

Title	相互作用のあるシミュレーションを統合する都市デジタルツイン基盤の研究
Author(s)	大懸, 崇一郎
Citation	
Issue Date	2025-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/19787
Rights	
Description	Supervisor: 丹 康雄, 先端科学技術研究科, 修士 (情報科学)

修士論文

相互作用のあるシミュレーションを統合する都市デジタルツイン基盤の研究

大懸 崇一郎

主指導教員 丹 康雄

北陸先端科学技術大学院大学
先端科学技術研究科
(情報科学)

令和7年3月

Abstract

In this study, we conducted research on a digital twin integration platform for interactive simulations in urban environments. As digital twins become more widely used for urban planning, energy management, mobility analysis, and smart city applications, it is essential to develop an integration framework that enables seamless interaction and coordination between multiple digital twins. This research focuses on the interactivity of digital twins, proposing a comprehensive approach to managing relationships and time synchronization within an integrated simulation environment.

We propose a digital twin integration platform that facilitates interaction among multiple digital twins by implementing a structured relationship management and time control mechanism. Relationship management is achieved through the construction of an attributed directed graph model, which captures dependencies and interactions between different digital twins. By analyzing these interaction networks, we can evaluate the stability and coherence of multi-digital twin simulations. Additionally, the time management component introduces a scheduling methodology that incorporates time-related elements within the integration platform. This method ensures proper synchronization and coordination between digital twins, thereby improving the efficiency and accuracy of interactive simulations.

One of the key challenges in interactive digital twin simulations is the potential divergence of interactions, where instability can arise due to feedback loops and unregulated dependencies. To address this, we analyze the stability of digital twin interactions and propose a time control mechanism that mitigates divergence effects. This approach enables controlled execution of digital twins while maintaining a balance between real-time simulation and predictive modeling.

To validate our approach, we implemented the proposed digital twin integration platform and conducted a series of experiments involving multiple digital twins. These experiments included execution management based on predefined data models, real-time interaction monitoring, and performance evaluations of the scheduling mechanism. The results demonstrate that our framework effectively integrates digital twins, ensuring stable interactions while enhancing computational efficiency.

Through this study, we propose a robust digital twin integration platform that supports seamless collaboration among interactive digital twins. By introducing an interaction-aware scheduling mechanism, our platform enhances predictive accuracy and computational performance, enabling more reliable urban simulations.

The proposed integration framework contributes to the advancement of smart city applications, providing a foundation for scalable and adaptable digital twin ecosystems that can support real-time decision-making and scenario analysis.

概要

本研究では、都市環境における相互作用するシミュレーションの統合を目的としたデジタルツイン基盤の構築を行う。都市デジタルツインは、都市計画、エネルギー管理、モビリティ解析、スマートシティアプリケーションなど多岐にわたる分野で活用されている。しかし、現在のデジタルツインはそれぞれ独立して運用されており、相互のデータ共有が不十分であり、統合フレームワークの欠如が課題となっている。本研究では、都市デジタルツイン間の相互作用を管理するために、関係性管理と時間制御の機能を備えた統合基盤を提案する。加えて、都市における異なるシミュレーションのデータを統合することで、より精度の高い都市解析が可能となることを示す。

提案するデジタルツイン統合基盤は、有向グラフを用いた関係性管理と、時間同期を考慮したスケジューリング機構を組み込むことで、複数のデジタルツイン間の相互作用を適切に制御する。関係性管理では、各デジタルツイン間の依存関係をグラフモデルとして表現し、相互作用の安定性を解析することで、適切な連携を実現する。また、時間制御では、シミュレーション間の時間軸のズレを調整するスケジューリング手法を導入し、リアルタイム性と予測精度を向上させる。さらに、デジタルツイン同士のデータ交換の最適化を図り、よりスムーズな相互作用を可能にするためのアルゴリズムを実装する。これにより、都市環境の変化に対応したシミュレーションの精度向上を目指す。さらに、提案基盤の有効性を検証するために、複数のデジタルツインを用いた実験を実施した。

実験では、シナリオデータに基づく実行管理、リアルタイムな相互作用のモニタリング、スケジューリング機構の性能評価を行い、その結果、提案基盤がデジタルツインの統合において安定した相互作用を確保しつつ、計算効率を向上させることを確認した。また、データ統合の影響を詳細に分析し、都市デジタルツインの実用性を高めるための最適なパラメータを検討した。結果として、デジタルツイン間のデータ共有が促進され、より正確な予測が可能となった。本研究を通じて、都市デジタルツインの統合基盤として、相互作用を考慮したスケジューリング機構を導入することで、より精度の高い都市シミュレーションの実現が可能であることを示した。

本研究の成果は、スマートシティの発展に寄与し、リアルタイム意思決定やシナリオ分析を支援する汎用的なデジタルツイン基盤の構築に貢献するものである。さらに、デジタルツイン間の相互データ交換を最適化することで、都市計画やエネルギー管理、交通制御などの分野における応用の可能性を広げる。今後、本基盤をさらに発展させ、異なる都市環境に適用可能なフレームワークとしての拡張を図る。本研究では、都市デジタルツインの相互作用を考慮し、関係性管理と時間制御を組み込んだ統合基盤を提案した。その結果、安定した相互作用と計算効

率の向上を確認し、スマートシティ分野への貢献が期待されることを示した。また、提案した統合基盤が都市のデジタルツインの活用を促進し、より高度な都市シミュレーションの実現に貢献することが明らかとなった。

目次

第1章	はじめに	1
1.1	背景	1
1.2	目的	1
1.3	本論文の構成	2
第2章	デジタルツイン・都市デジタルツイン	3
2.1	デジタルツインの構成	5
2.2	都市デジタルツイン	6
2.3	デジタルツイン関連技術	8
2.3.1	IoT	8
2.3.2	センサー, アクチュエータ	8
2.3.3	データ	9
2.3.4	シミュレーション	10
2.3.5	予測モデル表現	10
2.4	連携技術	12
第3章	都市デジタルツイン連携基盤の課題	13
3.1	独立したデジタルツインの連携接続問題	14
3.2	複雑化したデジタルツイン管理問題	15
3.2.1	複雑化したデジタルツイン間の関係管理問題	16
3.2.2	複雑化したデジタルツイン間の時間管理問題	17
第4章	デジタルツイン連携基盤の提案手法	18
4.1	関係グラフモデルによる相互作用性の解析	19
4.2	相互作用の実行管理	22
4.2.1	デジタルツインの連携	24
4.2.2	相互作用するデジタルツインの連携	26
4.3	時間軸管理	28
4.3.1	スケジューリング管理	28

4.4	ユースケースに応じたデジタルツイン連携	30
4.5	デジタルツイン連携基盤の構成要素	30
4.5.1	データモデル	30
4.5.2	データ管理モジュール	33
4.5.3	時間管理モジュール	34
4.5.4	接続関係管理モジュール	34
4.5.5	シナリオ管理モジュール	34
4.5.6	実行管理モジュール	35
第5章	実験	38
5.1	実装	38
5.1.1	システム構成と目的	39
5.1.2	処理フローの設計	39
5.1.3	モジュール間の連携	39
5.1.4	システム内部の設計	40
5.1.5	実験内容	40
5.2	都市デジタルツインのユースケース	41
5.3	関係性管理の実験	42
5.3.1	シナリオによるシミュレーション条件の設定	43
5.3.2	関係グラフモデルの作成処理	43
5.3.3	シナリオによる実行可能性の判定	44
5.3.4	データパステーブルの作成処理	45
5.4	時間管理の実験	47
5.4.1	相互作用関係における時間管理実験	48
5.4.2	安定性を考慮した時間管理実験	50
第6章	評価	56
6.1	実験に対する評価	56
6.1.1	ユースケースに関して	56
6.1.2	時間制御に関して	56
6.2	結論	56
6.3	課題に対する評価	57
6.4	考察	58
6.4.1	結論	60
6.5	今後の展望	60

第7章 おわりに	62
.1 時間操作ベクトルと予測ベクトルにおける時間空間の証明	66
.1.1 ベクトル空間の定義	66
.1.2 ベクトル空間生成の証明	66
.1.3 結論	67
本研究に関する对外発表	69

目 次

2.1	デジタルツイン構成	5
2.2	デジタルツイン構成種類	6
3.1	デジタルツイン連携	14
4.1	デジタルツイン連携基盤	19
4.2	デジタルツイン連携基盤の処理フロー	20
4.3	経路探索フローチャート	21
4.4	相互作用 1	26
4.5	時間制御処理	36
4.6	実時間とシナリオ時間の関係	36
4.7	シナリオ時間とシミュレーション時間の関係	36
4.8	メタデータモデル	37
4.9	入出力データモデル	37
4.10	シナリオデータモデル	37
5.1	スースケース図	38
5.2	システム構成図	39
5.3	デジタルツイン登録の処理フロー	40
5.4	モジュール間のシーケンス図	41
5.5	デジタルツイン連携基盤の構成	42
5.6	API サーバーの実行結果	42
5.7	管理サーバーの実行結果	43
5.8	WebUI の接続に関する実行結果	43
5.9	デジタルツインテストデータの送信結果	44
5.10	都市デジタルツインユースケース	45
5.11	関係グラフモデルの生成結果	47
5.12	相関関係のある予測シミュレーション結果	52
5.13	相関関係のある予測シミュレーション数値変化の結果	54
5.14	相関関係のある予測シミュレーション予測時間の結果	55

表 目 次

4.1	予測モデルと関連要素の説明	24
4.2	時間管理に必要な要素	28
4.3	デジタルツイン連携基盤のモジュール構成	30
5.1	シナリオデータの実行可能ルート検索	44
5.3	ノードの構成	48
5.4	ノードの依存関係	49
5.5	ノード情報	52
5.6	エッジ情報（ノード間の関係）	53
5.7	ノードごとの予測時間上限	53
6.1	デジタルツイン連携基盤におけるデータモデルの比較	59

第1章 はじめに

1.1 背景

現在の日本は少子高齢化や過疎化、自然災害などさまざまな課題を抱えている。このような課題の解決手法として、近年では都市のデジタルツインが注目されている。デジタルツインとは、物理空間の情報をもとに仮想空間を再現し、現実世界から取得したセンサ情報を用いることで高精度なシミュレーションを行う技術である。デジタルツインを活用することで、現実に近いシミュレーションを実現することが可能となる。都市のデジタルツインを利用することで、人口動態の予測、土地利用の最適化、災害予測、社会インフラの維持に役立てることができる[1]。これは自治体の効率的かつ安定した運営につながる。一方で、現状では目的ごとに独立したデジタルツインが運用されており、それらが相互に連携していないのが通常である。都市のさまざまな側面を総合的にとらえる仕組みは依然として少ない。

1.2 目的

本研究では、都市デジタルツインにおける相互作用するシミュレータを連携させるための機能について検討する。シミュレーションが相互作用する状況では、互いにデータをやり取りすることで外部条件が変化し、それぞれのシミュレーションの予測性能の向上が期待される。しかし、互いのデータやり取りにおいて、適切な相互作用関係でない場合はシミュレーション結果が発散する可能性も考えられる。本研究では、独立していたデジタルツインを連携させ、動的なデータを利用した予測を可能にすることで、より現実に近いシミュレーションを実現する。

1.3 本論文の構成

第1章

研究背景と研究目的について述べる

第2章

デジタルツインについて調査した現状と連携技術について述べる

第3章

相互作用するデジタルツイン連携の課題について述べる

第4章

本研究で提案する連携基盤について述べる

第5章

本研究の実験結果について述べる

第6章

本研究の評価について述べる

第7章

本論文のまとめと今後の課題について述べる

第2章 デジタルツイン・都市デジタルツイン

デジタルツインは、物理空間と仮想空間の融合によって情報技術を活用し、製造システムの柔軟性や効率を向上させる技術である。デジタルツインの目的は、評価、最適化、予測の能力を備えた物理エンティティの状態や動作を忠実にエミュレートする仮想モデルを作成することである [2]。デジタルツインの概念は、NASAのアポロ計画において、ミッション中の宇宙船の状態をミラーリングするために2つの同一の宇宙船が作られたことに起因している [3]。

産業界におけるデジタルツインは、設計段階、製造段階、サービス段階に分けて考えることができる [4]。- 設計段階では、シナリオや構想の検証において設計者を支援する役割を果たす。- 製造段階では、生産要素間の相互作用を分析し、意思決定者の支援に貢献する。- サービス段階では、リアルタイムの状態監視やメンテナンスなどを通じて製品寿命を予測することに役立つ。

しかし、デジタルツインを実現するためには、物理システムを忠実に模倣するための高度なデータとモデルが必要であり、その構築には多大な労力が伴う [5]。

近年、インダストリー 4.0 の枠組みにおいて、デジタルツインの活用が重要視されている。この概念では、分散化・相互運用性・仮想化・リアルタイム処理・サービス指向・モジュール化といった設計原則が求められる [6]。特に、サイバーフィジカルシステム (CPS) は、物理システムのデジタル複製を作成し、実際のシステムとの相互連携を維持しながらデータ収集やシミュレーションを行う役割を果たす [7]。このデジタルコピーは、現実世界のプロセスを監視・解析するための基盤となり、デジタルツイン技術の発展につながった [8]。

デジタルツインは「物理的な製品のデジタル製品対応物」として定義されることが多い [9]。その概念アーキテクチャは、create (作成)、communication (通信)、aggregation (集約)、analyze (分析)、insight (洞察)、act (行動) の6つのステップで構成される [10]。このような定義をもとに、デジタルツインは物理的なオブジェクトやシステムのデジタルコピーであり、接続された機能データや運用データを共有する一般的な概念となっている。デジタルコピーを活用することで、リアルタイムデータや履歴データを基に、物理資源の状態を評価し、機械学習やシ

ミュレーション技術を用いて最適化や予測を行い、物理資産のパフォーマンスを向上させることが可能である。デジタルツインはさまざまな分野で活用されており、特に代表的な分野として、製造業とヘルスケアについて述べる。

製造業における応用

デジタルツインは、製品のライフサイクル全体（設計・生産・物流・保守）にわたって利用されており、各工程での最適化や異常検知を可能にする [11]。

ヘルスケアにおける応用

ヘルスケア領域において、デジタルツインは 医療機器 および 患者管理 の2つの側面で活用が進んでいる。

ヘルスケアにおけるデジタルツイン

メンテナンスの必要性を事前に特定し、リモート監視を通じて安全な運用を支援する。また、プロトタイプ開発前のシミュレーションによって設計の精度を向上させる。

患者のデジタルツイン

個別化医療の実現に向けたアプローチとして注目されており、患者データを基に病状のシミュレーションを行うことで、診断精度の向上や治療計画の最適化を支援する [12]。

2.1 デジタルツインの構成

本研究におけるデジタルツインの構成は、シミュレーション、センシング、およびそれらの情報を蓄積するためのデータベースで構成される(図2.1)。また、インターフェースの種類によって、図2.2の3種類に分類される。一つ目は、デジタルツインを構成する要素であるシミュレーション、データベース、デバイスの各要素にそれぞれインターフェースが設けられているタイプである。二つ目は、シミュレーションとデータベースの間にインターフェースがなく、デバイスとデータベースの間にインターフェースが設けられているタイプである。三つ目は、シミュレーション、データベース、デバイスのすべてが連結されているタイプである。これらの3種類の構成すべてにおいて、デジタルツインの起動・実行などを制御するためのインターフェースが設けられている。

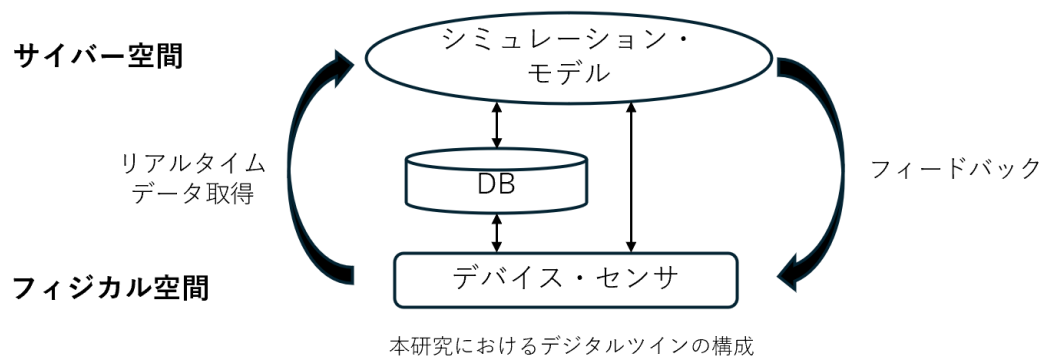


図 2.1: デジタルツイン構成

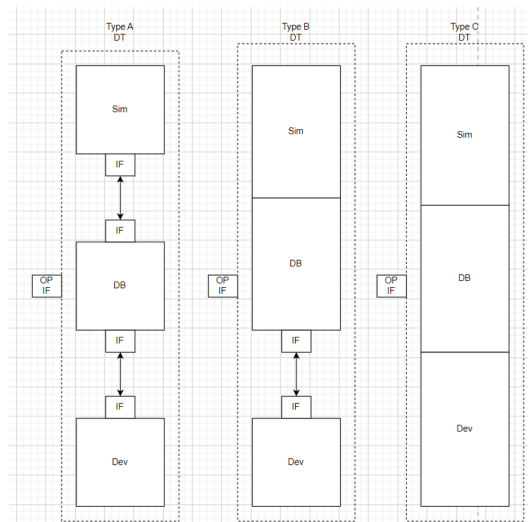


図 2.2: デジタルツイン構成種類

2.2 都市デジタルツイン

スマートシティにおけるデジタルツイン（DT）は、都市全体の仮想モデルを構築する技術であり、特に都市の電力ネットワークやインフラのデジタル表現を含む。この技術を活用することで、信号機やウォーターポンプなど、都市内に分散するさまざまな設備を遠隔監視できるようになる。都市型 DT は、予防的な戦略立案や新規インフラの影響評価に活用される [13]。

都市におけるデジタルツインの適用対象は、人口密度が高く、整備されたインフラと密集する建造物を持つ居住地域が中心となる。DT の活用により、都市環境の管理や住民の生活の質の向上が期待される。その具体的な応用分野として、スマートビルディング、交通管理、大気質モニタリング、インフラ管理、循環型都市経済のシミュレーションが挙げられる。これにより、人流や緊急避難時のシミュレーションが可能となる。また、DT の導入による利点として、予防保守の強化、運用の効率化、コスト削減、市民サービスの向上、安全対策の最適化などが挙げられる。さらに、リアルタイムの都市環境モニタリングや災害時の緊急避難ルート計画にも貢献する。

世界および日本国内の代表的なデジタルツイン事例として、Virtual Singapore、PLATEAU、豊洲プロジェクト、および東京都デジタルツイン実現プロジェクトについて紹介する [14]。

Virtual Singapore

Virtual Singapore は、シンガポール政府が推進するスマートシティプロジェクトであり、都市全域を 3D モデル化したデジタルツインを構築している。このプロジェクトは、都市環境の可視化を通じて都市計画、エネルギー管理、防災対策などの意思決定を支援することを目的としている [15]。

PLATEAU

PLATEAU は、日本の国土交通省が主導する全国規模の 3D 都市モデルデータセットの開発プロジェクトである。この取り組みは、現実の都市空間をサイバー空間に再現し、都市の現状を可視化することで、統合的な都市計画や問題の特定・解決を促進することを目指している [16]。

豊洲プロジェクトを対象とした都市型デジタルツイン

清水建設とオートデスクが協力し、豊洲地区を対象とした都市型デジタルツインの構築を進めている。このプロジェクトでは、BIM（Building Information Modeling）と ICT 技術を活用し、建物・インフラの BIM・CIM（Construction Information Modeling）データ、広域地形モデル、シミュレーションデータを統合したプラットフォームを開発している。これにより、人流、物流、交通、防災機能の最適化を目指す [17]。

東京都デジタルツイン実現プロジェクト

東京都では、デジタルツインを支える 3D デジタル地図を作成し、シミュレーションを通じて都市運営の改善を図るプロジェクトを進行中である。このプロジェクトでは、具体的な活用事例を探りながら、複雑な社会課題の解決や都市の QOL（生活の質）の向上に向けた施策を検討している [18]。

2.3 デジタルツイン関連技術

2.3.1 IoT

IoT (Internet of Things) は、さまざまな「物理的な対象」がネットワークを通じて接続され、情報をやり取りしたり、デバイスの制御を行ったりする仕組みを指す。ここでいう「物理的な対象」には、センサーやアクチュエータだけでなく、家電製品、住宅設備、自動車などが含まれ、近年その数は急速に増加している。

通信技術の発展に伴い、IoTはすでに多くの分野で活用されており、通信分野自体の成長は比較的安定している。しかしながら、今後特に発展が見込まれる領域として、ヘルスケア、産業分野、スマートホームや家電といったコンシューマ市場が挙げられる。

都市デジタルツインにおいて、環境データのセンシングやリアルタイムな情報収集を実現するためには、IoT技術の活用が不可欠である。これにより、都市の各種インフラの状態を正確に把握し、より効率的な都市運営を支援することが可能となる。

2.3.2 センサー, アクチュエータ

センサの概要

センサとは、機械、熱、化学、電磁気、光、生物といった多様な物理的・化学的特性を検出し、それらを機械が処理可能な信号へと変換する装置である。現実空間のデータを仮想環境に適用するデジタルツインを実現するためには、センサが取得した情報を適切に処理・送信し、デジタルモデルと同期させることが不可欠である。これにより、リアルタイムの監視、最適化、シミュレーションが可能となる。

デジタルツインやIoTの導入には、センサネットワークの構築が不可欠である。センサネットワークとは、複数のセンサを適切に配置し、それらを連携させながらデータの取得・解析・制御を行う仕組みのことである。このネットワークは、環境モニタリング、スマートホーム、自動運転、産業用設備管理など、多様な分野で活用されている。

センサネットワークを構成する主要な要素として、センサノードが挙げられる。センサノードは、センサ本体、無線通信モジュール、マイクロプロセッサ、電源ユニットといったコンポーネントで構成され、物理的・化学的な情報を測定し、それを処理してデータとして送信する。センサネットワークにおいて、データの送受信には無線通信技術が不可欠であり、用途に応じて適した規格を選択すること

が求められる。代表的な通信規格には、Bluetooth、ZigBee、Wi-Fi、NFCがあり、それぞれ異なる特性を持つ。Bluetoothは短距離通信向けでウェアラブルデバイスやスマートホーム機器に適しており、ZigBeeは低消費電力でメッシュネットワーク構築が可能なためスマートメーターや産業用途に利用される。Wi-Fiは高速通信が可能であるが消費電力が高いため主に固定設置のデバイス向けであり、NFCは極短距離でのデータ交換が可能で電子決済やタグ認証に活用される。

センサは、デジタルツインの実現に不可欠なデータ収集装置であり、そのデータを適切に処理・伝送するためにはセンサネットワークの構築が重要となる。用途に応じたセンサノードの設計と通信技術の選択により、リアルタイムなデータ活用や最適なシステム運用が可能となる。

アクチュエータの概要

アクチュエータとは、入力されたエネルギーを運動エネルギーへと変換する装置を指す。アクチュエータを利用することで、直線運動、回転運動、振動運動などを再現することが可能である。また、動力源としては主に電力、油圧、空気圧の3種類が存在し、それぞれ特定の用途に適した特性を持つ。

センサとアクチュエータは、デジタルツインやIoTシステムにおいて相互に連携し、リアルタイムな情報の取得と物理的なフィードバックを可能にする重要な要素である。

2.3.3 データ

デジタルツインは、物理システムの仮想モデルを構築し、その挙動をリアルタイムに解析および最適化する技術である。本研究では、デジタルツインにおいて使用されるデータを分類し、それぞれの特徴および役割について整理する。これらのデータを適切に管理・統合することで、デジタルツインは実世界のシステムを忠実に模倣し、高度な最適化・異常検知・予測制御を実現できる。

データ種類	内容
静的データ	物理特性、設計情報、仕様、構成情報
動的データ	環境データ、エネルギーデータ、設備状態、交通データ、通信データ
履歴データ	エネルギー消費履歴、設備故障履歴、気象データ履歴、ユーザー行動履歴、交通データ履歴

シミュレーションデータ	エネルギー需要予測、設備劣化シミュレーション、 気象シナリオ、交通流シミュレーション
相互作用データ（シナリオデータ）	スマートホーム連携、エネルギーマネジメント、 モビリティ連携、環境変化対応
制御データ（アクションデータ）	スマート家電制御、ロボット・自動機械制御、電力フロー制御

2.3.4 シミュレーション

デジタルツインで使用されるシミュレーション・モデルの構築方法は、主に2つに分類される。1つ目は、物理的な事象をもとに数理モデルを使用して構築する方法である。2つ目は、データを基にして仮想モデルを構築する方法である。

数理モデルを基にしたモデルの構築方法の例としては、有限要素法などが挙げられる。数理モデルを使用したシミュレーション・モデルの構築では、物理現象を近似してモデル化するため、実世界の結果と比較した場合に誤差が生じるという特徴がある。

一方、データを基にした仮想モデルの構築方法の例としては、機械学習を活用した方法がある。データを基にしたシミュレーション・モデルの構築では、モデリングに使用するデータに大きく依存するため、解釈性能が低くなるという特徴がある。

2.3.5 予測モデル表現

シミュレーションについて、予測モデル、状態空間モデル、差分方程式、伝達関数、微分方程式、シミュレーションベースの手法を「線形モデル」と「非線形モデル」に分類し、それぞれの特徴と代表的な例を以下に示す。以下の表は、各カテゴリにおける線形および非線形モデルの代表例をまとめたものである。

カテゴリ	線形モデルの例	非線形モデルの例
予測モデル	線形回帰、ロジスティック回帰	ニューラルネットワーク、LSTM
状態空間モデル	線形制御、電気回路	非線形ロボット、気象モデル
差分方程式	デジタルフィルタ、ARMA	カオス系、複雑な動的システム
伝達関数	振動系、フィードバック制御	非線形近似（ボルテラ級数など）
微分方程式	RLC 回路、振動系モデル	化学反応モデル、流体力学

カテゴリ	線形モデルの例	非線形モデルの例
シミュレーション	線形制御の数値解析	天体運動、車両ダイナミクス

2.4 連携技術

シミュレーションの統合技術の一例として、High Level Architecture (HLA) が挙げられる。HLA は、米国防総省の主導で策定された分散シミュレーションの標準規格であり、ミドルウェアを介したデータ共有や時刻同期機能を提供する。この技術を活用することで、異なるシミュレーション環境間でのスムーズな情報交換が可能となり、大規模かつ複雑なシステムの評価や最適化を支援する [19]。

シミュレーションとエミュレーションの連携基盤として知られる Smithonian は、MQTT プロトコルを利用した Publish/Subscribe 方式のメッセージ交換により、リアルタイムな同期と連携を実現する。この仕組みは、IoT デバイスや分散型システムとの連携を容易にし、スケーラブルなシミュレーション環境の構築を可能にする [20]。

ARIA は Smithonian を活用し、水害発生時の被害状況をリアルタイムで反映しながら予測を行う。さらに、被害予測結果をもとに避難情報の発信タイミングを最適化し、災害被害の低減効果を定量的に評価する仕組みを導入している。加えて、ARIA は気象データや地理情報システム (GIS) と統合し、環境要因を考慮したシミュレーションの高度化を図っている [21]。

FIWARE は、欧州連合 (EU) が支援するオープンソースのスマートシティ向けプラットフォームであり、IoT、デジタルツイン、データ統合を支援する機能を備えている。FIWARE はコンテキストデータ管理を中心とし、さまざまなセンサやシステムからのデータを統合・解析することが可能である。HLA や Smithonian と組み合わせることで、より高度な都市シミュレーションやインフラ管理が実現できる [5]。

FIWARE の主要コンポーネントである Orion Context Broker は、データの取得、保存、共有を行うための中核技術であり、リアルタイムの情報処理を可能にする。また、QuantumLeap や Cygnus といったデータ分析ツールと連携することで、長期間のデータ履歴の蓄積や機械学習を活用した予測モデルの構築が可能となる [22]。

今後の発展として、HLA や Smithonian、FIWARE を組み合わせたシミュレーション技術の適用範囲を広げ、交通管理、エネルギー最適化、都市インフラ管理といった分野に応用することが期待される。これにより、デジタルツインの高度化が進み、より精緻な都市計画や防災戦略の策定が可能となる。

第3章 都市デジタルツイン連携基盤の課題

本研究では、都市デジタルツインを統合する基盤の提案を行う。都市のデジタルツインは、物理空間（現実世界）の建物や空間、人やインフラ、気候などのあらゆる情報を含む。都市におけるデジタルツインは、都市計画や交通、防災、スマートホームなど、さまざまな目的で利用されている。現在のデジタルツインは個々の目的のために作成されており、垂直的な利用が中心である。それぞれのデジタルツインは個々の目的を持っているが、対象としている都市は同一であり、空間や時間を共有している。物理空間で共有している情報がある場合、それを仮想空間上でも統合し、相互に情報を交換することで、デジタルツイン同士が相互作用を生み出すことが可能となる。

しかし、デジタルツインを統合する取り組みはまだ十分に行われていない。現在、センシングデータをさまざまな分野のシミュレーションで利用する事例はあるが、シミュレーションの出力結果を別のシミュレータに渡すことでシミュレーション同士が相互作用を生み出すようなデジタルツインの統合は実現されていない。

図3.1は、3つのデジタルツインが互いに依存データを交換しながら実行されている様子を示している。通常のデジタルツインは、センシングデータ、静的データ、動的データを入力としてシミュレーションを実行する。しかし、静的データが他のデジタルツインによって動的に変化する場合、出力結果も変化する。この出力変化は、さらに別のデジタルツインにも影響を与える。

相互作用するデジタルツインを取り扱うためには、以下の課題が考えられる。

- 独立したデジタルツインの連携接続の問題
- 複雑化したデジタルツイン管理問題

本章では、これらの課題について議論する。

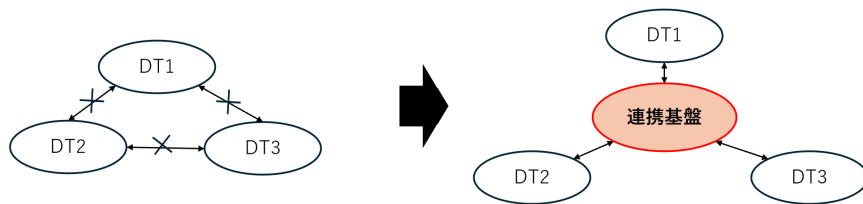


図 3.1: デジタルツイン連携

3.1 独立したデジタルツインの連携接続問題

デジタルツインの連携における重要性は、異なるシステム間でのデータの円滑な統合にある。この連携を実現するためには、各システムが使用するデータを共通の形式で表現し、効率的に結びつけるための情報と基盤が必要である。

まず、異なるシステム間でのデータ連携を行うには、データモデルを活用してデータを結合する仕組みを構築する必要がある。データモデルは、各システムで使用するデータの構造や意味を統一的に記述するための枠組みを提供する。このモデルを基に、連携基盤内で各システムのデータを一元的に管理し、相互運用を可能にする。

次に、データ変換を担うミドルウェアとしての機能が、連携基盤において必要不可欠である。このミドルウェアは、異なるフォーマットやプロトコルで構造化されたデータを変換し、データモデルに基づいて適切な形に整形する。これにより、システム間のデータ交換が効率的に行えるようになる。

さらに、データ連携を成功させるためには、連携基盤内でデータモデルの構造をあらかじめ定義する必要がある。これには、デジタルツインの連携に必要なパラメータを網羅的に記述することが含まれる。適切なデータモデルが確立されていない場合、データの一貫性や相互運用性が損なわれ、連携の実現が困難になる可能性がある。

デジタルツインの連携性は、各構成要素間のインターフェース設計にも大きく依存する。デバイス、データベース、シミュレーションといった構成要素間に適切なインターフェースが設定されている場合、柔軟なデータ連携が可能となる。一方で、インターフェースが不足している場合、データや機能の透明性や操作性が低下し、連携基盤の運用に制限が生じる恐れがある。

以上のように、デジタルツインの連携基盤では、各デジタルツインが使用するデータ、提供する機能、構造、属性、実行内容を包括的に管理する必要がある。この管理を実現するために、データモデルとデータ変換機能が連携基盤の中核的な役割を果たす。これらの要素が適切に設計され運用されることで、異なるシステ

ム間のデジタルツインの連携が円滑に進み、より効果的なシステム間の統合が実現する。

3.2 複雑化したデジタルツイン管理問題

デジタルツインを効果的に連携するためには、各デジタルツインがどのような関係を持ち、どのようにデータを共有し、相互に影響を及ぼすのかを明確に整理する必要がある。これを実現するためには、デジタルツイン間の関係を明示的に管理し、それを適切にモデル化する仕組みが求められる。しかし、現在のシステム連携に関する研究では、個々のデジタルツインの独立性を前提としたものが多く、デジタルツイン間の関係を体系的に管理する方法は十分に確立されていない。特に、デジタルツインが複数分野にまたがる場合、それぞれのデータの整合性や相互作用を適切に制御することが課題となる。

本研究では、デジタルツイン間の関係を整理し、連携基盤内での管理を強化するために、デジタルツイン間の関係管理問題とデジタルツイン間の時間管理問題の2つの主要な観点に着目する。

まず、デジタルツイン間の関係管理問題については、デジタルツイン間の依存関係の明確化、データの統合手法、連携基盤内での動的な接続管理の仕組みなどが重要な要素となる。デジタルツインは単独で機能するのではなく、多くの場合、他のデジタルツインと連携することで、その価値を最大化する。例えば、都市のデジタルツインにおいて、交通デジタルツインとエネルギーデジタルツインが相互にデータを共有することで、電動車両（EV）の充電需要と交通流を同時に最適化することが可能になる。このようなケースでは、単なるデータ交換だけでなく、データの意味的な整合性、更新頻度、データのなど時系列も管理する必要がある。しかし、現状ではデジタルツイン間の接続を静的な API やデータベース連携に頼ることが多く、動的に変化する環境への適応が難しいという課題がある。この問題を解決するためには、デジタルツインの関係性をネットワークとして管理し、リアルタイムで関係を更新できるグラフモデルなどの手法を導入することが考えられる。

次に、デジタルツイン間の時間管理問題については、異なる時間スケールで動作するデジタルツインの同期をどのように取るかが課題となる。例えば、リアルタイム制御を行うスマートグリッドのデジタルツインはミリ秒単位の更新が求められる一方、気象シミュレーションのデジタルツインは数時間から数日の時間スケールで予測を行う。このような異なる時間解像度のシステムを統合する場合、データの更新タイミングや時間的な整合性の確保が必要となる。また、デジタルツイ

ンが分散環境で動作する場合、ネットワーク遅延や処理時間のばらつきが生じるため、それに伴うデータの時間的ズレ（タイムラグ）をどのように補正するかも重要な課題となる。これらの問題に対応するためには、非同期処理の最適化、タイムスタンプ管理の強化、デジタルツイン間の時間補正アルゴリズムの導入などが求められる。

本研究では、以上の2つの課題についてより詳細に分析し、デジタルツインの関係管理と時間管理の両方の側面から、新たな管理手法を提案する。次のセクションでは、各課題をさらに掘り下げ、それぞれの問題点と解決の方向性について詳しく論じる。

3.2.1 複雑化したデジタルツイン間の関係管理問題

デジタルツイン間の相互作用をモデル化し、複数のデジタルツイン連携基盤においてどのように取り扱うのかについての説明する。連携基盤に接続されたデジタルツイン間において相互作用の関係がある場合、これらを連携基盤内でどのように取り扱うのかという問題がある。

デジタルツインをモデル化する目的は、複雑な物理システムやプロセスをデジタル上で再現し、その挙動や相互作用を効率的に解析することである。モデル化することにより、以下のようなメリットが得られる。

- シミュレーションを通じた予測精度の向上：物理的な制約を考慮しつつ、将来の挙動を高精度に予測可能である。
- 制御や最適化の効率化：システム間の相互作用を把握し、最適な制御方法を見出す手助けとなる。
- データ共有と透明性の向上：統一的なモデルにより、異なるシステム間でのデータ共有が円滑になる。
- システム全体の振る舞いの理解：モデル化された相互作用を通じて、システム全体の挙動を統一的に把握可能となる。

相互作用する関係をモデル化して扱うことで、連携基盤内で関係グラフの作成や、相互作用関係による特性を使用したスケジューリングや制御を実現することが可能である。本件におけるデジタルツイン連携基盤では、各デジタルツインシミュレーションについてモデル化して取り扱う。具体的には、デジタルツイン間の相互作用を行列表現することで、効率的な解析やシステム全体の挙動を予測する基盤を提供する。

閉ループとしての相互作用

デジタルツイン間の相互作用を閉ループとして扱う場合、各システムが他のシステムに影響を与え、その影響が再び自身に戻るようなサイクルが形成される。このような閉ループ構造を考慮することで、次のような利点が得られる。

- フィードバック効果の解析：システム間の影響がどのように循環し、全体に影響を与えるかを理解することが可能である。
- 安定性の評価：閉ループ構造における振る舞いを分析することで、システム全体の安定性を評価できる。
- 最適化の強化：フィードバックループを活用し、システムの性能を最大化する設計が可能となる。

例えば、閉ループ構造においては、フィードバックの遅延や増幅効果がシステムのダイナミクスにどのように影響するかを解析することで、設計や運用における重要な知見を得ることができる。これにより、システム全体の挙動をより詳細に把握し、効率的な管理や制御が可能となる。

以上の数式や考察は、さまざまな相互作用の種類やシミュレーションモデルに対応しており、具体的なシステムの設計や解析に利用可能である。

3.2.2 複雑化したデジタルツイン間の時間管理問題

複数のシステムを連携させた場合、時間軸の問題と同期問題が発生する。同期問題は、シミュレーション間で異なる計算時間や特性により時間のずれが生じる。この各要素間で発生するずれを考慮した計算方法が必要となる。

従来の提案方法としては、バリア同期による待ち時間を設けた実行方法がある。これは、複数のシミュレーションにおいて、計算時間が最も遅いシミュレーションを基準とし、他のシミュレーションは最も遅いシミュレーションが計算を終わるまで待機するという方法である。

待ち時間を使用したバリア同期によるシミュレーション方法のメリットは、計算時間が最も遅いシミュレーションだけを考慮するため、簡単かつ確実な同期を実現できる点である。デメリットとしては、他のシミュレーションは最も遅いシミュレーションの計算結果が出力するまで待機する時間が発生することである。

第4章 デジタルツイン連携基盤の提案手法

本研究では、デジタルツイン技術を活用した統合的な連携基盤を提案する。デジタルツインは物理空間のデータを仮想環境に反映し、リアルタイムで監視や制御を可能とする技術である。しかし、複数のデジタルツインが存在する環境では、それぞれが独立して動作することが多く、相互連携の不足が課題となる。本研究では、デジタルツインの連携を効率化し、データ共有、因果関係の管理、時間軸の調整を行う基盤を開発することで、これらの課題を解決する。

提案する連携基盤は、以下の主要機能を備える。

- **データ連携:** API を通じたデータ取得および情報共有を実施。
- **制御管理:** デジタルツイン間の関係性を管理し、相互作用による影響を制御。
- **因果関係の管理:** 接続性および時間性の観点から因果関係を解析し、適切な連携制御を実施。
- **時間管理:** シミュレーションや実世界の時間軸を調整し、履歴データと予測モデルを活用した同期制御を行う。

デジタルツイン間関係を管理するために、**属性付き有向グラフ**を用いる。これにより、各デジタルツインの接続性を明示的に管理し、以下の点を実現する。図4.2に連携基盤に接続したデジタルツインの連携基盤内における具体的な処理フローを記載する。本章では、提案するデジタルツイン連携基盤内の相互作用による問題を解決するための内部処理のうちに主要な3つの処理の処理と連携基盤の構成についての提案する。

- デジタルツインの関係をグラフモデルで分析。
- 相互作用の安定性を分析。
- 安定性に基づく関係性管理と時間管理手法。

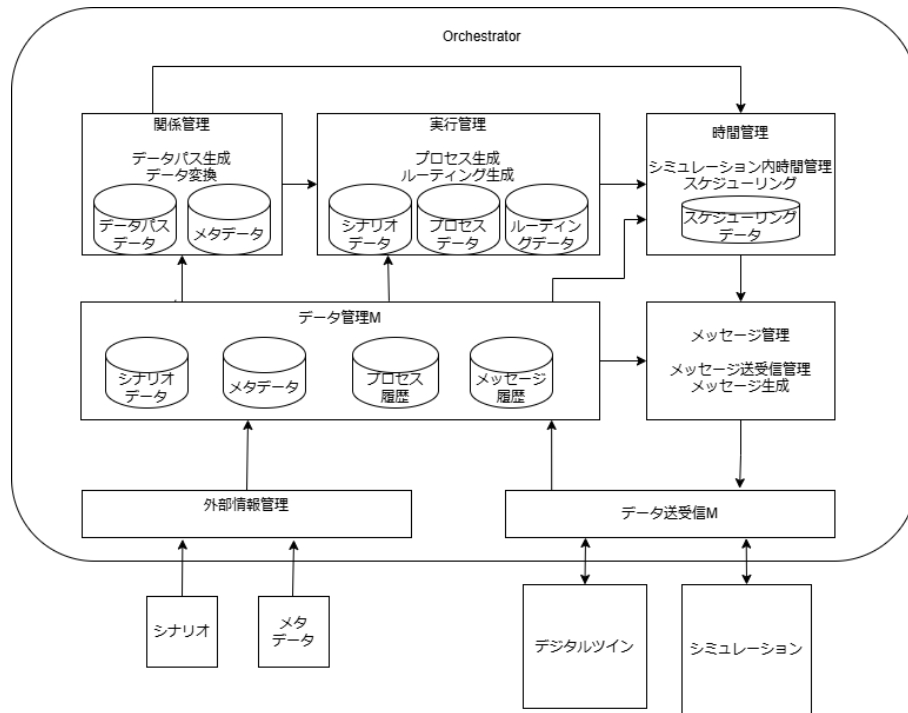


図 4.1: デジタルツイン連携基盤

4.1 関係グラフモデルによる相互作用性の解析

関係モデルグラフの構築と相互作用の関係の解析説明する。関係モデルは各デジタルツインのメタデータに記載された入出力情報をもとに構築する。

シミュレータに入出力が定義されているため、シミュレータをノードと考えることでシミュレーション間の関係を有向グラフとして考えることができる。そのため、シミュレータの網羅的なデータベースの構築は、有向グラフ探索における全探索問題として考えることができる。データベースの全探索において、考慮すべき条件は以下の三点である。1つ目が、シミュレータ間の関係性において、相互関係を含むため閉路について考慮する必要がある点である。グラフ内のノードが複数の隣接ノードを持つ場合において、それぞれの分岐を独立して探索する必要がある。2つ目が、シミュレーションの連携により一つのシミュレータに対して類似した入力が複数存在するため、分岐について考慮する点である。訪問しているノードを追跡し、同じ経路内での再訪が発生した場合、それが閉路と判断する必要がある。3つ目が、データベースとして使用するために経路をリスト化し、そのための全探索を行う必要がある点である。全経路を探索するためには、一度訪問したノードに対しても再度訪問する再帰的な探索が必要である。

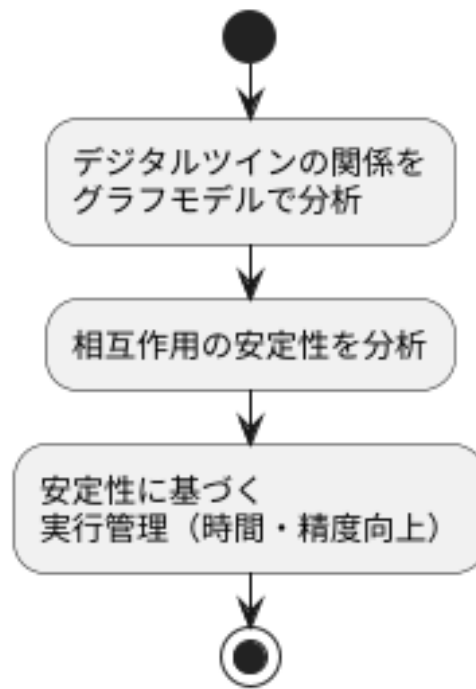


図 4.2: デジタルツイン連携基盤の処理フロー

これらを解決するための探索アルゴリズムは、再帰的な深さ優先探索を利用した方法である。これにより、デジタルツイン連携基盤に登録された各デジタルツインの関係を管理することができる。ここで、構築したデジタルツイン関係グラフモデルは、各デジタルツインのデータベース管理と、各デジタルツイン関係の解析による相互関係を検出に使用することができる。データベースの探索アルゴリズムには深さ探索法（DFS）を使用する。深さ探索にいて以下の条件を考慮する。
分岐の管理: グラフ内のノードが複数の隣接ノードを持つ場合、それぞれの分岐を独立して探索する。

閉路の検出: 訪問中のノードを追跡し、同じ経路内で再訪問が発生した場合、それが閉路であると判断する。

全探索の実現: 再帰的な深さ優先探索（DFS）を利用して、すべての経路を網羅する。

以下の図に探索アルゴリズムを示す。

分岐への対応: 隣接ノードをすべて順番に訪問し、再帰的に探索を進めることで分岐を網羅する。

閉路の検出: 現在の経路（path）に既に含まれているノードを再訪問した場合、そ

4.2 相互作用の実行管理

本研究では、デジタルツイン連携基盤内における連携と時間同期のために必要な要素を整理する。デジタルツインの相互作用を適切に制御し、時間同期を確保するために、予測モデル、シミュレーション、予測シミュレーション、接続データ変換モデル、履歴データベース、バリア同期が重要である。

予測モデル

予測モデルは、入力データに基づいて異なる時間の同一事象を出力する関数であり、再帰的な予測を可能とする。これにより、時間軸に沿った未来の事象を推測できる。時間 t の入力 A_t から、時間 $t+n$ の出力 A_{t+n} を得る式は以下の通り。

$$A_{t+n} = f(A_t, t, n) \quad (4.1)$$

ここで、 f は予測関数、 A_t は入力値、 n は予測時間差である。

シミュレーション

シミュレーションは、入力事象と相関する異なる形式の出力を生成し、システムの振る舞いを解析するために用いられる。時間 t の入力 A_t から、時間 t の出力 B_t を得る式は以下の通り。

$$B_t = f(A_t, t) \quad (4.2)$$

この関数の形式はシミュレーションモデルに依存する。

予測シミュレーション

予測シミュレーションは、予測モデルとシミュレーションの特性を組み合わせたモデルであり、異なる時間の結果を出力する。

$$B_{t+n} = f(A_t, t, n) \quad (4.3)$$

ここで、 f は予測関数、 A_t は入力、 n は予測時間差である。

接続データ変換モデル

接続データ変換モデルは、異なるデータ表現を持つ要素間でデータの変換を行う役割を持つ。異なる表現のデータ A'_t を入力とし、統一された形式の A_t を出力する。

$$A_t = f(A'_t, t) \quad (4.4)$$

ここで、 f はデータ変換関数であり、データの統一的な取り扱いを可能にする。

履歴データベース

履歴データベースは、過去のデータを蓄積し、予測モデルやシミュレーションに利用可能とする。これにより、時間同期の精度を向上させる。

$$A_t = \text{Retrieve}(A'_t, t) \quad (4.5)$$

ここで、Retrieve は過去のデータから適切な値を取得する関数である。

本研究では、これらの要素を統合したデジタルツイン連携基盤を構築し、異なるデジタルツイン間の時間同期と相互作用を最適化する。

要素	説明
予測モデル	現在または過去のデータを基に、将来の状態を予測するモデル。例えば、線形回帰、時系列分析などが含まれる。
シミュレーションモデル	実際のシステムを模倣して、異なる条件やシナリオにおける結果を予測するためのモデル。モンテカルロシミュレーションやシステムダイナミクスが含まれる。
予測シミュレーションモデル	予測モデルとシミュレーションモデルを組み合わせたアプローチ。予測結果をシミュレーションに組み込んで、より現実的な予測を行う。
接続データ変換モデル	各要素間のデータを変換し接続するためのモデル。データモデルやオントロジーが含まれる。
履歴データベース	過去のデータを蓄積するためのデータベース。データの履歴を保持し、予測モデルやシミュレーションに利用される。

表 4.1: 予測モデルと関連要素の説明

4.2.1 デジタルツインの連携

シミュレーション間の関係性について説明する。類似の出力結果を出力するモデルについて以下のように述べるができる。図??より、シミュレーションモデルにおいて、時間 t の時点での A_t と B_t を入力として、時間 t の C_t を出力する式、時間 t の時点での A_t 、 B_t 、および D_t を入力として、時間 t の C'_t を出力する式は以下のように表される。

$$C_t = \text{Simulate1}(A_t, B_t, t)$$

$$C'_t = \text{Simulate2}(A_t, B_t, D_t, t)$$

ここで、Simulate1、Simulate2 はシミュレーション関数であり、Simulate1 において、 A_t と B_t は時間 t における入力値、 C_t はシミュレーション結果として得られる出力値であり、Simulate2 において A_t 、 B_t 、および D_t は時間 t の入力値、 C'_t はシミュレーション結果として得られる出力値である。

ここで使用される変数 A_t と B_t は同様であり、 C_t と C'_t は同一の事象に関するものであるとしたとき、データモデルにより、 C_t と C'_t については同様に扱う

ことができる。しかし、Simulate1 と、Simulate2 に関しては、C という結果を出力するための入力変数が異なるため同様な関数として取り扱うことができない。

一方で、入力変数が E、出力変数が B'_t である関数 Simulate3 に関しては、シミュレーションモデルにおいて、時間 t の時点での E_t を入力として、時間 t の B'_t を出力する式は以下のように表される。

$$B'_t = \text{Simulate3}(E_t, t)$$

ここで、Simulate3 はシミュレーション関数であり、 E_t は時間 t の入力値、 B'_t はシミュレーション結果として得られる出力値である。 B_t と B'_t を変換するデータモデルを使用することで以下のように表すことが可能である。

$$B_t = f(B'_t, t)$$

$$C_t = \text{Simulate1}(A_t, \text{Simulate3}(E_t, t), t)$$

このようにある関数がデータモデルにより別の関数に対して接続できるようになることを、データ連携によるシミュレータの接続と考える。

4.2.2 相互作用するデジタルツインの連携

相互作用の定義としては、以下のように関数において A を入力とし出力を B とする関数 F1、B を入力とし、A を出力とする F2 がある場合これらをひとつの関数として取り扱う場合 F という 1 つの関数として取り扱うことで以下の図 4.4 のように表すことができ、これらの F 1、F2 はそれぞれ、F という関数の部分関数であると考えることができる。このように複数の関数をまとめて行列表現でき、それらの行列表現の中でたがいに共通の変数を取り扱う関係を相互作用と定義する。以下の 3 つのシミュレーションモデルが相互作用するときの行列表現を示す。

simulate 4 :

$$B_t = f_4(A_t, t)$$

simulate 5 :

$$C_t = f_5(B_t, t)$$

simulate 6 :

$$A_t = f_6(C_t, t)$$

これらの相互作用を行列形式で表すと、次のように表現できる。

$$\begin{bmatrix} A_t \\ B_t \\ C_t \end{bmatrix} = \begin{bmatrix} f_6(C_t, t) \\ f_4(A_t, t) \\ f_5(B_t, t) \end{bmatrix}$$

ここで、各関数 f_4 、 f_5 、および f_6 はシミュレーションモデルの関数であり、それぞれが対応する入力値に基づいて出力を計算する。

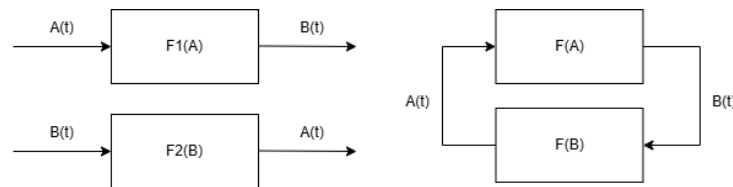


図 4.4: 相互作用 1

シミュレーションモデル 4, 5, 6 における相互作用を行列形式で表現する。シミュレーションモデル 4 は時間 t の時点での入力 A_t に基づき、時間 $t + n$ の出力 B_{t+n} を得るモデルであり、次の式で表される：

$$B_{t+n} = f_4(A_t, t, n)$$

シミュレーションモデル 5 は時間 t の時点での入力 B_t に基づき、時間 $t+n$ の出力 C_{t+n} を得るモデルであり、次の式で表される：

$$C_{t+n} = f_5(B_t, t, n)$$

シミュレーションモデル 6 は時間 t の時点での入力 C_t に基づき、時間 $t+n$ の出力 A_{t+n} を得るモデルであり、次の式で表される：

$$A_{t+n} = f_6(C_t, t, n)$$

これらのモデルは相互作用し、次の行列形式で表現される：

$$\begin{bmatrix} A_{t+n} \\ B_{t+n} \\ C_{t+n} \end{bmatrix} = \begin{bmatrix} f_6(C_t, t, n) \\ f_4(A_t, t, n) \\ f_5(B_t, t, n) \end{bmatrix}$$

上記の行列は、時間 t の各変数 A_t, B_t, C_t に基づいて、時間 $t+n$ の各変数 $A_{t+n}, B_{t+n}, C_{t+n}$ を計算するための関数がどのように相互作用するかを示している。

4.3 時間軸管理

連携基盤における時間軸の定義と管理手法について説明する。本研究で提案する連携基盤で扱う時間は複数存在する。現実世界の時間、シナリオ内の時間、シミュレーションやデバイス等の各要素内の時間の3種類である。現実世界の時間は最も絶対的な時間の進みがあるため、これを基準として他のシナリオ内の時間を考える。シナリオ内の時間の流れは、一つのユースケース単位で扱う場合、連携して動作するシミュレーションにおいて標準となる時間の流れであり、これを基準として各要素内の時間軸を相対的に考える。各要素内の時間の進みは、シナリオ内の時間の流れとは別に、各要素が持つ時間軸である。シミュレーションごとに特性が異なるためシナリオ内の時間の流れとは分けて感がある必要がある。

表 4.2: 時間管理に必要な要素

要素	役割
履歴データベース	過去の状態を記録し、システムの再現性を向上
予測モデル	将来のシステム状態を推定し、適応的な連携を可能に
時間スケールコントローラ	実時間、シミュレーション時間、シナリオ時間を適切に調整

4.3.1 スケジューリング管理

連携基盤内における時系列の整合性に関する管理手法について説明する。時間の流れを制御するにはシナリオ内の時間の設定によりユースケースの時間の流れが決定する。シミュレーション内の時間軸を基準とし、予測モデル、履歴データベースにより未来、過去の情報を使用することで、時系列の整合性を担保する。

デジタルツイン連携基盤におけるスケジューリング手法の要点として以下の3つがある。まず1つ目は、**未来時刻を扱う予測モデル**に関する制御である。予測モデルは、現在の状態から将来値を算出し、他モデルへの入力や次の時刻の初期条件として利用できる。出力と入力の次元が異なる場合には、変換関数を利用してデータを整形することが重要である。2つ目は、**過去時刻を参照する履歴データベース**の活用である。モデルによっては「過去時刻に戻って再計算する」あるいは「過去の実績データを再利用する」処理が必要であり、この際に履歴データベースを用意すると柔軟な再参照やエラー補正が可能となる。3つ目は、**バリア同期**を用いた時間合わせです。モデルごとに計算速度やステップ幅が異なる場合、「最も遅

いモデルに合わせて残りのモデルを待機させる」バリア同期により、全モデルが同一の時刻・イベント状態に到達するまで並列計算を調整し、不整合を防ぐ。

これらの手順をまとめたシミュレーションの進行フローとしては、大まかに次のようになる。

- **初期化:** 各モデルが必要とする初期状態（過去データベース、未来予測のパラメータなど）を設定する。
- **メインループ:**
 - － 各モデルが所定の時刻・ステップまで進み、予測モデルや過去参照モデルはそれぞれ未来／過去のデータを処理する。
 - － バリア同期により、全モデルが同じ時刻（またはイベント時刻）に達するまで待機し、同期後に必要があれば出力を他モデルへ転送する。
- **終了判定:** 全体の最終時刻や目的条件（例: 一定のシミュレーション時間や特定のイベント達成）を満たしたらシミュレーションを終了し、結果を解析・保存する。

このような仕組みにより、あるモデルが未来を推定し、別のモデルが過去のデータを再参照しながら計算を行っても、全体としてはバリア同期で時刻を調整するため、一貫した時間軸のもとで相互作用が行える。さらに、複数の予測モデルや過去参照が重複して存在する場合でも、上記の方法を組み合わせることで複雑な依存関係を整理でき、予測モデル・履歴データ・通常シミュレーションとの連携が可能になる。この一連のスケジューリング方法は、大規模なデジタルツイン環境や分散シミュレーションにおいて、モデル間の時間的整合性を保ちながら高い柔軟性と相互作用性を実現するために不可欠である。

4.4 ユースケースに応じたデジタルツイン連携

これらの要素をもとにして、ユースケースに合わせて関係性と時間を管理する必要がある。ここにおけるユースケースとは、使用するシミュレーションとそのタイミングを指定したものである。これらの情報を指定することでシミュレーションをはじめて運用することができる。シナリオ情報を指定するためには、各シミュレータの入力や出力をマッチングさせるための情報が必要になる。そのため、シミュレータの入出力を記したシミュレーションのメタデータを用意することで、シミュレータ間の入出力についてマッチングし、適切なデータパスを構築することができるようになる。連携可能なシミュレータのすべてのデータパスを網羅的にマッチングし、リスト化することで、ユースケースに合わせたシミュレーションによるデータパスの構築を容易に行うことができる。

4.5 デジタルツイン連携基盤の構成要素

提案するデジタルツイン連携基盤は、以下の6つのモジュールから構成される。

表 4.3: デジタルツイン連携基盤のモジュール構成

モジュール	役割
接続管理モジュール	デジタルツイン間の接続を制御
時間管理モジュール	時間同期および時間軸制御を行う
外部情報管理モジュール	外部データとの連携を管理
実行管理モジュール	実行可能性の判定とプロセス管理を担当
シナリオ管理モジュール	シナリオベースの連携制御
データ管理モジュール	データの保存および活用の最適化

4.5.1 データモデル

デジタルツインの統合的な管理と運用を実現するためには、データの整合性を保ちつつ、各デジタルツイン間での通信やデータ共有を適切に管理する必要がある。本研究では、デジタルツイン間の関係性を明示的に定義し、それぞれのデジタルツインの役割、データの流れ、相互作用を明確にするために、**メタデータモデル**、**入出力データモデル**、**シナリオデータモデル**の3つのデータモデルを提案する。これらのデータモデルは、デジタルツインの管理基盤として機能し、システ

ム全体の構造を整理するとともに、リアルタイムでのデータ通信や、シミュレーション環境における動的なデータ処理を可能にする。

以下では、それぞれのデータモデルの設計理念、構成要素、およびデジタルツイン連携基盤内での具体的な利用方法について詳細に説明する。

メタデータモデル

メタデータモデルでは、デジタルツイン連携基盤に接続するデジタルツインの情報について記される。メタデータモデルは、デジタルツインの識別情報や基本的な属性、通信プロトコルなどの管理を目的とする。本モデルの主な役割は、各デジタルツインの特性を統一的に管理し、基盤内での相互運用性を向上させることである。これにより、新しいデジタルツインの追加や変更を容易にし、動的なシステム拡張にも対応できる。

本モデルの主要な属性は以下の通りである。

- **twin_id**: デジタルツインを一意に識別する識別子。この ID を基に、他のデジタルツインとの関係性やデータ交換を管理する。
- **name**: デジタルツインの名称。人間が識別しやすい形で記述される。
- **category**: デジタルツインのカテゴリ（例：交通、エネルギー、環境など）。これにより、デジタルツインの用途や特性を分類し、適切な連携を促進する。
- **status**: デジタルツインの現在の状態（例：稼働中、停止中、メンテナンス中）。これにより、デジタルツインの動作状況をリアルタイムに把握できる。
- **location**: デジタルツインの物理的または仮想的な位置情報。地理的に関連するデジタルツイン間の連携に利用される。
- **communication_protocol**: デジタルツインが使用する通信プロトコル（例：MQTT、HTTP、WebSocket）。異なる通信方式を使用するデジタルツイン間の互換性を確保する。
- **mqtt_topic**: デジタルツインが通信に利用する MQTT トピック。MQTT ベースのデータ交換を実施する際に使用される。
- **control_topic**: デジタルツインの制御に使用するトピック。デジタルツインの動作を外部から制御するために利用される。

- **update_frequency**: デジタルツインのデータ更新頻度。データの新鮮さを維持し、最新の情報を取得するための指標となる。
- **last_updated**: 最後にデータが更新されたタイムスタンプ。データの履歴管理や変更の追跡に使用される。

メタデータモデルは、デジタルツインの統合管理を実現するために不可欠であり、新規のデジタルツインを容易にシステムへ組み込むための基盤として機能する。

入出力データモデル

入出力データモデルは、デジタルツイン間で送受信されるデータの構造を定義する。本モデルの目的は、異なるデジタルツイン間でデータを統一的なフォーマットで管理し、相互運用性を向上させることである。

本モデルでは、以下の属性を定義する。

- **twin_id**: データを送信するデジタルツインの識別子。
- **data_type**: 送信されるデータの種類（例：温度、電力消費量、車両速度）。
- **data_value**: データの実際の値。
- **timestamp**: データが生成または取得された時間。

本モデルの導入により、デジタルツイン間でのデータの一貫性を確保し、リアルタイム処理を容易にする。特に、異なる時間スケールを持つデジタルツインが連携する場合、タイムスタンプ管理が重要となる。

シナリオデータモデル

シナリオの定義について説明する。シナリオは、「想定するユースケースで使用する要素の指定」、「各シミュレーションの設定」の2つの要素から構成される。まず、想定するユースケースのシナリオで使用する要素の指定では、シミュレーション・モデル・データベース・デバイスについて指定する。次に、各シミュレーションの設定では、シミュレーションで使用する時間をはじめとした各要素を実行するために必要な項目について指定する。以上の2つの項目を指定することでユースケースに応じたデジタルツインやシミュレーションの実行が可能となる。

シナリオデータモデルは、デジタルツインが特定のシナリオやイベントに基づいて動作するためのスケジュールやアクションを定義する。本モデルは、デジタ

ルツインの動的な制御や自動化を可能にし、リアルタイムのシミュレーション環境や意思決定システムに適用される。

- **scenario_id**: シナリオの識別子。
- **twin_id**: シナリオに関連するデジタルツインの識別子。
- **execution_time**: シナリオが実行される時間（タイムスタンプ）。
- **action**: 実行されるアクションの内容（例：デバイスのオン・オフ、データ送信）。

シナリオデータモデルは、特定の条件下でデジタルツインがどのように動作すべきかを記述し、事前に決定されたルールに基づいた運用を可能にする。

4.5.2 データ管理モジュール

本研究で提案するデータモデルについて説明する。本研究で提案するデジタルツイン連携基盤において使用するデータモデルは、メタデータとシナリオデータである。これらのデータモデルは連携基盤内に外部情報として入力することを想定する。また、データ管理モジュールでは実行や各デジタルツインの状態についてのデータ管理も行う。

履歴データベース

連携基盤における共通データベースとその役割について説明する。共通データベースの役割は各デジタルツインのシミュレーション結果、デバイスの取得情報、実行結果を蓄積することである。共通データベースに蓄積した各デジタルツインのシミュレーション結果を使用することで、他のデジタルツインにおけるシミュレーションについても計算可能となる。デバイスの取得情報、シミュレーションの入出力結果によって、新たなシミュレーションの追加が可能となる。デジタルツイン連携において共通のデータベースを設けることにより、各デジタルツインが持つデータを共有し、利用することで独立したデジタルツインを連携することが可能となる。

4.5.3 時間管理モジュール

連携基盤内における時間の管理手法について説明する。本研究で提案する連携基盤で取り扱う時間に関する要素は、予測モデル、履歴モデル、予測シミュレーションである。予測モデルは入力した情報より、先の情報を出力する要素のことであり。履歴モデルは過去の情報を参照する要素であり、共通データベース内において適切な記録を参照することで、過去の記録を得ることができる。予測シミュレーションにおいては、入力した情報と異なる出力をするシミュレーションを予測予測モデルを組み合わせたものであるため、各要素の連携に構築が可能である。

4.5.4 接続関係管理モジュール

連携基盤における関係性の管理手法について説明する。関係性管理では連携基盤に接続されたシミュレーション、モデル、データベース、デバイスの要素間の関係について管理する。連携基盤に接続された各要素の情報はメタデータに記述することで連携基盤内で取り扱うことができる。メタデータには各要素の接続情報、使用するデータ形式、設定情報が記載される。

データ連携管理

データの入出力に関する管理手法について説明する。データの入出力については、MQTTBrokerによる入出力について管理する。各デジタルツインの要素に対するデータの入力と出力について監視することで、各要素間の連携において受け渡すデータをたがいに認知し、適切なデータデータを適切なタイミングで選択することができる。

4.5.5 シナリオ管理モジュール

連携基盤内におけるシナリオについて説明する。デジタルツインの連携において、連携基盤に接続した各デジタルツインやシミュレーションをユースケースを実際に連携して実行するためには、目的に応じてどのように連携するかを指定する必要がある。シナリオとは、デジタルツイン連携基盤においてどのように各コンポーネント（シミュレーション、モデル、データベース、デバイス）を使用するのか指定するためのデータである。シナリオにより、連携基盤と接続したシミュレーションを含む各種のコンポーネントを使用した実行が可能となる。

シナリオに基づいたシナリオ時間管理

シナリオに実行における時間軸をシナリオデータより設定し、各シミュレーションの実行に反映させる。シナリオ時間が著しく早い場合はシミュレーションの実行時間が間に合わず実行することができない。そのためシナリオ時間軸が実行可能か判定する必要がある。

シナリオに基づいたルーティング管理

シナリオの実行について説明する。シナリオ内の、「想定するユースケースで使用する要素の指定」により指定された各要素を実行するためのプロセスを作成する。プロセス作成のためのフローを以下に示す。プロセス作成のためには、リスト化されたデータパスより、ユースケースで使用する各要素と一致したデータパスを選択する。実行時には、シナリオ内に記述している「各シミュレーションの設定」で指定した情報を各要素に伝達する。以上の作業により、ユースケースに応じたデジタルツインやシミュレーションの実行が可能となる。

4.5.6 実行管理モジュール

連携基盤における実行管理モジュールの役割は、シミュレーションの実行プロセスを管理することである。以下で、プロセス管理する方法について説明する。

プロセス管理

プロセス管理では、シミュレーションの実行を管理するためにプログラムの起動と終了を管理する。本提案において対応する起動の種類は、周期的な起動と、イベントのような特定の条件による起動の2種類ある。周期的な起動に関しては、一定時間やプロセスが終了したとき周期的に実行する法である。イベントのような特定条件による起動は、あらかじめ実行する条件を設定し、その条件を満たしたときに実行する方法である。シミュレーションにおいて一定数以上のデータが必要な場合などが特定条件による起動にあたる。

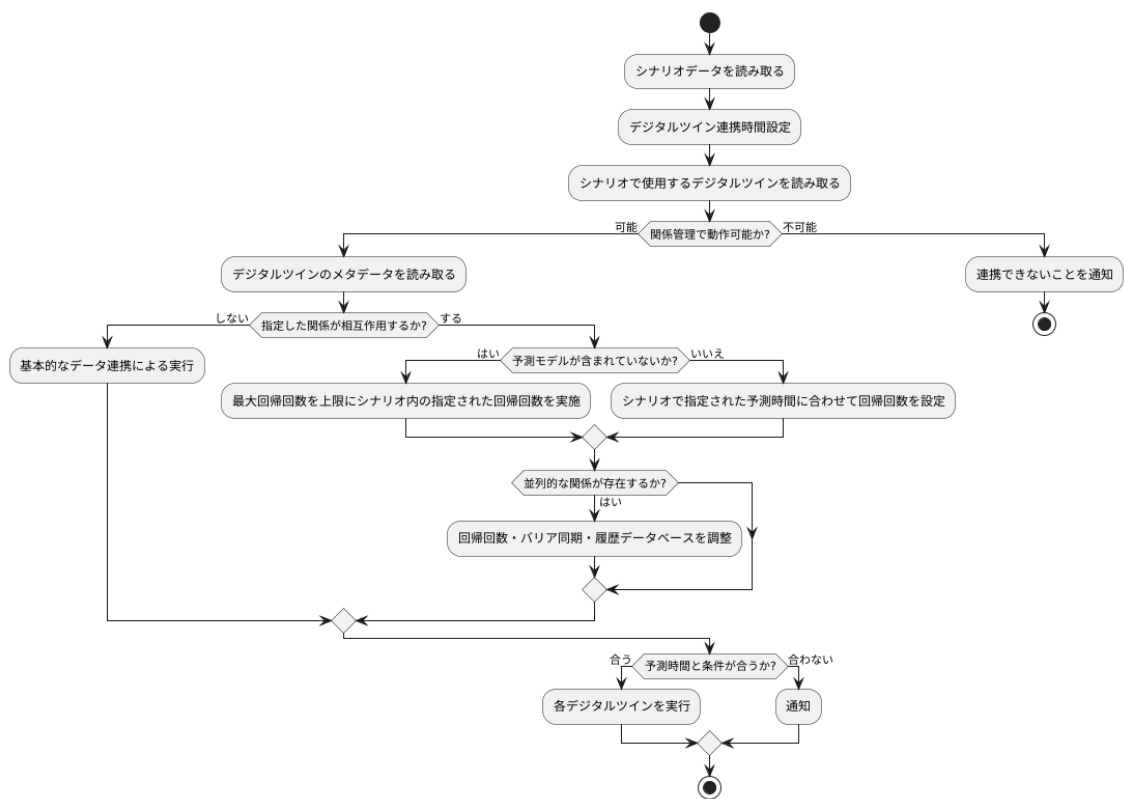


図 4.5: 時間制御処理

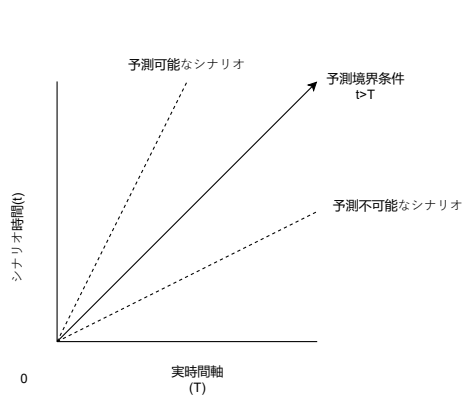


図 4.6: 実時間とシナリオ時間の関係

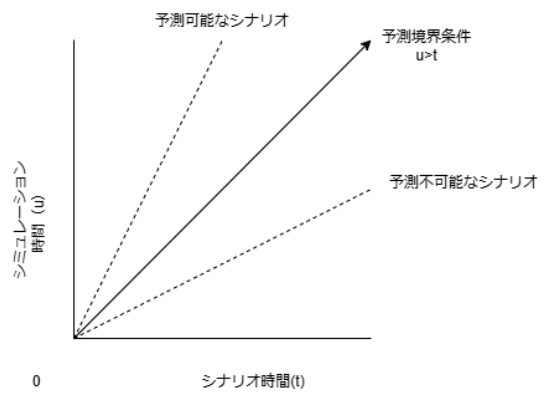


図 4.7: シナリオ時間とシミュレーション時間の関係

```
{
  "twin_id": "string",
  "name": "string",
  "category": "string",
  "status": "string",
  "location": "string",
  "communication_protocol": "string",
  "mqtt_topic": "string",
  "control_topic": "string",
  "update_frequency": 0,
  "last_updated": 0
}
```

図 4.8: メタデータモデル

```
{
  "twin_id": "string",
  "data_type": "string",
  "data_value": 0,
  "timestamp": 0
}
```

図 4.9: 入出力データモデル

```
{
  "scenario_id": "string",
  "twin_id": "string",
  "execution_time": 0,
  "action": "string"
}
```

図 4.10: シナリオデータモデル

第5章 実験

本研究における実験では提案手法で述べた相互作用性のあるシミュレーションの関係を管理し、想定するシミュレータの実行管理について実験する。連携基盤内のアルゴリズムとして、関係性管理に関する実験、時系列管理に関する実験、シナリオ管理に関する実験について実施する。これにより、連携基盤内において相互作用するシミュレーションの管理手法に対する有効性を示す。

5.1 実装

本研究において実装したデジタルツイン連携基盤のユースケースを図5.1に示す。本システムの利用者は2つのグループに分けられる。デジタルツイン連携基盤を利用するユーザーと、連携基盤に接続するデジタルツインを開発した開発者である。デジタルツインの開発者はデジタルツインの情報をメタデータとして連携基盤に登録する必要がある。デジタルツイン連携基盤の利用者は、連携基盤に使用するデジタルツインの使用方法を記したシナリオをデジタルツイン連携基盤に登録する必要がある。これにより、デジタルツイン連携基盤の接続した複数のデジタルツインを連携させた実行が可能となる。以下にデジタルツイン連携基盤の実装について説明する。

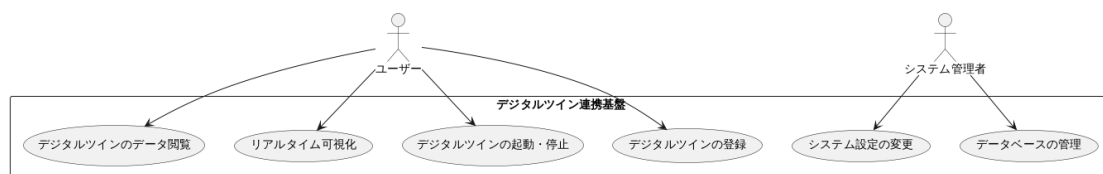


図 5.1: ユースケース図

5.1.1 システム構成と目的

デジタルツイン連携基盤は、都市デジタルツインの相互作用を効率的に管理し、リアルタイムデータ処理および依存関係の可視化を行うために設計された。本研究では、非同期処理とデータ永続化を活用し、登録されたデジタルツインを動的に制御する。

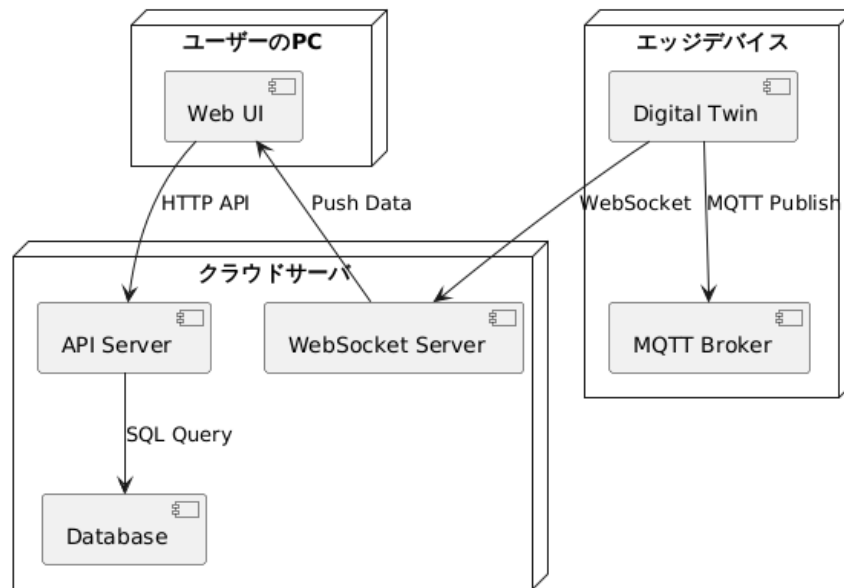


図 5.2: システム構成図

5.1.2 処理フローの設計

本システムにおける処理フローは、図 5.3に示すように、デジタルツインの登録、リアルタイムデータ処理、可視化までを含む。

5.1.3 モジュール間の連携

各モジュールの相互作用は、図 5.4に示すシーケンス図に基づく。具体的には、ユーザーの登録操作によりデジタルツインが動的に生成され、WebSocket を介したリアルタイムデータ更新が行われる。

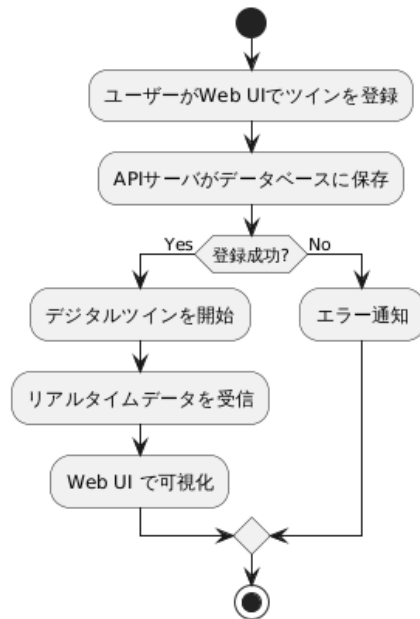


図 5.3: デジタルツイン登録の処理フロー

5.1.4 システム内部の設計

各コンポーネントの詳細は図 5.5 に示すクラス図で説明される。主な機能として、データベース操作、リアルタイムデータ処理、MQTT 通信をサポートする。

5.1.5 実験内容

本実験では、以下の手順でデジタルツインの登録および動作を検証する。

1. Web UI からデジタルツインを登録する。
2. API サーバが登録データをデータベースに保存する。
3. デジタルツインがリアルタイムデータを収集し、可視化する。

実行結果について API サーバーの実行結果を図 5.6 に、WEBAPI サーバーの実行結果を図 5.7 に、WEB の表示結果を図 5.8 に、テストデータ送信プログラムの実行結果を図 5.9 に示す。

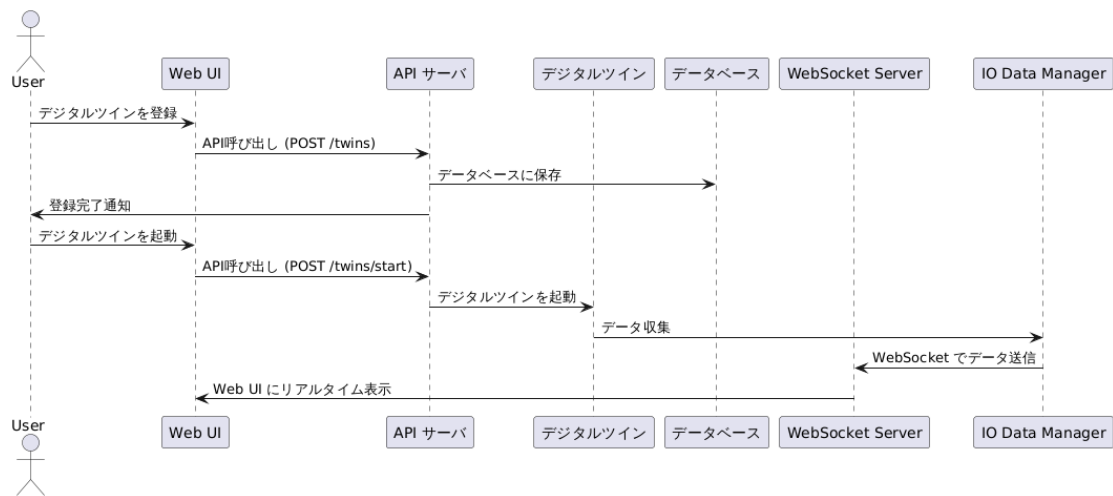


図 5.4: モジュール間のシーケンス図

5.2 都市デジタルツインのユースケース

本研究で想定するデジタルツインについて、スマートホームにおける居住者を中心に家電、家庭内電力、PV 発電を扱うスマートホーム、電力、天候をはじめとする環境、EVをはじめとするモビリティからなるデジタルツインの関係について以下について図 5.10に示す。

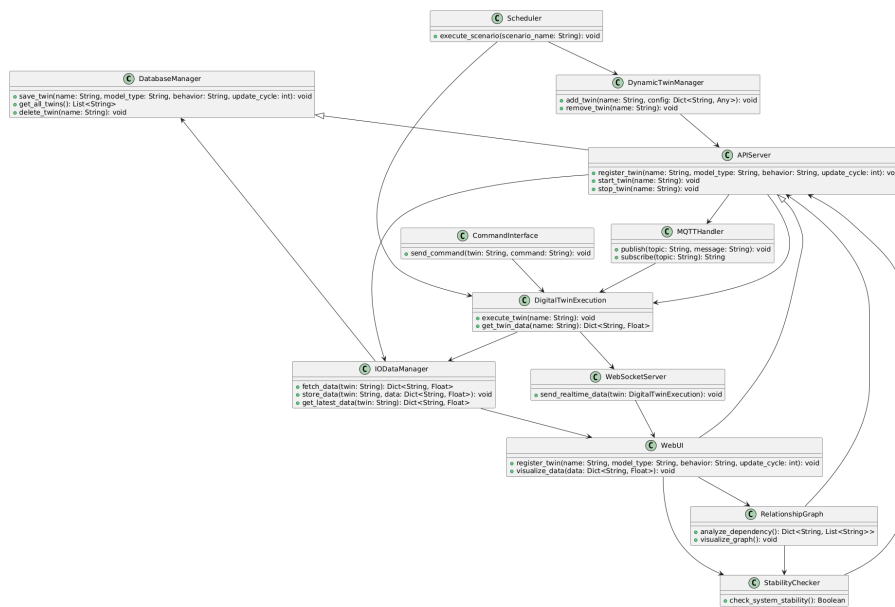


図 5.5: デジタルツイン連携基盤の構成

```

INFO: connection closed
✓ Twin: Twin A, Data: {'temperature': 25.3, 'humidity': 49.2}
✓ Twin: Twin A, Data: {'temperature': 25.3, 'humidity': 49.2}
INFO: ('127.0.0.1', 53231) - "WebSocket /ws" [accepted]
✓ WebSocket 接続成功 (API サーバー)
INFO: connection open
INFO: ('127.0.0.1', 53233) - "WebSocket /ws" [accepted]
✓ WebSocket 接続成功 (API サーバー)
INFO: connection open
INFO: ('127.0.0.1', 53232) - "WebSocket /ws" [accepted]
✓ WebSocket 接続成功 (API サーバー)
INFO: connection open
INFO: ('127.0.0.1', 53229) - "WebSocket /ws" [accepted]
✓ WebSocket 接続成功 (API サーバー)
INFO: connection open
INFO: ('127.0.0.1', 53230) - "WebSocket /ws" [accepted]
✓ WebSocket 接続成功 (API サーバー)
INFO: connection open
INFO: ('127.0.0.1', 53228) - "WebSocket /ws" [accepted]
✓ WebSocket 接続成功 (API サーバー)
INFO: connection open
INFO: connection open
✓ Twin: Twin A, Data: {'temperature': 25.3, 'humidity': 49.2}
  
```

図 5.6: API サーバーの実行結果

5.3 関係性管理の実験

連携基盤における関係性管理実行結果を示す。関係性の管理では以下の2つの処理について実験する。メタデータより連携基盤内に入力されたデータをもとに互いのシミュレーションの依存関係についてデータモデルを構築する処理。構築したグラフデータをもとに経路探索を実施し、経路テーブルと、相互作用のある巡回路テーブルの2つのデータパスに関するテーブルを作成する処理。これらの実験結果について示す。



図 5.7: 管理サーバーの実行結果

```
C:\Users\OGAKE Shuichiro\Desktop\EnviromentSmartHouse\orchestrater_v3>streamlit run web_ui.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://150.65.231.20:8501

✔ WebSocket 接続成功 (Web UI)
2025-02-04 02:06:15.847 Thread 'Thread-3 (start_websocket_listener)': missing ScriptRunContext! This warning can be ignored when running in bare mode.
```

図 5.8: WebUI の接続に関する実行結果

5.3.1 シナリオによるシミュレーション条件の設定

シナリオの実行結果について示す。本実験においてシナリオ実行では、以下の2つの処理について扱う。関係性管理より探索した関係から特定の関係を見つけ、相互作用関係を判定する処理。メタデータより、使用するシミュレーションの特性より収束性を判定し、最大再帰回数を設定する処理。シナリオデータより、目標予測やシミュレーションの予測について指定し、実行する処理。以上の3つの処理について実験する。

5.3.2 関係グラフモデルの作成処理

本実験において、各シミュレーションのメタデータより構築したグラフモデルを図 5.11に示す。

```

C:\Users\OGAKE Shuichiro\Desktop\EnviromentSmartHouse\orchestrater_v3\TestDT>python test_twin.py
✓ WebSocket 接続成功 (テストツイン)
送信: {'twin': 'Twin_A', 'data': {'temperature': 25.3, 'humidity': 49.2}}
送信: {'twin': 'Twin_A', 'data': {'temperature': 25.3, 'humidity': 49.2}}
送信: {'twin': 'Twin_A', 'data': {'temperature': 25.3, 'humidity': 49.2}}
送信: {'twin': 'Twin_A', 'data': {'temperature': 25.3, 'humidity': 49.2}}
送信: {'twin': 'Twin_A', 'data': {'temperature': 25.3, 'humidity': 49.2}}
送信: {'twin': 'Twin_A', 'data': {'temperature': 25.3, 'humidity': 49.2}}
送信: {'twin': 'Twin_A', 'data': {'temperature': 25.3, 'humidity': 49.2}}
送信: {'twin': 'Twin_A', 'data': {'temperature': 25.3, 'humidity': 49.2}}

```

図 5.9: デジタルツインテストデータの送信結果

5.3.3 シナリオによる実行可能性の判定

以下の図 5.1 にシナリオデータにより指定した関係がデジタルツイン関係グラフモデルより、シナリオで指定したデジタルツインの関係が実行可能か探索した結果を示す。ルートが存在すればシナリオで指定したデジタルツインが実行可能である。

表 5.1: シナリオデータの実行可能ルート検索

Target Route	Exists in Routes	Closed Loop
Environment (Influences) → Human (Uses) → SmartHomeAppliance	Yes	No

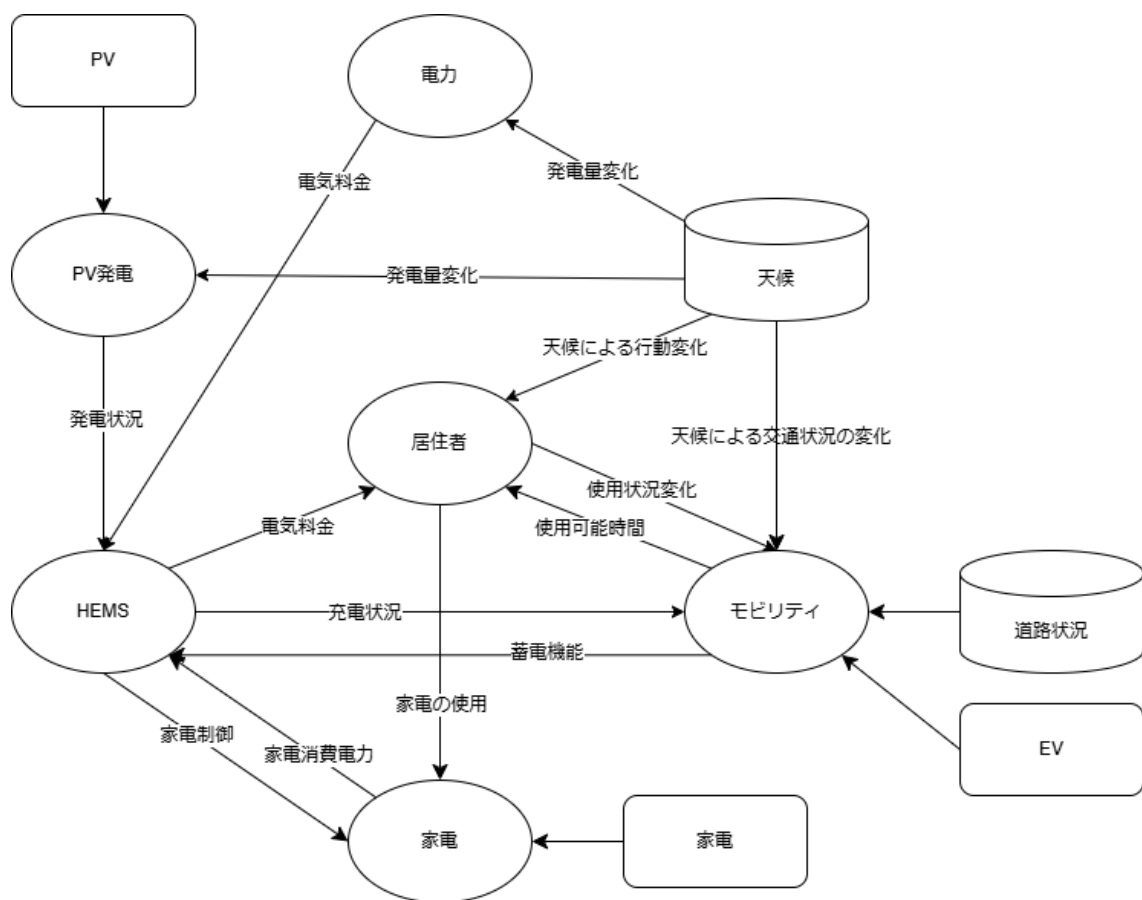


図 5.10: 都市デジタルツインユースケース

5.3.4 データパステーブルの作成処理

本実験において構築したグラフモデルより探索したデータパス経路と、その中でも閉路関係にあるデータパステーブルを以下に示す。

経路	出発ノード	接続・利用ノード	到達ノード
1	martHomePV (Reports)	SmartHomeHEMS (Connects)	martHomePV (Reports)
2	Mobility (Provides)	Human (Controls)	Mobility

経路	出発ノード	接続・利用ノード	到達ノード
3	Mobility (Provides)	Human (Uses) → SmartHomeAppliance (Communicates) → SmartHomeHEMS (Co- ordinates)	Mobility
4	Mobility (Supports)	SmartHomeHEMS (Monitors) → Human (Controls)	Mobility
5	Mobility (Supports)	SmartHomeHEMS (Coordinates)	Mobility
6	Human (Uses)	SmartHomeAppliance (Communicates) → SmartHomeHEMS (Monitors)	Human

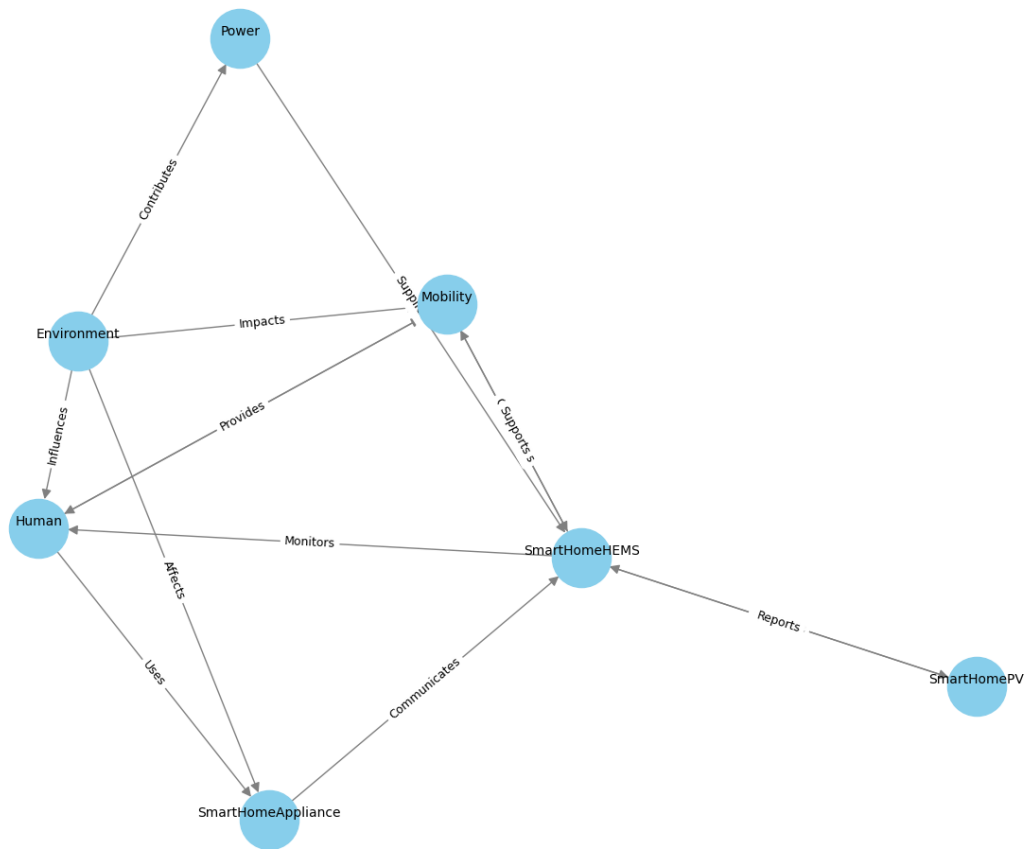


図 5.11: 関係グラフモデルの生成結果

5.4 時間管理の実験

時系列管理の実験の概要を説明する。シミュレーションについて以下のシミュレーションを対象として実験をした。デジタルツイン技術は、実世界のエンティティを仮想空間でシミュレーションし、リアルタイム監視や予測を可能にする。しかし、ノード間の依存関係やサイクル構造は、システムの応答時間や予測精度に影響を与える可能性がある。本研究では、5つのノードを持つシステムをシミュレーションし、依存関係が予測時間に与える影響を分析する。

5.4.1 相互作用関係における時間管理実験

ノードの構成

本研究では、各ノードに時間スケール (time scale) と 予測間隔 (predict interval) を設定し、シミュレーション時間を調整する。表 5.3 に各ノードの特性を示す。

$$B_{t+n} = \text{node}_1(A_t, t, n) \quad (5.1)$$

$$C_{t+n} = \text{node}_2(B_t, t, n) \quad (5.2)$$

$$A_{t+n} = \text{node}_3(C_t, t, n) \quad (5.3)$$

$$D_{t+n} = \text{node}_4(E_t, t, n) \quad (5.4)$$

$$E_{t+n} = \text{node}_5(F_t, t, n) \quad (5.5)$$

表 5.3: ノードの構成

ノード ID	名称	時間スケール	予測間隔 (s)
node1	A	1.0	5
node2	B	1.5	10
node3	C	2.0	15
node4	D	1.5	10
node5	E	2.0	15

ノード間の依存関係

ノードは特定の関係性（制御, 利用, 接続）を持ち、それぞれの時間発展に影響を及ぼす。特に $\text{node1} \rightarrow \text{node2} \rightarrow \text{node3} \rightarrow \text{node1}$ というサイクル構造を含む。表 5.4 にノード間の関係を示す。

表 5.4: ノードの依存関係

出発ノード	到達ノード	関係性
node1	node2	制御 (Controls)
node2	node3	利用 (Uses)
node3	node1	接続 (Cycle)
node4	node5	接続 (Connects)

シミュレーション条件

本シミュレーションは以下の条件で実施した。

- シミュレーション時間: 50 秒
- 基準時間ステップ: 1 秒
- 収束判定: 許容誤差 1×10^{-3}
- 最大反復回数: 10
- 予測時間上限: 各ノードに個別設定 (500～800 秒)

シミュレーション手法

各ノードのシミュレーション時間は、時間スケールを考慮して進行する。依存関係を持つノードは、最大依存時間を取得し、反復計算により予測時間を更新する。

相互作用関係における時間管理実験の結果

時間の管理について5つの予測シミュレーション結果に関して、図 5.12に示す。実験結果より、デジタルツイン相互作用関係を構築する予測については、逐次的な関係の連携に比べ、予測サイクル回数の制御により遠くの時間に関して予測が可能であることがわかる。

Algorithm 1 ノード間依存関係を考慮した予測時間計算

- 1: 各ノードのシミュレーション時間を初期化
- 2: 予測時間をゼロに設定
- 3: **for** $i = 1$ to max_iterations **do**
- 4: 各ノードの依存関係を取得
- 5: 予測時間を更新：

新予測時間 = $\max(\text{シミュレーション時間}, \text{依存ノードの予測時間}) + \text{予測間隔}$

- 6: 予測時間の上限を適用
 - 7: **if** 収束条件を満たす **then**
 - 8: **終了**
 - 9: **end if**
 - 10: **end for**
-

5.4.2 安定性を考慮した時間管理実験

実験では5.4.1で使用した式(5.1~5.2)のシミュレーションに対し、簡易的な線形モデル、非線形モデル、その他のモデルとして離散的なモデルを与え実験をした。5.3.1における収束判定と最大回帰数の設定処理において設置した、最大再帰回数をもとに計算をした。

シミュレーション条件

基準時間: 1.0 秒

予測モデル:

- 線形モデル (Linear): $S(t) = S_0 + 0.5 \cdot t$
- 非線形モデル (Nonlinear): $S(t) = S_0 + 0.1 \cdot t^2$
- その他モデル (Other): ランダム変動を加える

各ノードのシミュレーションは上記のモデルに基づいて行われ、ノード間の依存関係を考慮しながら計算が進行する。

Algorithm 2 ノードシミュレーションと予測時間計算

```
1: 入力: グラフデータ  $G = (V, E)$ , シミュレーション時間  $T$ , 収束許容誤差  $\epsilon$ , 最大反復回数  $I_{max}$ 
2: 出力: 各ノードの状態変化  $S(v, t)$  および 予測時間  $P(v, t)$ 
3: for 各ノード  $v \in V$  do
4:    $S(v, 0) \leftarrow$  ランダム初期値
5:    $P(v, 0) \leftarrow 0$ 
6: end for
7: for  $t = 1$  to  $T$  do
8:   for 各ノード  $v \in V$  do
9:     if  $v$  のタイプは線形 then
10:       $S(v, t) \leftarrow S(v, t-1) + 0.5 \times t$ 
11:     else if  $v$  のタイプは非線形 then
12:       $S(v, t) \leftarrow S(v, t-1) + 0.1 \times t^2$ 
13:     else
14:       $S(v, t) \leftarrow S(v, t-1) + \text{rand}(-0.1, 0.1)$ 
15:     end if
16:   end for
17: end for
18:  $\text{dependency\_map} \leftarrow$  エッジデータから依存関係を構築
19: for  $i = 1$  to  $I_{max}$  do
20:   更新フラグ  $\leftarrow$  False
21:   for 各ノード  $v \in V$  do
22:      $P(v, t) \leftarrow S(v, t)$ 
23:     if  $v$  に依存するノードが存在 then
24:       for 依存ノード  $u$  do
25:          $P(v, t) \leftarrow \max(P(v, t), P(u, t))$ 
26:       end for
27:     end if
28:      $P(v, t) \leftarrow P(v, t) + \text{予測間隔} \times \text{サイクル補正}$ 
29:     if  $|P_{\text{new}}(v, t) - P_{\text{old}}(v, t)| > \epsilon$  then
30:       更新フラグ  $\leftarrow$  True
31:     end if
32:   end for
33:   if 更新フラグ が False then
34:     終了 (収束)
35:   end if
36: end for
```

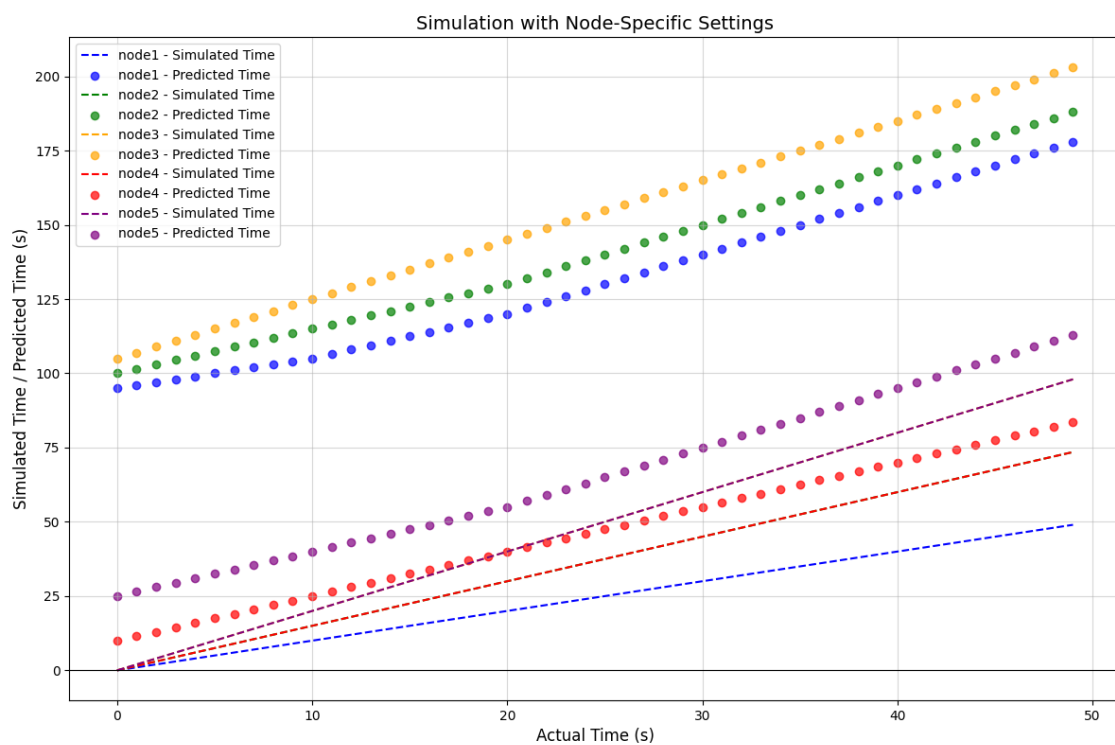


図 5.12: 相関関係のある予測シミュレーション結果

アルゴリズム

実験結果について、モデル間の関係をもとにした数値の計算結果グラフについて図 5.13に、予測した時間と実際の時間の関係について図 5.14に示す。

表 5.5: ノード情報

ノード ID	名称	タイプ	時間スケール	予測間隔
node1	A	線形	1.0	5
node2	B	非線形	1.5	10
node3	C	線形	2.0	15
node4	D	その他	1.5	10
node5	E	非線形	2.0	15

表 5.6: エッジ情報（ノード間の関係）

出発ノード	到達ノード	関係情報
node1	node2	制御 (Controls)
node2	node3	利用 (Uses)
node3	node1	接続 (Connects)
node4	node5	接続 (Connects)

表 5.7: ノードごとの予測時間上限

ノード ID	予測時間上限 (秒)
node1	10000
node2	1000
node3	10000
node4	1000
node5	1000

実験結果

シミュレーション結果を図 5.13, 5.14 に示す。あらかじめ設定したシミュレーションタイプ（線形、非線形、その他）のふるまいによりデジタルツインの安定性を判定した値をもとに最大回帰数を設定し、予測シミュレーションを実施した。実験結果より、最大回帰回数を適切に反映し、再帰的な計算の挙動を制御していることがわかる。

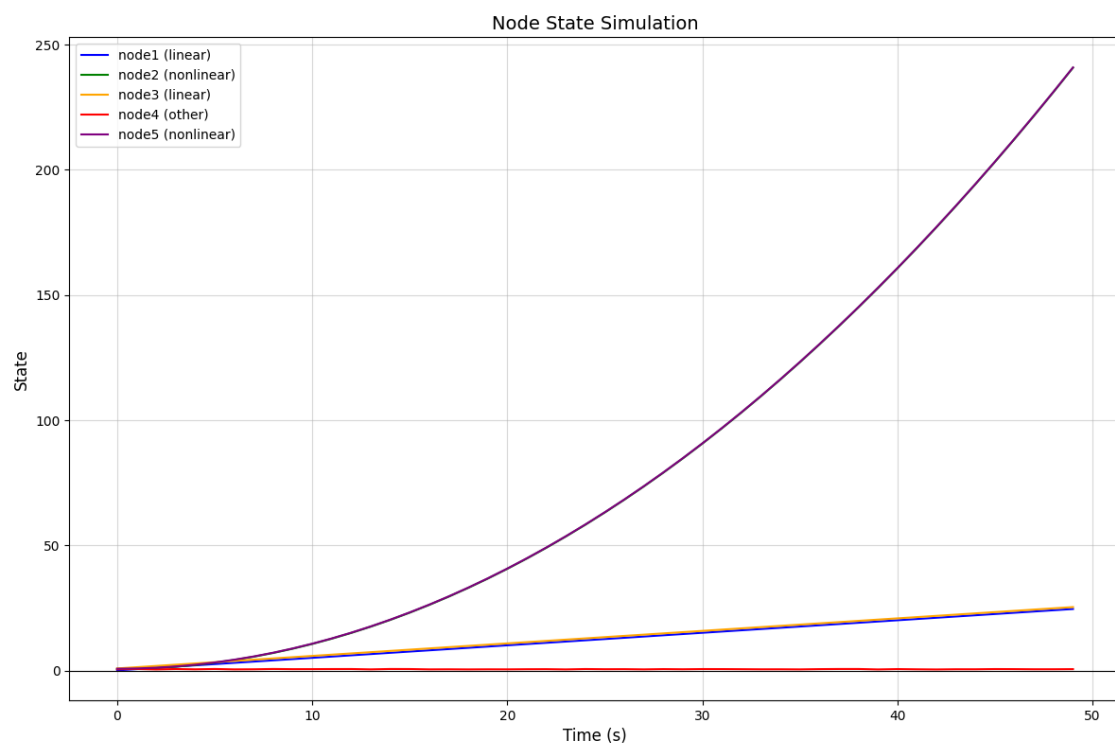


図 5.13: 相関関係のある予測シミュレーション数値変化の結果

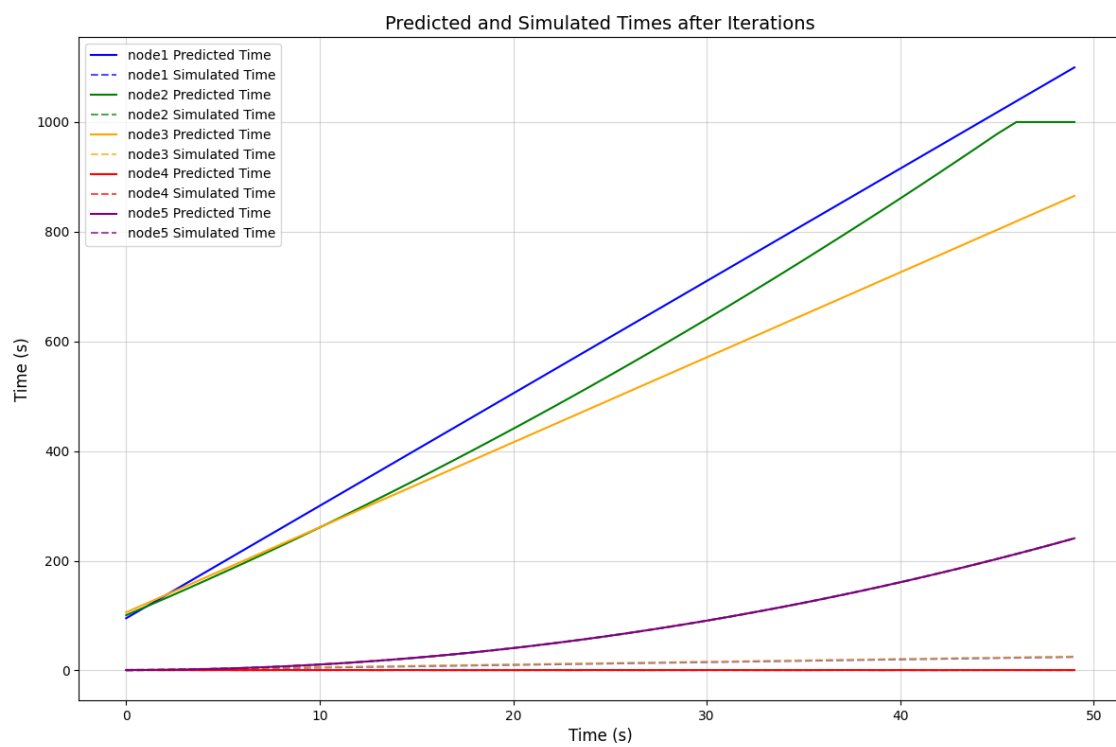


図 5.14: 相関関係のある予測シミュレーション予測時間の結果

第6章 評価

6.1 実験に対する評価

6.1.1 ユースケースに関して

本実験から以下の知見が得られた：

- ユースケースに対応したデジタルツインの関係グラフモデルをデジタルツイン連携基盤に接続した各デジタルツインメタデータより作成できた。
- シナリオで指定したデジタルツインの関係について、実行可能性と相互作用性について判定できた。

6.1.2 時間制御に関して

本実験から以下の知見が得られた：

- サイクルを含むデジタルツイン（node1, node2, node3）は予測時間が増大し、収束に時間を要する。
- サイクルを含まないデジタルツイン（node4, node5）は短時間で収束した。
- 予測時間の上限設定により、異常な発散を防ぐことができた。

6.2 結論

本研究では、デジタルツイン間の相互作用性による依存関係を考慮したシミュレーション手法を適用し、サイクル構造の影響や収束特性を分析することで課題解決に取り組んだ。本研究で提案したデジタルツイン連携基盤により相互作用するデジタルツイン連携における基本的な機能の実装と、相互作用の特徴である再帰性を利用したシミュレーションのための相互作用関係の安定性について評価し、デジタルツイン連携に組み込んだ。

6.3 課題に対する評価

相互作用するデジタルツインの連携基盤において、以下の課題解決に取り組んだ。

- 独立したデジタルツインの連携接続の問題
- 複雑化したデジタルツイン間の関係管理問題
- 複雑化したデジタルツイン間の時間管理問題

複数のシステム間の連携接続問題に対しては、API連携とデジタルツインメタデータによる連携基盤の実装により、データをモニタリングすることが可能となった。デジタルツイン間の相互作用の定義については、属性付き有向グラフモデルとして、連携基盤内で関係管理をし、経路全探索と閉路（相互作用関係）探索と、各でデジタルツイン間の相互作用をモデルとして取り扱うことで安定性の判断が可能となった。時間制御に関する同期問題及び時間軸の整合性の問題については、デジタルツイン連携において時間に関係する要素をモデル化し、時間に関係ある要素と相互作用について判定することで、各デジタルツインを同期するために必要なリスト作成アルゴリズムを提案した。以上より、相互作用するデジタルツイン連携をするためのデジタルツイン連携基盤の構築において必要な要素と処理について提案した。これらの提案をもと、より具体的なデジタルツインを対象とすることで実用可能な連携基盤の実現につながると考える。

6.4 考察

表 6.1では、本研究で定義した3つのデータモデルの目的、主要な属性、および活用例について比較を行った。デジタルツイン連携基盤を構築する上で、それぞれのデータモデルは異なる役割を果たし、連携のスムーズな実現に寄与する。

メタデータモデルの重要性

メタデータモデルは、デジタルツインの基本情報を統一的に管理する役割を果たし、特に大規模なシステムにおいてデジタルツインの識別や関係性の整理に重要な機能を提供する。各デジタルツインの識別子 (`twin_id`) を統一的に管理することで、異なるツイン間のデータ交換を一貫性のあるものにし、通信プロトコル (`communication_protocol`) の指定により、MQTT、HTTP、WebSocket などの異なる通信方式にも対応可能となる。さらに、デジタルツインの状態 (`status`) を管理することで、動作中のツイン、メンテナンス中のツイン、障害が発生したツインの区別を明確にし、連携基盤全体の監視と管理を効率化する。

また、更新頻度 (`update_frequency`) や最終更新時刻 (`last_updated`) を記録することで、データの鮮度を把握し、リアルタイム性が求められるシステムでの適用性を向上させる。例えば、スマートシティの管理システムにおいて、エネルギー管理、交通流管理、気象予測などの異なるデジタルツインが統合される場合、それぞれのデータ更新頻度を適切に設定することで、過去のデータに依存した意思決定の遅延を防ぐことができる。

入出力データモデルの役割

入出力データモデルは、デジタルツインが連携基盤を通じて送受信するデータのフォーマットを標準化し、異なるデータソース間の整合性を確保する。特に、異なるデータタイプ（例：温度、電力消費量、CO2 濃度など）を統一的に扱うための `data_type` 属性は、データ解析基盤や AI モデルとの連携において不可欠である。

また、`timestamp` を用いることで、異なるデジタルツインが異なる時間スケールで動作する環境下でも、適切なデータ統合を可能にする。例えば、スマートホームのデジタルツインが1分ごとに電力消費量を記録し、気象デジタルツインが1時間ごとに気象データを提供する場合、それぞれのデータが正しく統合されるように時間補正を行うことができる。

さらに、リアルタイム監視の観点から、センサーデータの収集に適用することで、異常検知や予測モデルの精度向上に寄与する。例えば、製造業における設備

表 6.1: デジタルツイン連携基盤におけるデータモデルの比較

データモデル	目的	主要属性	用途・活用例
メタデータモデル	デジタルツインの基本情報と通信設定の管理	twin_id (識別子), name (名称), category (カテゴリ), status (状態), location (位置), communication_protocol (通信プロトコル), mqtt_topic (MQTT 通信トピック), control_topic (制御トピック), update_frequency (データ更新頻度), last_updated (最終更新時刻)	デジタルツインの登録・管理、通信設定、状態管理、各ツイン間の関係構築、システム拡張時の統合管理
入出力データモデル	デジタルツイン間のデータ交換の統一化	twin_id (送信元識別子), data_type (データ種類), data_value (データ値), timestamp (取得時刻)	センサーデータの収集、リアルタイム監視、解析システムとの連携、予測モデルへのデータ提供、異なる時間スケールでのデータ統合
シナリオデータモデル	デジタルツインの動作シナリオの管理	scenario_id (シナリオ識別子), twin_id (関連デジタルツイン識別子), execution_time (実行時刻), action (アクション内容)	自動化制御、イベント駆動型アクションの実行、複雑なシミュレーション、異なるデジタルツイン間の同期動作、最適化アルゴリズムの適用

管理では、振動センサや温度センサのデータをリアルタイムで収集し、異常発生時に即座にアラートを送信するシステムを構築することが可能となる。

シナリオデータモデルによる動的制御

シナリオデータモデルは、デジタルツインが事前に定義されたルールに基づいて動作することを可能にし、システムの自動化や最適化を支援する。特に、`scenario_id` を基にシナリオを識別し、`execution_time` を指定することで、時間に依存した制御が可能となる。

例えば、交通管理システムにおいて、朝の通勤時間帯に信号制御アルゴリズムを変更するシナリオを設定することで、混雑の緩和を図ることができる。また、エネルギーマネジメントシステムでは、太陽光発電の発電量に応じて、蓄電池の充電や放電のスケジュールを動的に調整するシナリオを設定することで、電力の需給バランスを最適化することができる。

さらに、イベント駆動型のアクションを組み合わせることで、異なるデジタルツイン間の同期を実現することができる。例えば、気象予測デジタルツインが大雨の予測を発信した場合、インフラ管理デジタルツインが排水システムの制御を開始し、洪水リスクを低減するような統合的な運用が可能となる。

6.4.1 結論

以上のように、メタデータモデル、入出力データモデル、シナリオデータモデルの3つを組み合わせることで、デジタルツイン連携基盤の拡張性、柔軟性、実用性を向上させることができる。今後の研究では、これらのデータモデルの実装および検証を行い、異なるユースケースに適用することで、その有効性をさらに評価していく予定である。

6.5 今後の展望

本研究では相互作用のあるシミュレーションのデジタルツイン連携基盤について提案した。相互作用の特徴に着目した、連携基盤内での処理について提案した。本実験では、実際のデジタルツインではなく、デジタルツインの構造を持ったシミュレーションによる相互作用を管理するためのアルゴリズムの実験と、都市デジタルツインを想定したユースケースを使用した連携基盤の運用に関する一連の流れを実験した。今後の展望としては、大規模計算機を使用した実際の都市デジ

タルツインによる実証実験による、より現実に近い環境下での使用するための相互作用のあるデジタルツインを連携基盤の開発が見込まれる。そのためには、連携基盤内における様々なデジタルツインと連携するための汎用的なメタデータモデルの検討、大規模な連携に耐えるためのデータベースの強化や通信システムが必要となる。今後は、大規模ネットワークへの適用やリアルタイム最適化手法との統合、多様なデジタルツインと連携するためのデータモデルの検討を進める予定である。

第7章 おわりに

本研究では、独立したデジタルツインを連携を実現するために、デジタルツイン連携基盤と、連携基盤に登録したデジタルツインを管理するためのアルゴリズムについて提案した。提案したデジタルツイン連携基盤では、デジタルツインメタデータモデル、デジタルツイン入出力データモデル、シナリオデータモデルに登録することで、デジタルツインの管理とシナリオによる連携した運用が可能になった。提案したデジタルツイン連携基盤では、相互作用性のあるシステムを連携することにより発生する複雑化した関係を管理するために提案したデジタルツイン関係解析アルゴリズムでは、登録したデジタルツイン間の関係について解析し、連携可能なデジタルツインに関するリストと、相互作用関係にあるデジタルツインのリストを作成することができた。相互作用のあるデジタルツインの連携基盤において、提案したアーキテクチャと相互作用するデジタルツインを管理するためのアルゴリズムと運用方法を提案した。本研究では、デジタルツイン間の連携を支援するために、メタデータモデル、入出力データモデル、シナリオデータモデルを提案した。これらのデータモデルを活用することで、デジタルツインの統合的な管理が可能となり、システム全体の柔軟性と拡張性が向上する。今後の研究では、これらのデータモデルを実際のデジタルツイン基盤に適用し、その有効性を検証するとともに、さらなる最適化を図る予定である。今後の展望として、本研究で構築したデジタルツイン連携基盤を使用し、実際のデジタルツインシステムを構築し、より複雑な関係を組み込んだ都市デジタルツインの実現や、これらで大規模計算機を利用した検証を実施する予定である。

参考文献

- [1] T. Ruohomäki, E. Airaksinen, P. Huuska, O. Kesäniemi, M. Martikka and J. Suomisto, "Smart City Platform Enabling Digital Twin," 2018 International Conference on Intelligent Systems (IS), Funchal, Portugal, pp. 155-161, (2018)
- [2] Concetta Semeraro, Mario Lezoche, Hervé Panetto, Michele Dassisti, Digital twin paradigm: A systematic literature review, *Computers in Industry*, Volume 130, (2021).
- [3] F. Tao, H. Zhang, A. Liu and A. Y. C. Nee, "Digital Twin in Industry: State-of-the-Art," in *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405-2415, (2019).
- [4] M. Hermann, T. Pentek, B. Otto, 「Design Principles for Industrie 4.0 Scenarios: A Literature Review」, (2015).
- [5] B. Rodič, "Industry 4.0 and the New Simulation Modelling Paradigm", vol. 50, no. 3, (2017).
- [6] Concetta Semeraro, Mario Lezoche, Hervé Panetto, Michele Dassisti, Digital twin paradigm: A systematic literature review, *Computers in Industry*, Volume 130, (2021).
- [7] . Ríos, J. C. Hernandez-matias, M. Oliva and F. Mas, "Product Avatar as Digital Counterpart of a Physical Individual Product: Literature Review and Implications in an Aircraft", *Transdisciplinary Lifecycle Analysis of Systems*, pp. 657-666, (2015).
- [8] F. Pires, A. Cachada, J. Barbosa, A. P. Moreira and P. Leitão, "Digital Twin in Industry 4.0: Technologies, Applications and Challenges," 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, pp. 721-726, (2019).

- [9] H. van Houten, "The rise of the digital twin: how healthcare can benefit", (2018), [online] H. van Houten, "The rise of the digital twin: how healthcare can benefit", 2018, [online] Available: <https://www.philips.com/aw/about/news/archive/blogs/innovation-matters/20180830-the-rise-of-the-digital-twin-how-healthcare-can-benefit.html>.
- [10] A. Palau, "Digital Twins in cities: A Virtual Replica of Urban Networks", 2018, [online] Available: <http://www.resccue.eu/blog/digital-twins-cities-virtual-replica-urban-networks>.
- [11] Singh, M.; Srivastava, R.; Fuenmayor, E.; Kuts, V.; Qiao, Y.; Murray, N.; Devine, D. Applications of Digital Twin across Industries: A Review. Appl. Sci., 12, 5727. (2022).
- [12] Coorey, G., Figtree, G.A., Fletcher, D.F. et al. The health digital twin to tackle cardiovascular disease—a review of an emerging interdisciplinary field. npj Digit. Med. 5, 126 (2022).
- [13] Shahat, E.; Hyun, C.T.; Yeom, C. City Digital Twin Potentials: A Review and Research Agenda. Sustainability (2021), 13, 3386. <https://doi.org/10.3390/su13063386>
- [14] T. Obi and N. Iwasaki, "Smart Government using Digital Twin in Japan," (2021)
- [15] Gary White, Anna Zink, Lara Codecá, Siobhán Clarke, A digital twin smart city for citizen feedback, Cities, Volume 110, (2021). International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, pp. 1-4, (2021)
- [16] 椿 優里, 3D都市モデルの整備・活用・オープンデータ化“Project PLATEAU”, AI・データサイエンス論文集, 5 巻, 2 号, p. 1-12, (2024).
- [17] 総務省情報流通行政局情報通信政策課情報通信経済室, “デジタルツインの現状に関する調査研究の請負成果報告書”, pp. 14-15, (2021)
- [18] A.Grett and Xi.Wang, "Leverging a Digital Twin Interface for Multimodal Transportation Reslience, Connectivity, and Equity - A Case Study of Toyosu, Tokyo" International Conference on Aplied Energy (ICAE2024), Japan, Ni-gata, pp. 1-5, (2024)

- [19] E. A. Lee. Cyber Physical Systems: Design Challenges, Proc. of the 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'08), pp. 363–369 (2008).
- [20] P. Leitão and S. Karnouskos and A. W. Colombo. Industrial Automation based on Cyber-Physical Systems Technologies: Prototype Implementations and Challenges, Computers in Industry, vol. 81, pp. 11–25 (2016).
- [21] B. Rodič. Industry 4.0 and the New Simulation Modelling Paradigm, International Journal of Simulation Modelling, vol. 50, no. 3, pp. xx–yy (2017).
- [22] F. Pires, A. Cachada, J. Barbosa, A. P. Moreira, and P. Leitão. Digital Twin in Industry 4.0: Technologies, Applications and Challenges, 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, pp. 721–726 (2019).

.1 時間操作ベクトルと予測ベクトルにおける時間空間の証明

本研究における時間操作についての予測性能について証明する。本研究の時間操作では、与えられた時間操作ベクトル $N = [k, m]$ と予測ベクトル $P = [N + t, U + t]$ に関して、これらが生成するベクトル空間について証明を行う。範囲条件として、スカラー変数 t は $t > 0$ であり、また $\frac{T}{t} > 1$ が成立する。

.1.1 ベクトル空間の定義

ベクトル空間は、以下の条件を満たす集合である：

- ****加法の閉性****: 任意のベクトル $\mathbf{v}_1, \mathbf{v}_2$ がベクトル空間に含まれるならば、それらの和 $\mathbf{v}_1 + \mathbf{v}_2$ もベクトル空間に含まれる。
- ****スカラー倍の閉性****: 任意のベクトル $\mathbf{v}$ とスカラー c に対して、 $c\mathbf{v}$ もベクトル空間に含まれる。
- ****零ベクトルの存在****: ベクトル空間には零ベクトル（原点）が存在する。

これらの条件を満たすかどうかを確認するため、与えられたベクトル N および P が生成する空間について詳述する。

.1.2 ベクトル空間生成の証明

ベクトル N および P の定義

与えられたベクトル $N = [k, m]$ は、スカラー k および m に依存するベクトルであり、次のように定義される：

$$N = [k, m]$$

また、ベクトル P は次のように定義される：

$$P = [N + t, U + t]$$

ここで、 t はスカラーであり、 $t > 0$ の範囲で変動する。

加法とスカラー倍の閉性

ベクトル N と P の加法とスカラー倍の閉性を確認する。まず、任意の2つのベクトル $N_1 = [k_1, m_1]$ と $N_2 = [k_2, m_2]$ に対して、その和は次のように計算される：

$$N_1 + N_2 = [k_1 + k_2, m_1 + m_2]$$

これはベクトル空間内に含まれ、加法の閉性を満たす。同様に、スカラー倍 $cN = [ck, cm]$ はベクトル空間に含まれるため、スカラー倍の閉性も満たす。

零ベクトルの存在

$t = 0$ のとき、 P は次のように表される：

$$P = [N + 0, U + 0] = [N, U]$$

したがって、零ベクトルが生成されることが確認できる。

範囲条件 $\frac{T}{t} > 1$ の影響

範囲条件 $\frac{T}{t} > 1$ は、スカラー t が小さいときに T/t が 1 より大きくなることを示す。この条件により、ベクトル空間内でスカラー倍が効き、ベクトル空間が広がることが示唆される。

.1.3 結論

以上の証明により、与えられたベクトル N および P は加法とスカラー倍に対して閉じており、零ベクトルを含むため、ベクトル空間を生成することが確認された。また、範囲条件 $t > 0$ および $\frac{T}{t} > 1$ によって、ベクトル空間は広がり、所定の範囲内で閉じた構造を持つことが示された。

謝辞

本研究を執筆するにあたり、多くの方々から温かいご助言とご支援を賜りました。ここに深く感謝の意を表します。

まず、主指導教員である丹康雄教授には、日頃のゼミや研究の議論、発表資料の作成に至るまで、丁寧かつ的確なご指導を賜りました。教授の豊富な知見と貴重な助言のおかげで、本研究を進めることができました。心より感謝申し上げます。

また、審査員をお引き受けくださったリム勇仁准教授、宇多仁准教授、BEURAN, Razvan Florin 准教授には、中間発表やゼミ活動を通じて、新たな視点や示唆に富むご助言をいただきました。先生方からのご指摘や励ましが、本研究の完成に向けた大きな糧となりました。深く御礼申し上げます。

さらに、丹研究室の皆様には、日頃より通信分野に関する知見を深める機会をいただき、また、研究活動において多くのご協力と温かい支えを賜りました。皆様と共に学び、議論を交わした日々は、私にとってかけがえのない経験となりました。心から感謝申し上げます。

最後に、学生生活を支えてくれた家族や友人に、心より感謝の気持ちを伝えたいと思います。皆様の励ましや支えがあったからこそ、研究に専念し、ここまで歩んでくることができました。本当にありがとうございました。

本研究に関する対外発表

- 大懸崇一郎・リム勇仁・丹康雄, “相互作用のあるシミュレーションを統合する都市デジタルツインの基盤の研究”, 2024 年度電気・情報関係学会北陸支部連合大会 II. 1, Sep. 2024.