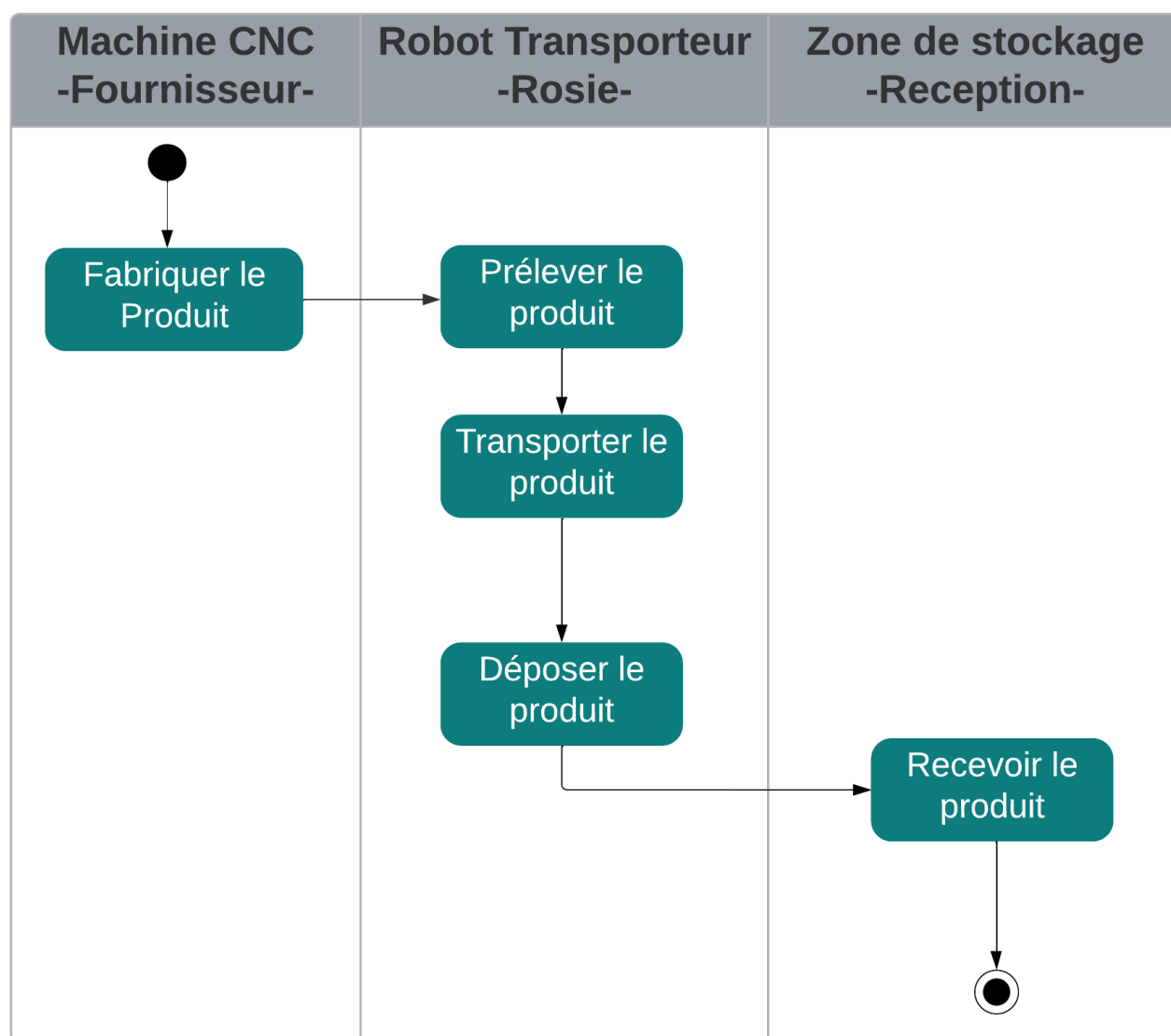


FIGURE B.1 Diagramme d'activité UML illustrant le transfert du produit de la machine CNC vers la zone de réception dans le SCP-I

ANNEXE C PHASE 2 : RÉCEPTION, IDENTIFICATION, INSPECTION ET EXPÉDITION DU PRODUIT

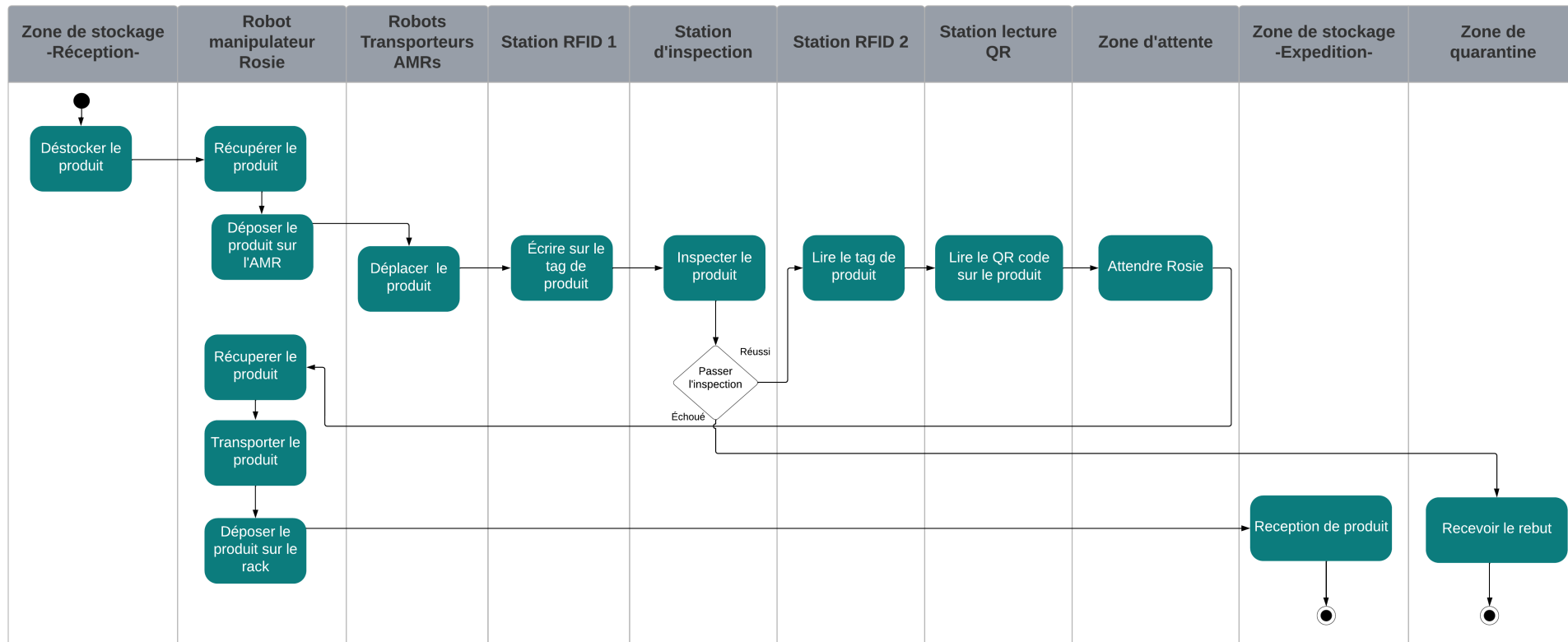


FIGURE C.1 Diagramme d'activité UML du processus de réception, d'identification, d'inspection et d'expédition des produits dans le SCP-I.