

Conception et développement du jumeau numérique pour démonstrateur de micro-réseau de bâtiment

Gerardo BECERRA¹, Adriana AGUILERA-GONZALEZ¹, Ionel VECHIU¹, Haider ALI², Bruno FRANCOIS³

¹Univ. Bordeaux, ESTIA-Institute of Technology, EstiaR, F-64210 Bidart, France

²ENSAM, L2EP, 59000 Lille, France

³Ecole Centrale de Lille, L2EP, 59000 Lille, France

RESUME – Les technologies émergentes telles que les micro-réseaux pour bâtiments et les jumeaux numériques jouent un rôle clé dans la transition vers un système énergétique durable. Ce travail présente les étapes suivies pour concevoir et développer un jumeau numérique destiné à un démonstrateur de micro-réseau pour bâtiments. La méthodologie décrite inclut la définition des exigences du jumeau numérique, la conception de son architecture, le développement de ses sous-systèmes, ainsi que les tests et la validation du système. Le jumeau numérique sera utilisé comme une plateforme de recherche innovante pour le développement et l'évaluation d'algorithmes avancés de surveillance, de prévision, d'optimisation, de contrôle et de diagnostic.

Mots-clés – *building microgrid, digital twin, system architecture, energy management system, fault detection and diagnosis.*

1. INTRODUCTION

Les bâtiments représentent une part significative de la consommation énergétique mondiale et des émissions de gaz à effet de serre, tout en étant largement inefficaces et dépendants du réseau principal [1]. La transition vers un système énergétique durable exige l'intégration et l'optimisation des énergies renouvelables dans les bâtiments. Les micro-réseaux pour bâtiments (BMG) sont une technologie prometteuse pour soutenir cette transition, permettant de maximiser l'autoconsommation grâce à l'utilisation d'énergie produite localement, et contribuer à la flexibilité du réseau principal. Le défi principal est de respecter les codes de réseau tout en assurant la fiabilité, la rentabilité et la durabilité du système.

Un système de gestion de l'énergie (EMS) efficace est essentiel pour optimiser les flux d'énergie dans un BMG, en tenant compte des contraintes techniques, économiques et réglementaires. Les approches hiérarchiques de gestion de l'énergie (HEMS) basées sur le contrôle prédictif par modèle (MPC) ont prouvé leur efficacité en gérant des exigences multiples et parfois conflictuelles, contrairement aux approches basées sur des règles fixes [2]. Les EMS MPC nécessitent des modèles précis pour prédire le comportement du BMG, avec des paramètres ajustés pour refléter fidèlement le système physique. Cependant, à long terme, des dégradations de certains composants, comme les batteries ou les panneaux PV, peuvent affecter la précision des prévisions. Un mécanisme d'adaptation automatique est donc nécessaire pour ajuster les modèles en fonction des écarts entre les comportements prévus et réels, tout en distinguant les écarts dus à des changements réels ou à des défaillances du système.

Le concept de jumeau numérique (DT) suscite récemment un intérêt croissant en tant que solution prometteuse pour relever les défis liés à l'exploitation optimale des microréseaux de bâtiments (BMGs). Un DT peut être défini comme une réplique numérique d'un système physique, utilisée pour la surveillance, la commande, l'optimisation et la prise de décision. En outre, grâce aux mesures et aux données estimées par les modèles embarqués dans le DT, un ajustement des points de fonctionnement

et des flux d'énergie est rendu possible pour maintenir une gestion énergétique optimisée malgré l'apparition d'incertitudes [3, 4].

Malgré son adoption croissante, le concept de DT est encore en évolution, et il n'existe pas de définition universellement acceptée ni de jeu standardisé d'attributs. Plusieurs études ont cherché à identifier les caractéristiques fondamentales des DTs. Par exemple, [5] a identifié des attributs clés tels que la synchronisation, l'apprentissage et l'adaptation, le flux d'information bidirectionnel, la surveillance, les capacités prédictives et prescriptives, ainsi que l'optimisation. Ces caractéristiques ont été proposées comme exigences fondamentales pour la conception de l'architecture d'un DT. [6] a présenté une architecture de DT pour la fabrication intelligente, basée sur une approche par microservices. Leur cadre intègre des composants tels que la surveillance, la gestion des événements, la simulation, la prise de décision, l'interopérabilité, l'analyse de données, le traitement de données et la cybersécurité. Les auteurs ont souligné l'importance des technologies open-source pour la mise en œuvre efficace de ces fonctionnalités. Pour approfondir la compréhension des DTs, [7] a défini sept catégories d'exigences fonctionnelles : modèle, données, algorithmes, connexion, interaction, simulation et visualisation. Ils ont proposé une architecture de DT détaillée, en décrivant les rôles et les interactions de chaque composant.

Plusieurs organisations et consortiums ont également contribué au développement et à la normalisation des DTs, tels que le Digital Twin Consortium et l'Industrial Digital Twin Association (IDTA). L'objectif principal de ces organisations est de favoriser la collaboration entre l'industrie, le milieu académique et le gouvernement afin d'établir des bonnes pratiques, des lignes directrices et des standards pour le développement et le déploiement des DTs. Elles ont publié plusieurs rapports comprenant des spécifications [8, 9] ainsi que des recommandations [10, 11] pour la mise en œuvre des DTs dans divers domaines d'application. Ces rapports soulignent l'importance de l'interopérabilité, de la sécurité et de la confidentialité des données, ainsi que la nécessité d'un cadre commun pour faciliter l'intégration des DTs avec les systèmes et technologies existants.

Des applications des DTs ont été rapportées dans plusieurs secteurs, notamment la logistique [12], la fabrication [13], la santé [14], l'énergie [15], les communications [16], les villes intelligentes [17], et les transports intelligents [18]. Dans le contexte spécifique des microgrid DTs (MGDTs), [19] identifie trois attributs fondamentaux : les modèles des systèmes et processus physiques, la connectivité des données en temps réel, et l'adaptation des modèles. Les auteurs explorent diverses applications potentielles des DTs pour les microréseaux, incluant la conception, la commande et l'exploitation, ainsi que la formation des opérateurs, la maintenance prédictive et prescriptive, et le diagnostic de pannes. Dans [20], un outil de gestion énergétique permet de modéliser les incertitudes liées à la production PV et de la demande des charges électriques en temps réel et de quantifier la réserve de puissance nécessaire à planifier sur les



FIG. 1. Démonstrateur BMG ESTIA Berri

flexibilités.

De nombreuses études dans la littérature prétendent développer des DTs, mais en pratique, elles mettent en œuvre des modèles de simulation validés à l'aide de données réelles. De notre point de vue, une caractéristique déterminante d'un véritable DT est la présence d'une connexion en temps réel entre la plateforme DT et le système physique. Cette connexion est essentielle pour permettre la formation d'un système composite doté de capacités dépassant celles des composants physiques ou numériques pris isolément. [21] souligne cette distinction en notant que, contrairement aux modèles de simulation traditionnels — qui restent abstraits par rapport au système physique — les DTs maintiennent un lien continu et en temps réel avec l'actif physique, leur permettant de réagir dynamiquement aux stimuli du monde réel.

Plusieurs études récentes ont commencé à démontrer des avancées vers des implémentations pratiques de MGDTS. [22] a développé un modèle de simulation pour un microréseau de bâtiment (BMG) intégré à des algorithmes d'optimisation et de prévision visant à minimiser les coûts énergétiques sur une période de 24 heures. Le modèle est validé à l'aide de données réelles du BMG, et une plateforme hardware-in-the-loop (HIL) est utilisée pour tester les algorithmes proposés. [23] présente un microréseau à l'échelle laboratoire conçu pour démontrer un système de gestion de l'énergie (EMS) destiné à une communauté énergétique peer-to-peer (P2P). Bien que les modèles du microréseau et de l'EMS aient été développés et validés en simulation, les auteurs reconnaissent qu'aucune connexion en temps réel avec le système physique n'a été établie, ce qui ne correspond donc pas à un DT pleinement réalisé. De manière similaire, [24] décrit un autre microréseau à l'échelle laboratoire, dans lequel des modèles basés sur les données pour des panneaux (PV) et des éoliennes ont été développés et validés avec des données réelles. Ce travail présente le développement d'un DT axé sur la maintenance prédictive et l'auto-guérison, en exposant des résultats préliminaires et en discutant des défis et considérations liés à la mise à l'échelle de l'implémentation pour des systèmes plus grands.

Dans ce travail, nous décrivons les étapes suivies dans la conception et le développement en cours d'un DT pour un démonstrateur réel de BMG. Notre contribution principale consiste en la description de la méthodologie, des considérations, des défis et des enseignements tirés à chaque étape du processus. De plus, la disponibilité du démonstrateur BMG nous permet de démontrer les capacités du DT développé sur un système réel actuellement en fonctionnement. Nous présenterons les exigences définies pour le DT, l'architecture proposée, les étapes de mise en œuvre de chaque sous-système ainsi que l'approche adoptée pour les tests et la validation.

2. MÉTHODOLOGIE POUR LA CONCEPTION ET LE DÉVELOPPEMENT D'UN DT

Le développement du DT a suivi un cycle itératif composé de quatre étapes fondamentales : définition des exigences, conception architecturale, développement, tests et validation. Chacune

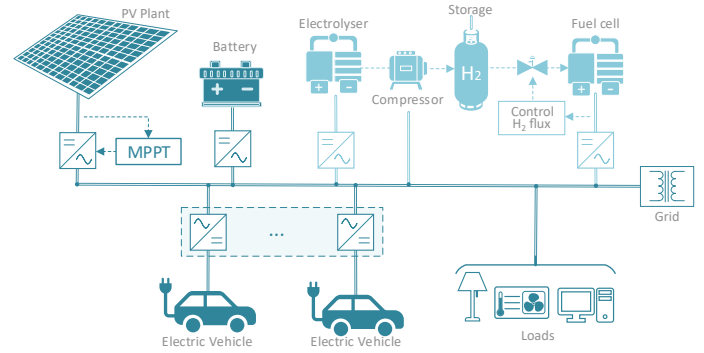


FIG. 2. Schéma simplifié du BMG

de ces étapes est décrite en détail dans les sous-sections suivantes.

2.1. Définition des Exigences du BMGDT

Le DT est développé pour le démonstrateur BMG ESTIA Berri au Technopôle Izarbel à Bidart, France. Ce bâtiment, illustré dans la Fig. 1, accueillant plus de 1 000 étudiants de l'École d'Ingénieurs ESTIA, dispose de 78,6 kWc de panneaux photovoltaïques (PV) sur le toit et 48,6 kWc sur un abri de parking, d'un système de stockage d'énergie à base de batteries (ESS) de 97 kWh, et de stations de recharge pour 6 voitures électriques (EV) et 45 vélos électriques. Le système sera prochainement étendu avec l'installation d'une chaîne hydrogène, qui sera utilisée pour le stockage d'énergie à long terme et pour alimenter une flotte de vélos à hydrogène. La consommation énergétique annuelle du bâtiment est d'environ 370 MWh, avec un objectif d'autoconsommation d'au moins 80%. Fig. 2 présente un schéma simplifié du BMG, illustrant les principales composantes et leur interconnexion.

Le démonstrateur a été conçu dès le départ comme un support expérimental pour la recherche sur l'intégration des énergies renouvelables dans les bâtiments, offrant ainsi un cas d'étude idéal pour des technologies comme le DT. Cela nous permet de tirer parti de l'infrastructure existante et des systèmes de collecte de données pour développer un DT capable de représenter fidèlement le système physique.

L'objectif principal du DT proposé pour le BMG (BMGDT) est de gérer efficacement l'exploitation du BMG en temps réel en optimisant différents objectifs économiques et techniques. Cet objectif est rendu possible grâce aux capacités de simulation du DT, qui permettent de prédire, d'analyser et d'optimiser le comportement du BMG sur différents horizons temporels. Nous avons défini les exigences fonctionnelles de notre DT en nous basant sur la classification proposée par [7], tout en adaptant ces exigences aux besoins spécifiques du démonstrateur BMG. Nous avons également défini des exigences non fonctionnelles axées sur les attributs de performance et de qualité du système. Ces exigences sont détaillées dans les sous-sections suivantes.

2.1.1. Exigences fonctionnelles du BMGDT

Modèle : Les modèles du DT représentent le système physique et ses composants. Ils sont utilisés pour réaliser des simulations, prédire le comportement et optimiser les performances. Le BMGDT doit être capable de :

- RF1.1 : Importer des modèles provenant de différentes sources.
- RF1.2 : Stocker les modèles de manière structurée et efficace.
- RF1.3 : Assembler les modèles des différents composants et sous-systèmes.
- RF1.4 : Tester et valider les modèles.
- RF1.5 : Adapter les modèles pour refléter les changements du système physique.

Données : Elles constituent l'atout principal du BMGDT. Les données sont utilisées pour paramétrer les modèles, entraîner les modèles, valider les algorithmes et surveiller le système physique. Le BMGDT doit être capable de :

- RF2.1 : Importer des données provenant de différentes sources.
- RF2.2 : Stocker les données de manière structurée et efficace.
- RF2.3 : Effectuer des traitements de données en temps réel et hors ligne, tels que le nettoyage, le filtrage, l'agrégation, la réduction de dimension et l'extraction de caractéristiques.
- RF2.4 : Générer des données représentant différentes conditions de fonctionnement ou scénarios.

Algorithmes : Les algorithmes encodent la logique et les règles utilisées pour traiter les données, exécuter des simulations et prendre des décisions. Le BMGDT doit être capable de :

- RF3.1 : Importer des algorithmes depuis différents dépôts ou bibliothèques.
- RF3.2 : Stocker les algorithmes de manière structurée et efficace.
- RF3.2 : Créer et modifier des algorithmes.
- RF3.3 : Tester et valider les algorithmes.
- RF3.4 : Exécuter les algorithmes en temps réel et hors ligne.

Connexion : La connexion entre le BMGDT et le système physique est cruciale pour l'échange de données en temps réel et la synchronisation. Le BMGDT doit être capable de :

- FR4.1 : Établir une connexion bidirectionnelle avec le système physique.
- FR4.2 : Assurer l'échange de données en temps réel entre le BMGDT et le système physique.
- FR4.3 : Garantir la sécurité et l'intégrité des données pendant la transmission.

Interaction : L'interaction entre les composants du BMGDT nécessite une coordination rigoureuse. Le BMGDT doit être capable de :

- FR5.1 : Garantir la synchronisation entre le BMGDT et le système physique.
- FR5.2 : Faciliter la communication entre les différents composants du BMGDT.
- FR5.3 : Détecter les conditions nécessitant l'exécution d'algorithmes ou de modèles spécifiques.

Simulation : La simulation est une fonctionnalité clé du BMGDT. Elle permet de représenter le comportement du système physique sous différentes conditions de fonctionnement. Le BMGDT doit être capable de :

- FR6.1 : Simuler le comportement du BMG.
- FR6.2 : Supporter différents modes de simulation (par exemple, temps réel, hors ligne).
- FR6.3 : Permettre des simulations avec différents scénarios ou configurations.
- FR6.4 : Stocker les résultats des simulations.

UI : Les capacités de visualisation du BMGDT permettent aux utilisateurs de surveiller le système, d'analyser les données et d'interpréter les résultats. Le BMGDT doit être capable de :

- FR7.1 : Fournir des outils de visualisation pour l'analyse et l'interprétation des données.
- FR7.2 : Supporter différents formats de visualisation (par exemple, graphiques, diagrammes, tableaux de bord).
- FR7.3 : Permettre aux utilisateurs de personnaliser les visualisations selon leurs préférences.

2.1.2. Exigences non fonctionnelles du BMGDT

Les exigences non fonctionnelles représentent les attributs de qualité du système, en mettant l'accent sur sa performance. Notre DT doit posséder les caractéristiques suivantes :

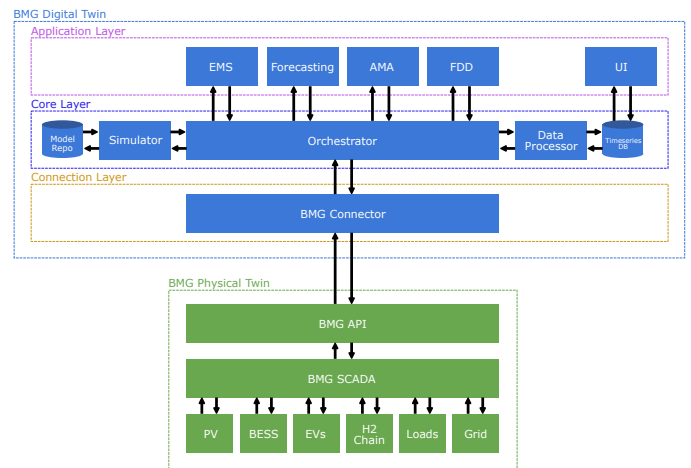


FIG. 3. Diagramme de l'architecture logicielle du DT conçu. Les flèches représentent le flux d'informations entre les sous-systèmes.

- NFR1 : Exactitude de modélisation supérieure à 95%.
- NFR2 : Traitement et visualisation des données en temps réel dans un délai de 1 seconde après la réception des mises à jour.
- NFR3 : Mise à jour des tableaux de bord utilisateur en moins de 5 secondes pour une expérience utilisateur fluide.
- NFR4 : Capacité de stockage pour conserver au moins 1 an de données.
- NFR5 : Temps d'exécution des algorithmes de simulation en temps réel inférieur à 1 seconde.
- NFR6 : Temps d'exécution des algorithmes de gestion énergétique, prévision, adaptation des modèles et détection de pannes inférieur à 1 minute.

2.2. Architecture du BMGDT

Une architecture logicielle offre une vue d'ensemble des composants et de leurs interactions, servant de plan directeur pour la conception et le développement du système. L'architecture de notre DT, illustrée dans la Fig. 3, reprend les principales fonctionnalités décrites dans [7], adaptées aux exigences spécifiques détaillées en 2.1. Elle se divise en plusieurs sous-systèmes, chacun chargé d'un ensemble de fonctionnalités précises.

BMG Connector : Responsable de la connexion entre le BMG et le DT, facilitant l'échange de données et la communication. Répond aux exigences FR4.1, FR4.2, FR4.3.

Orchestrator : Hub centrale des flux de données, responsable de la gestion des interactions entre les différents sous-systèmes et de l'assurance de leur synchronisation. Répond aux exigences FR5.1, FR5.2, FR4.1.

Data Processor : Effectue diverses tâches de traitement des données telles que le nettoyage, le filtrage, l'agrégation, la réduction de dimension et l'extraction de caractéristiques. Il génère également des données représentant différentes conditions de fonctionnement ou scénarios. Répond aux exigences RF2.1, RF2.2, RF2.3, RF2.4, RF3.1, NFR2.

Simulator : Exécute les modèles en temps réel, permettant la simulation du comportement du BMG sous différents scénarios. Répond aux exigences RF1.1, RF1.3, RF1.4, RF3.1, RF3.4, FR6.1, FR6.2, FR6.3, NFR5.

Model Repository : Stocke les modèles, y compris les modèles de simulation, de prévision et de détection de pannes. Répond aux exigences RF1.2.

Data Storage : Stocke les données en temps réel collectées par le Connecteur BMG, ainsi que les données de simulation générées par le Simulateur, en garantissant l'intégrité et la disponibilité des données. Répond aux exigences RF2.2, FR6.4.

EMS : Exécute l'optimisation du fonctionnement du BMG tout en respectant le code réseau relatif à l'autoconsommation. Répond aux exigences RF1.1, RF3.1, FR3.4, NFR6.

Forecasting : Prédit le comportement des variables climatiques et de la consommation énergétique. Répond aux exigences RF1.1, RF3.1, FR3.4, NFR6.

Automatic Model Adaptation (AMA) : Détecte les écarts entre le système physique et le DT. Exécute le processus d'ajustement des paramètres afin de garantir la précision du modèle. Répond aux exigences RF1.5, RF1.1, RF3.1, FR3.4, NFR6.

FDD : Détecte, diagnostique et signale les défauts des composants du BMG. Répond aux exigences RF1.1, RF3.1, FR3.4, NFR6.

UI : Permet de surveiller le fonctionnement du BMGDT, de visualiser les données historiques et de configurer les paramètres du système. Répond aux exigences FR7.1, FR7.2, FR7.3.

D'autres travaux ont proposé des architectures similaires pour les DT de MG. Dans [25], quatre applications des DT sont identifiées : contrôle et gestion énergétique en temps réel, protection, maintenance prédictive et prescriptive, ainsi que simulation et test en temps réel. Chaque application est associée à une architecture distincte, bien que les composants et leurs interactions ne soient pas décrits en détail. L'architecture liée à la simulation et aux tests en temps réel est celle qui se rapproche le plus de notre travail. [26] propose une architecture DT pour les bâtiments intelligents intégrant des modèles électriques et thermiques. Le système est structuré en trois couches : physique, simulation et plateforme logicielle. Bien que cette structuration soit proche de notre approche, nous intégrons la simulation dans la plateforme logicielle, en la considérant comme une fonction essentielle du DT. Les auteurs décrivent plus en détail les composants et valident l'architecture à travers des cas d'usage comme l'analyse de défauts capteurs et la gestion énergétique avec stockage par batterie. Une différence majeure réside dans le fait que leurs expérimentations sont réalisées sur une plateforme HIL, tandis que notre DT est déployé sur le démonstrateur BMG réel. Dans [27], l'architecture DT repose sur une orchestration de microservices pour le contrôle basé modèle, l'apprentissage sur données, la gestion des données, la simulation et la visualisation. Bien que cette répartition fonctionnelle ressemble à notre architecture, notre implémentation ne s'appuie pas sur des microservices. Les auteurs utilisent un modèle thermique simplifié du bâtiment et démontrent l'intégration d'un MPC avec des observateurs fondés sur modèle et données. Cependant, leur approche se concentre sur l'interaction simulée des composants plutôt que sur l'implémentation pratique et le déploiement du DT.

2.3. Développement du BMGDT

À partir de l'architecture proposée, nous avons commencé à développer les fonctionnalités de chaque sous-système. Les modèles du BMG ont été implémentés à l'aide de l'outil *Simscape Electrical* de *Simulink*. Cela inclut les modèles des PV et du ESS. Ces modèles ont été validés à l'aide de données historiques et en temps réel issues du démonstrateur BMG. Un bus DC a été utilisé pour modéliser les échanges de puissance active entre les différents éléments de la microgrid, incluant PV, ESS, EVs, charges et réseau. Ces modèles ont été exportés en tant qu'unités fonctionnelles de simulation (FMUs) et stockés dans le système de fichiers du BMGDT sous la forme d'un **Model Repository**, permettant au **Simulator** de les charger et de les exécuter. Les modèles de la chaîne hydrogène, comprenant l'électrolyseur, la pile à combustible et le stockage d'hydrogène, sont en cours de développement et seront validés une fois les équipements installés sur le démonstrateur BMG.

Le **Simulator** a été implémenté en Python, en s'appuyant sur la bibliothèque *FMPy* pour la gestion des modèles FMU. Cela inclut le chargement des FMU, leur initialisation, la configuration des paramètres, des entrées, l'exécution d'un pas de simula-

tion et la récupération des sorties. Le **Simulator** peut s'exécuter en temps réel, en utilisant comme entrées les données fournies par le **BMG Connector**, ou en mode hors-ligne, en utilisant des données historiques.

Pour faciliter l'interaction entre le DT et le BMG, une API a été développée au-dessus du système SCADA du BMG. Elle fournit des endpoints permettant de récupérer des données en temps réel, ainsi que d'envoyer des commandes et des consignes au BMG. Le **BMG Connector** a été développé en Python et utilise la bibliothèque *requests* pour communiquer avec l'API du BMG à une fréquence de 1 requête par seconde. Il implémente un mécanisme d'authentification pour assurer une communication sécurisée avec l'API du BMG. Il est également capable de gérer les erreurs et les tentatives de reconnexion en cas d'échec de communication. Le **Data Processor** a été implémenté en Python et utilise la bibliothèque *pandas* pour réaliser des tâches telles que la validation, le nettoyage et l'agrégation de données. Il peut récupérer des données historiques depuis la base de données ou des fichiers externes pour la simulation hors-ligne. Il collecte les données du **BMG Connector** et du **Simulator**, et les stocke dans une base de données de séries temporelles *TimescaleDB*. Il s'agit d'une extension de PostgreSQL optimisée pour les données de séries temporelles, permettant un stockage et une récupération efficaces de grands volumes de données. L'un de ses principaux avantages est de permettre l'analyse en temps réel à l'aide d'agrégats continus automatiquement mis à jour à mesure que de nouvelles données arrivent. Cela permet au sous-système **UI** de récupérer et d'afficher les données en temps réel ainsi que les indicateurs clés de performance (KPI). Dans ce cas, nous avons utilisé *Grafana* pour développer un prototype de tableau de bord. Le **Orchestrator** a été développé en Python et utilise la bibliothèque *asyncio* ainsi que la syntaxe *async/await* pour gérer les tâches asynchrones et assurer la synchronisation des différents sous-systèmes du BMGDT. En particulier, il permet une commande temporelle précise de l'exécution du **Simulator** afin de garantir son fonctionnement en temps réel, synchronisé avec le BMG.

Le **EMS** hiérarchique décrit dans [28] est actuellement en cours d'intégration dans le BMGDT. Il se compose d'un MPC économique de haut niveau (EMPC) et d'un MPC de suivi de bas niveau (TMPC). L'EMPC minimise le coût annuel estimé du BMG sur un horizon de 48 heures, en tenant compte du coût de l'électricité, de la dégradation de l'ESS et des incitations / pénalités liées à l'autoconsommation. Il génère des références pour l'état de charge (SoC) de la batterie et le niveau d'hydrogène (LoH), transmises ensuite au contrôleur de bas niveau. L'EMPC est en cours de mise en œuvre et inclura un module d'estimation des coûts annuels attendus du BMG et des taux d'autoconsommation. Le TMPC détermine les consignes de puissance pour la batterie et la chaîne hydrogène afin de suivre au plus près les références de SoC et de LoH. Il fixe également la puissance maximale produite par chaque chaîne PV pour éviter toute injection de puissance excédentaire dans le réseau.

Le problème d'optimisation est présenté dans les Eqs. (1) à (13), où $I_{bat,cha}$ et $I_{bat,dis}$ sont respectivement les courants de charge et de décharge de la batterie. Ces variables sont associées à $\delta_{bat,cha}$ et $\delta_{bat,dis}$, des variables binaires indiquant si la batterie est en charge ou en décharge à chaque pas de temps. P^{pv} est la puissance produite par les chaînes PV, P^{grid} la puissance prélevée sur le réseau, et P^{load} la puissance consommée par les charges. SoC^{bat} est le SoC de la batterie, calculé à partir de la charge de la batterie Q^{bat} , mise à jour à chaque pas de temps (Eq. (3)), et de la capacité de la batterie C^{bat} en Ah. L'objectif est de minimiser l'erreur quadratique entre SoC^{bat} et $SoC_k^{bat,ref}$ sur l'horizon de prédiction (Eq. (1)).

Les contraintes (5), (6) imposent respectivement les limites maximales de puissance de charge et de décharge de la batterie, tandis que les Eqs. (10), (11) garantissent que la batterie ne peut pas être chargée et déchargée simultanément. La contrainte

d'équilibre de puissance (Eq. (7)) garantit que la puissance fournie par les champs PV et le réseau est égale à celle consommée par les charges et injectée dans la batterie. Ici, la puissance de la batterie est calculée comme la différence entre les courants de charge et de décharge multipliée par la tension de la batterie V^{bat} . D'autres contraintes imposent des limites au SoC de la batterie (Eq. (8)), assurent des courants de charge et de décharge positifs (Eq. (9)), limitent la production PV (Eq. (12)), et empêchent l'injection au réseau (Eq. (13)). Ce problème d'optimisation est résolu sur un horizon de prédiction de 6 heures, avec un pas de temps configurable T_{TMPC} et un nombre de périodes N_{TMPC} . Actuellement, les modèles de la chaîne hydrogène n'étant pas encore implémentés, le TMPC ne traite que les consignes de puissance optimales pour la batterie et les chaînes PV.

$$\min_{\mathbf{I}_k^{\text{bat,cha}}, \mathbf{I}_k^{\text{bat,dis}}, \mathbf{P}_k^{\text{pv}}} \sum_{k=0}^{N_{\text{TMPC}}-1} \left[100 \left(\text{SoC}_k^{\text{bat}} - \text{SoC}_k^{\text{bat,ref}} \right) \right]^2 \quad (1)$$

$$\text{s.t. } \mathbf{Q}_0^{\text{bat}} = \mathbf{Q}_0 \quad (2)$$

$$\mathbf{Q}_{k+1}^{\text{bat}} = \mathbf{Q}_k^{\text{bat}} + T_{\text{TMPC}} \left(\mathbf{I}_k^{\text{bat,cha}} - \mathbf{I}_k^{\text{bat,dis}} \right) \quad (3)$$

$$\text{SoC}_k^{\text{bat}} = \frac{\mathbf{Q}_k^{\text{bat}} + T_{\text{TMPC}} \left(\mathbf{I}_k^{\text{bat,cha}} - \mathbf{I}_k^{\text{bat,dis}} \right)}{3600C^{\text{bat}}} \quad (4)$$

$$V^{\text{bat}} \mathbf{I}_k^{\text{bat,cha}} \leq \delta_k^{\text{bat,cha}} P^{\text{bat,cha,max}} \quad (5)$$

$$V^{\text{bat}} \mathbf{I}_k^{\text{bat,dis}} \leq \delta_k^{\text{bat,dis}} P^{\text{bat,dis,max}} \quad (6)$$

$$\mathbf{P}_k^{\text{pv}} + \mathbf{P}_k^{\text{grid}} = V^{\text{bat}} \left(\mathbf{I}_k^{\text{bat,cha}} - \mathbf{I}_k^{\text{bat,dis}} \right) + P_k^{\text{load}} \quad (7)$$

$$\text{SoC}_k^{\text{bat}} \in [\text{SoC}_{\text{min}}^{\text{bat}}, \text{SoC}_{\text{max}}^{\text{bat}}] \quad (8)$$

$$\mathbf{I}_k^{\text{bat,cha}} \geq 0, \quad \mathbf{I}_k^{\text{bat,dis}} \geq 0 \quad (9)$$

$$\delta_k^{\text{bat,cha}} + \delta_k^{\text{bat,dis}} \leq 1 \quad (10)$$

$$\delta_k^{\text{bat,cha}}, \delta_k^{\text{bat,dis}} \in \{0, 1\} \quad (11)$$

$$\mathbf{P}_k^{\text{pv}} \in [0, P_k^{\text{pv,max}}] \quad (12)$$

$$\mathbf{P}_k^{\text{grid}} \geq 0 \quad (13)$$

Le TMPC a été implémenté en Python en utilisant la bibliothèque `docplex` pour modéliser le problème d'optimisation, et `CPLEX Optimization Studio` pour le résoudre. Il a été testé à l'aide du modèle BMG, du **Simulator** et de données historiques chargées via le **Data Processor**, afin de valider ses performances. La Fig. 4 présente les résultats d'une simulation sur une journée, montrant que le TMPC parvient à suivre la référence de SoC de la batterie tout en respectant les contraintes.

Dans cette simulation, la puissance PV disponible $P_k^{\text{pv,max}}$ et la puissance des charges P_k^{load} ont été extraites directement des données historiques. Ces valeurs seront remplacées par les sorties du sous-système de **Forecasting** une fois celui-ci entièrement implémenté. Le problème d'optimisation du TMPC a été résolu toutes les 5 minutes, sur un horizon de prédiction de 6 heures. Un exemple de solution est illustré en Fig. 5, correspondant à l'optimum calculé à 08 :55. Le TMPC a trouvé les puissances optimales de la batterie permettant de suivre la référence de SoC, tout en respectant la contrainte d'absence d'injection au réseau et en utilisant la puissance PV disponible. On remarque quelques cas d'injection de puissance dans le réseau en Fig. 4, probablement dus à des imprécisions du modèle BMG par rapport au système réel.

Le système de **Forecasting** de l'irradiation solaire est en cours de développement afin de fournir des prévisions fiables sur un horizon de 6 heures. Il utilise comme entrées l'irradiation actuelle, l'heure de la journée et l'indice du jour, permettant au modèle de capturer les tendances journalières et saisonnières sans recourir à des données météorologiques supplémentaires. Un réseau de neurones artificiels (ANN) a été entraîné à partir des données historiques du BMG, avec un ajustement des hyperparamètres visant à équilibrer efficacité computationnelle et précision des prévisions. Le réseau est structuré en plusieurs couches avec des fonctions d'activation ReLU (Rectified Linear Unit), assurant des sorties non négatives cohérentes avec

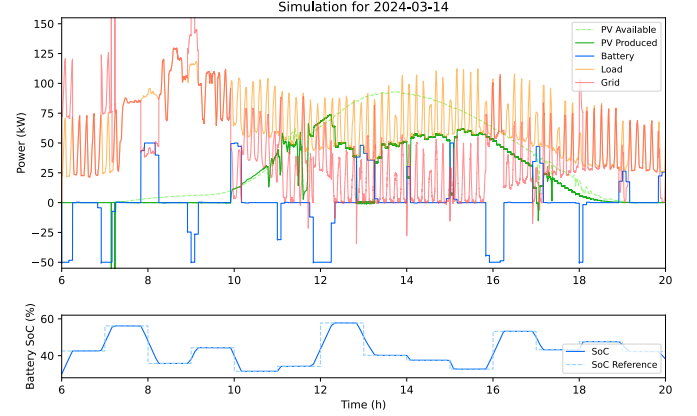


FIG. 4. Simulation du BMG à partir de données historiques de production PV et de consommation de charge, avec optimisation via le TMPC. La référence de SoC de la batterie est modifiée aléatoirement chaque heure.

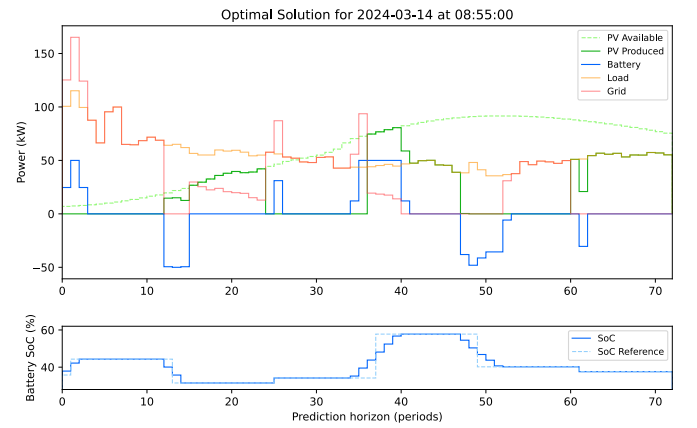


FIG. 5. Solution optimale du TMPC à 08 :55 avec $T_{\text{TMPC}} = 600$ s, $N_{\text{TMPC}} = 72$.

la nature physique de l'irradiation solaire. Une régularisation par dropout a été introduite entre les couches pour limiter le sur-apprentissage, et les variables d'entrée ont été normalisées par mise à l'échelle min-max. Un modèle de prévision des charges sera développé selon une approche similaire.

2.4. Tests et Validation

Des tests sont réalisés en continu tout au long du processus de développement, selon une approche itérative. Cela permet d'identifier et de corriger les problèmes rapidement, garantissant le bon fonctionnement du système à mesure que de nouvelles fonctionnalités sont ajoutées. Dans un premier temps, chaque sous-système a été testé individuellement, avec un accent mis sur les tests unitaires afin de valider leurs fonctionnalités. Cela inclut les tests des différentes fonctions des modèles PV et batterie déployés sous forme de FMU, comme le chargement des FMU, l'initialisation, la configuration des paramètres, l'exécution pas à pas, la récupération des sorties et leur comparaison aux résultats attendus. Des tests de validation ont également été réalisés à l'aide de données réelles du BMG, afin de s'assurer que les modèles respectent les exigences de précision définies. Par ailleurs, des tests unitaires ont été implémentés pour chaque sous-système du BMGDT, garantissant le bon fonctionnement de chaque composant de manière isolée. Pour la mise en œuvre de ces tests, la bibliothèque `pytest` de Python a été utilisée, permettant de définir des cas de test, de les exécuter et de vérifier les résultats. Actuellement, des tests d'intégration sont en cours de développement pour valider les interactions entre les sous-systèmes du BMGDT. Cela inclut des tests de flux de données de bout en bout, la synchronisation **Simulator-BMG Connector-Orchestrator**, la simulation hors ligne utilisant le **Simula-**

tor et l'**EMS** à partir des prévisions du module de **Forecasting**, ainsi que la boucle de rétroaction en temps réel impliquant le **BMG**, le **BMG Connector**, l'**EMS** et la **Forecasting**. En complément, une fonctionnalité de journalisation a été mise en place dans chaque sous-système à l'aide de la bibliothèque logging de Python, afin de suivre le déroulement de l'exécution et de capturer les erreurs.

3. CONCLUSIONS

Ce travail a présenté la méthodologie de conception et de développement d'un DT pour un démonstrateur de BMG. Les exigences, l'architecture et le processus de développement ont été détaillés, ainsi que l'état d'avancement actuel.

Le développement du BMGDT est en cours, avec une première version prévue pour la fin de l'année 2025. La majorité des fonctionnalités principales ont été implémentées, incluant les modèles du BMG, le **Simulator**, le **BMG Connector**, le **Data Processor** et l'**Orchestrator**. Les sous-systèmes **EMS** et **Forecasting** sont en cours de développement, le TMPC ayant déjà été implémenté et testé à partir de données historiques. Les sous-systèmes **FDD** et **AMA** sont également en développement, avec des premiers tests prévus dans les mois à venir. Une fois la première version déployée, elle sera validée dans des conditions réelles afin d'évaluer son impact sur la performance du BMG et les objectifs d'autoconsommation.

Plusieurs défis ont été rencontrés au cours du développement. Par exemple, la validation et le débogage de modèles BMG très détaillés se sont révélés difficiles aux premières étapes. Ces modèles, complexes et coûteux en calcul, rendaient les tests peu efficaces. Pour y remédier, une approche de modélisation progressive a été adoptée, en commençant par des modèles simplifiés, puis en ajoutant progressivement de la complexité. Cela a permis de valider les modèles de manière incrémentale, en assurant une précision suffisante sans introduire de complexité superflue. Un autre défi majeur a été la synchronisation des différents sous-systèmes, en particulier lors des simulations en temps réel. Cela a nécessité une conception soignée de l'**Orchestrator** pour garantir la synchronisation correcte des composants et la bonne gestion des flux de données. Il a aussi fallu renforcer la gestion des erreurs et les mécanismes de reprise dans le **BMG Connector**, afin d'assurer une communication robuste avec l'API du BMG.

Au-delà des fonctionnalités actuelles, d'autres évolutions sont prévues, telles que l'intégration de modèles thermiques pour le BMG, la simulation de scénarios « what-if », l'intégration d'API de prévisions météorologiques, ou encore la maintenance prédictive-prescriptive. L'objectif initial de ce projet était de faciliter le développement et les tests d'algorithmes avancés de gestion énergétique et de détection de défauts pour les BMG. Nous envisageons désormais le BMGDT comme une plateforme de recherche pour les technologies avancées de BMG, susceptible de contribuer au développement de systèmes énergétiques plus performants et durables.

ACKNOWLEDGEMENT

Les auteurs remercient l'Institut Carnot ARTS, l'ESTIA, le L2EP et Centrale Lille pour leurs aides financières sur ce projet *Optimisation d'un Micro-Réseau Intégré aux Bâtiments pour l'Autoconsommation d'Energie Renouvelable*.

4. RÉFÉRENCES

- [1] IEA. « Perspectives for the Clean Energy Transition. The Critical Role of Buildings ». In : *International Energy Agency* (2019), p. 1-114.
- [2] D. Y. YAMASHITA et al. « Hierarchical Model Predictive Control to Coordinate a Vehicle-to-Grid System Coupled to Building Microgrids ». In : *IEEE Transactions on Industry Applications* 59.1 (2023), p. 169-179. DOI : 10.1109/TIA.2022.3215978.
- [3] X. YAN, D. ABBES et B. FRANCOIS. « Uncertainty analysis for day ahead power reserve quantification in an urban microgrid including PV generators ». In : *Renewable Energy* 106 (2017), p. 288-297. ISSN : 0960-1481. DOI : <https://doi.org/10.1016/j.renene.2017.01.022>.

- [4] X. WEN, D. ABBES et B. FRANCOIS. « Modeling of photovoltaic power uncertainties for impact analysis on generation scheduling and cost of an urban micro grid ». In : *Mathematics and Computers in Simulation* 183 (2021), Special Issue: ELECTRI-MACS 2019 ENERGY - Modelling and computational simulation for control and diagnosis in renewable energy systems, energy storage, innovative devices and materials, p. 116-128. ISSN : 0378-4754. DOI : <https://doi.org/10.1016/j.matcom.2020.02.023>.
- [5] C. NWOGU et al. « Towards a Requirement-driven Digital Twin Architecture ». In : *Procedia CIRP* 107 (2022). Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022, p. 758-763. ISSN : 2212-8271. DOI : <https://doi.org/10.1016/j.procir.2022.05.058>.
- [6] V. DAMJANOVIC-BEHRENDT et W. BEHRENDT. « An open source approach to the design and implementation of Digital Twins for Smart Manufacturing ». In : *International Journal of Computer Integrated Manufacturing* 32.4-5 (2019), p. 366-384. DOI : 10.1080/0951192X.2019.1599436.
- [7] F. TAO et al. « makeTwin: A reference architecture for digital twin software platform ». In : *Chinese Journal of Aeronautics* 37.1 (2024), p. 1-18. ISSN : 1000-9361. DOI : <https://doi.org/10.1016/j.cja.2023.05.002>.
- [8] *Specification of the Asset Administration Shell. Part 1: Metamodel*. Rapp. tech. IDTA Number: 01001-3-0-2. Accessed: 2025-05-15. Industrial Digital Twin Association (IDTA), 2025. URL : <https://industrialdigitaltwin.org/en/content-hub/aasspecifications/specification-of-the-asset-administration-shell-part-1-metamodel-idta-number-01001-3-0-2>.
- [9] A. BUDIARDJO et D. MIGLIORI. *Digital Twin System Interoperability Framework*. Rapp. tech. Accessed: 2025-05-15. Digital Twin Consortium, 2021. URL : <https://www.digitaltwinconsortium.org/wp-content/uploads/sites/3/2022/06/Digital-Twin-System-Interoperability-Framework-12072021.pdf>.
- [10] *Details of the Asset Administration Shell. From idea to implementation*. Rapp. tech. Accessed: 2025-05-15. Plattform Industrie 4.0, 2019. URL : <https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/vws-in-detail-presentation.html>.
- [11] D. MCKEE. *Platform Stack Architectural Framework: An Introductory Guide*. Rapp. tech. Accessed: 2025-05-15. Digital Twin Consortium, 2023. URL : <https://www.digitaltwinconsortium.org/wp-content/uploads/sites/3/2023/07/Platform-Stack-Architectural-Framework.pdf>.
- [12] Y. LIU, S. PAN et E. BALLOT. « Unveiling the potential of digital twins in logistics and supply chain management: Services, capabilities, and research opportunities ». In : *Digital Engineering* 3 (2024), p. 100025. ISSN : 2950-550X. DOI : <https://doi.org/10.1016/j.dte.2024.100025>.
- [13] A. CIMINO et al. « Enhancing internal supply chain management in manufacturing through a simulation-based digital twin platform ». In : *Computers & Industrial Engineering* 198 (2024), p. 110670. ISSN : 0360-8352. DOI : <https://doi.org/10.1016/j.cie.2024.110670>.
- [14] M. A. HASAN et al. « Smart health practices: Strategies to improve healthcare efficiency through digital twin technology ». In : *Smart Health* 36 (2025), p. 100541. ISSN : 2352-6483. DOI : <https://doi.org/10.1016/j.smhl.2025.100541>.
- [15] A. ABDESSADAK et al. « Digital twin technology and artificial intelligence in energy transition: A comprehensive systematic review of applications ». In : *Energy Reports* 13 (2025), p. 5196-5218. ISSN : 2352-4847. DOI : <https://doi.org/10.1016/j.egyr.2025.04.060>.
- [16] N. P. KURUVATTI et al. « Empowering 6G Communication Systems With Digital Twin Technology: A Comprehensive Survey ». In : *IEEE Access* 10 (2022), p. 112158-112186. DOI : 10.1109/ACCESS.2022.3215493.
- [17] M. ALVI et al. « Global perspectives on digital twin smart cities: Innovations, challenges, and pathways to a sustainable urban future ». In : *Sustainable Cities and Society* 126 (2025), p. 106356. ISSN : 2210-6707. DOI : <https://doi.org/10.1016/j.scs.2025.106356>.
- [18] V. M. G. MARTINEZ, D. R. CAMPELO et M. R. N. RIBEIRO. « Sustainable Intelligent Transportation Systems via Digital Twins: A Contextualized Survey ». In : *IEEE Open Journal of Intelligent Transportation Systems* 6 (2025), p. 363-392. DOI : 10.1109/OJITS.2025.3553696.
- [19] N. BAZMOHAMMADI et al. « Microgrid Digital Twins: Concepts, Applications, and Future Trends ». In : *IEEE Access* 10 (2022), p. 2284-2302. DOI : 10.1109/ACCESS.2021.3138990.
- [20] X. YAN, D. ABBES et B. FRANCOIS. « Development of a tool for urban microgrid optimal energy planning and management ». In : *Simulation Modelling Practice and Theory* 89 (2018), p. 64-81. ISSN : 1569-190X. DOI : <https://doi.org/10.1016/j.simpat.2018.09.006>.
- [21] T. WAGNER et al. « Digital Twins in Power Systems: A Proposal for a Definition ». In : *IEEE Power and Energy Magazine* 22.1 (2024), p. 16-23. DOI : 10.1109/MPE.2023.3328581.
- [22] H. JIANG et al. « Digital Twin of Microgrid for Predictive Power Control to Buildings ». In : *Sustainability* 16.2 (2024). ISSN : 2071-1050. DOI : 10.3390/su16020482.
- [23] M. DELLALY, S. SKANDER-MUSTAPHA et I. SLAMA-BELKHODJA. « A digital twin model-based approach to cost optimization of residential community microgrids ». In : *Global Energy Interconnection* 7.1 (2024), p. 82-93. ISSN : 2096-5117. DOI : <https://doi.org/10.1016/j.gloe.2024.01.008>.
- [24] M. M. H. SIFAT et al. « Novel abstractions and experimental validation for digital twin microgrid design: Lab scale studies and large scale proposals ». In : *Applied Energy* 377 (2025), p. 124621. ISSN : 0306-2619. DOI : <https://doi.org/10.1016/j.apenergy.2024.124621>.
- [25] Y. WU et al. « Digital Twins for Microgrids: Opening a New Dimension in the Power System ». In : *IEEE Power and Energy Magazine* 22.1 (2024), p. 35-42. DOI : 10.1109/MPE.2023.3324296.
- [26] L. HADJIDEMETRIOU et al. « A digital twin architecture for real-time and offline high granularity analysis in smart buildings ». In : *Sustainable Cities and Society* 98 (2023), p. 104795. ISSN : 2210-6707. DOI : <https://doi.org/10.1016/j.scs.2023.104795>.
- [27] MAREE, JOHANNES P. et BAGLE, MARIUS. « A Building Automation and Control micro-service architecture using Physics Inspired Neural Networks ». In : *E3S Web Conf.* 362 (2022), p. 13001. DOI : 10.1051/e3sconf/202236213001.
- [28] D. YASSUDA YAMASHITA, I. VECCHIU et J.-P. GAUBERT. « Two-level hierarchical model predictive control with an optimised cost function for energy management in building microgrids ». In : *Applied Energy* 285 (2021), p. 116420. ISSN : 0306-2619. DOI : <https://doi.org/10.1016/j.apenergy.2020.116420>.