

CIP 개념 기반 디지털트윈 에이전트 구현

류현석¹, 송인용², 김종원³

¹광주과학기술원 전기전자컴퓨터공학부

²광주과학기술원 AI 대학원

³광주과학기술원 AI 대학원

{hyeonseokryu, inyong, jongwon}@smartx.kr

W

Digital Twin Agent Implementation based on CIP Concept

HyeonSeok Ryu¹, InYong Song², JongWon Kim³

¹Electric Electron Engineering and Computer Science, Gwangju Institute of Science Technology

²AI Graduate School, Gwangju Institute of Science Technology

³AI Graduate School, Gwangju Institute of Science Technology

요 약

최근 Digital Twin 과 Cyber-Physical-System 을 융합하여 시스템을 설계 · 예측 · 분석 · 최적화하기 위한 노력이 증가하고 있다. Unmanned Ground Vehicle 은 이런 방법론이 적용 중인 대표 예시이다. 본 논문에서는 Digital Twin 과 Cyber-Physical-System 을 융합하고 유지보수와 관리, 데이터 수집이 쉬운 시스템 설계 구조 Control-Infra-Product(CIP)을 제안한다. 시스템 계층을 사용자 관점에 따라 정의하며, 유저, 서비스 제공자, 제품(Physical, Virtual) 총 3 개로 분리된다. 각 관점은 Control Layer, Infra Layer, Product Layer 로 정의한다. 시스템의 사용 주체로 서버에 제어 명령 전달 및 전체적인 관제와 시각화를 담당하는 Control Layer 로 정의한다. Product Layer 는 Product 들을 묶어 정의되며, Product 는 현실과 가상에서 mirroring 이 될 두 개체 (Camera, Lidar, Car 등)를 하나의 쌍을 지칭한다. Infra Layer 는 서비스를 관리하는 주체로, 양단에서 Physical 과 Virtual 을 엮고 유저의 제어를 Product 까지 전달 및 전처리의 책임을 진다. 본 논문에서는 CIP 개념 구조의 적용과 Kubernetes 오케스트레이션을 활용한 시스템 구현을 통해 시스템 확장에 유연한 구조와 유지보수에 유용한 indoor 자율 주행 시스템을 개발하였다.

1. 서론

Digital Twin 에서 Physical Entity 는 그와 동일한 Virtual Entity 로 긴밀하게 mirroring 된다. 양쪽 Entity 에서 수집된 데이터는 Cyber-Physical-System(CPS) 기술을 활용해 Virtual Entities 를 시뮬레이션 한다. 이후 이에 대응되는 Physical Entities 에 대한 설계 · 예측 · 분석 · 최적화 등의 난이도를 낮추고 있다 [1].

초기의 CPS 은 안전의 관점에서 제안이 되었고, 위험하며, 안전이 요구되는 절차를 완전히 기계가 대체함으로써 노동자가 사망하는 것을 없애기 적용되었다. 이후 비행 시스템 등 비교적 큰 규모의 시스템에서 적용되며 종사자들의 안전을 보장하게 되었으나 그 외의 분야에서는 CPS 의 도입비용이 비싸고 시스템 제작 시간소요가 커 적용이 어려웠다[1].

하지만 5G 무선 통신과 3D 시뮬레이션 툴, AI 기술의 발달로 Virtual Entity 의 제작 시간소요가 단축되고, 공간 · 시설의 자동장애탐지와 같은 큰 규모의 지능형 시스템 구현에 필요한 센서, 장비가 가상에 구현됨에

따라 근래에는 다양한 분야에서 CPS 와 Digital Twin 을 접목시켜 설계 · 테스트하여 가상과 현실을 통합하는 한편 시스템 제작 비용을 낮추기 위한 다양한 노력이 점차 증가하고 있다.

그 중 Physical Entity 와 Virtual Entity 로부터 동시에 수집된 양쪽 데이터를 데이터 마이닝, AI 등에 활용해 전체적인 시스템(예:HAVC, 스마트 빌딩, UGV)의 성능을 고도화하기 위한 방법도 각광받고 있다 [2].

본 논문에서는 SLAM 이나 Lio-SLAM 알고리즘 논문에서 필요한 데이터를 Physical Entity 와 Virtual Entity 로부터 데이터를 수집하는 한편, 추후 데이터 베이스를 구축하는 등 논문자들이 다양한 시도와 변형을 통해 논문 할 수 있는 Digital Twin 환경을 설계하는 방법론인 Control-Infra-Product 을 제안하고자 한다.

구체적으로 실험환경의 Digital Twin 구현 및 실증은 2 단계인 자율 관제를 가능케하는 수준의 모사를 표방하며 [3], Physical Entity 인 ClearPath/Husky, GIST AI S7 건물, Husky 에 부착된 Lidar, Camera, IMU 센서

등을 Virtual Entity 에 Mirroring 하여 Digital Twin 을 구현한다.

이러 수집된 데이터는 구축된 데이터 파이프 라인을 따라 저장·가공·전처리 되며, 해당 파이프라인은 대규모의 분산 시스템으로 신뢰성이 검증된 Kafka 와 Data Distribution Service(DDS)를 적용 하였다. 마지막으로 가상환경 구현에는 Omniverse/IsaacSim, 제어에는 Next.js 웹 프레임워크를 하였다.

2. 관련 논문

a. 기존 디지털 트윈 논문 동향

기존 논문에서 Digital Twin 의 응용 가능성에 대한 다양한 논의가 진행되어 왔다 [2]. 특히, CPS 와 같은 최신 분야에서 Digital Twin 과의 결합을 통해 수학적 모델을 기반으로 자동화 시스템의 제작과 운영을 성공적으로 고도화한 사례들이 다수 입증되었다.

하지만, Digital Twin 의 구현과 이를 CPS 까지 연결하는 것은 여전히 복잡해, 진척이 잘 되지 않았다. 또한 다양한 Entities 가 존재하는 특성 상 멀티-모달 데이터 통합, Physical & Virtual Entities 간 상호작용 구현 난이도가 존재한다 [4, 9].

b. CPS 와 AI 통합의 어려움

AI 와 CPS 를 결합하는 방법은 지속적으로 제시되어 되어 왔으나 AI 블랙박스 현상으로 인해 안전성에 대한 의혹이 제기되어 왔다. 이를 해결하기 위한 방법으로 설명 가능한 AI(XAI)를 도입하는 경우가 일반적이다 [5].

또 다른 방법으로 AI 와 가상 시뮬레이션을 결합하여 시스템 안정성을 수차례 검증한 후 실제 환경에 적용 하는 방법이 있다. 특히 제조업에서는 로봇 팔 조립, 공장 환경의 자동화 장비 제어 등을 가상 환경에서 실증하는 사례가 늘고 있다. 이러한 방식은 추후 실제 설비 구축에 소요되는 비용과 시간을 절감할 뿐만 아니라, 물리적 오류로 인한 리스크를 최소화하는 데 기여하고 있다 [6].

c. CPS 구성 및 설계 방안

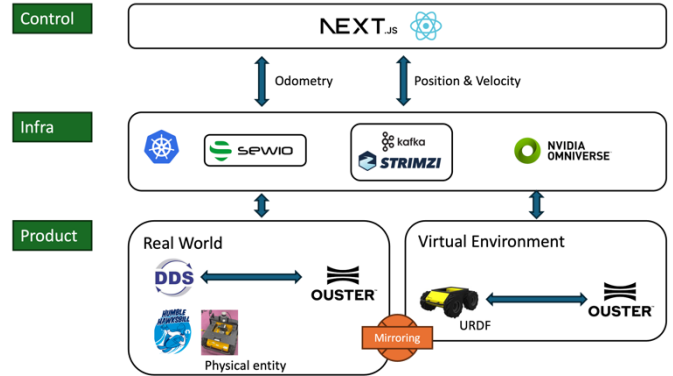
본 논문에서는 현실 Indoor 환경과 가상세계에서 Unmanned Aerial Vehicle(UGV)를 동시에 운용하는 환경을 구축한다. 해당 시스템은 상호간의 Odometry 를 동기화 시키며, IMU, Lidar, Camera 센서로 부터 자율 주행 생성되는 데이터를 효율적으로 수집하고 기기 제어를 쉽게 유지하는 시스템 설계법인, Control-Infra-Product(CIP) 구조를 제안한다. 시스템을 사용자의 관점에서 사용자를 유저, 서비스 제공자, 판매 제품으로 나누고, 각 사용자가 가장 많이 필요로하는 기능을

묶어 각각의 Control, Infra, Product Layer 로 정의하였다.

본 구조의 시스템은 기존의 인프라 프레임워크를 활용한 결합이 쉬워지고 추후 양쪽 Entity 의 추가 및 데이터 수집, 관제 및 시스템 장애를 파악하기 쉬워진다는 장점이 있다.

3. 설계

본 논문에서는 UGV 를 예시로 전체 시스템을 Control-Infra-Product (CIP) 구조를 채택해 설계하였다.



(그림 1) CIP System Design

- **Control Layer:** 유저가 Entities 에 필요한 제어 명령을 Infra Layer 에 전달하고, UWB, IMU, Lidar, Odometry, Camera 센서 데이터, 시스템 정보를 한눈에 확인하는 레이어이다. 이는 유저와 제품이 직접적으로 소통해야 하는 부분을 대체함으로써 Entities 의 추가시에도 시스템을 중단하지 않고 지속할 수 있어야 한다.

또한 Physical Entities 와 Virtual Entities 양쪽에서 수집되는 데이터를 통합, 시각화하여 관제와 모니터링 및 장애 탐지를 보다 쉽게 할 수 있도록 도와준다.

- **Product Layer:** Virtual/Physical Entity 쌍을 한 Product 로 정의한다. ClearPath/HuskyUGV 와 Nvidia/Omniverse 의 Husky URDF, Ouster/Lidar 와 LidarURDF 외에도 카메라 등의 다양한 Product 가 있다. 이러한 Product 들은 ROS2/DDS, Omnivers/inter process communication 을 통해 동기화된다. 동일한 Product 로 분류되는 Entity 들은, InfraLayer 에 정의된 프로세싱 방법으로 가공 및 제어되어 현실과 가상을 통합시키는 역할을 수행한다. 이 과정에서 각 Product 는 Infra Layer 대한 의존성을 가지게 된다.

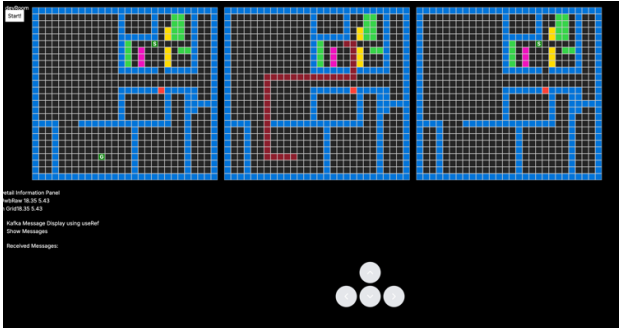
- **Infra Layer:** 유저가 생성한 제어명령을 QoS 에 따라 Product 에 전달하며, Physical Entity 와 Virtual Entity 로부터 수신 받은 Ouster, Lidar, IMU 센서의 데이터를 전처리 및 스트리밍, 데이터 베이스에 저장하며 동기화한다. 또한 전체적인 시스템의 운영과 관제의 품질을 제어하는 역할을 한다..

Control-Infra-Product 개념 기반 설계를 통해

Product 에 대한 제어를 인프라와 제품으로부터 분리하고, 제품의 세부적인 제어의 의존성을 Infra Layer 에 역전시킴으로써 유지보수를 쉽게 한다. 또한 Infra Layer 를 통해 오가는 현실과 가상의 데이터를 독립적으로 수집할 수 있다.

4. 구현

Control-Infra-Product 방법론에 따라 시스템을 구현하였다.



(그림 2) Position(A*) & Velocity Based Controller



(그림 3) UWB Sensor Map in NetAILab-1F

Control Layer 는 React/Next.js 를 이용해 속도와 위치 기반 제어, 관제 기능을 구현하였다. 속도는 speed control feedback loop 로 제어하며, 위치는 속도에 상한이 있는 P 제어(PID) Close Loop System 으로 설계되었다. 위치기반 제어를 위해 사용된 (그림 2) 위의 비트맵은 전체의 절대좌표를 소유하며, Husky 의 현재 좌표값을 표현한다(초록점).

이는 UWB(Ultra-Wide-Band) RLTS 센서가 Time Difference of Arrival(TDoA) 방식을 사용하여 GIST AI 융합 리빙랩의 Anchor 수신 범위 내에 있는 Tag 를 부착한 객체의 위치를 추적하고, UGV 의 실제 및 omniverse 상의 위치를 Infra Layer 에서 스트리밍 하였다. TDoA 방식은 Anchor 커버리지 구역내에서만 위치 추적이 가능하다는 단점이 있지만 단순한 통신 방식

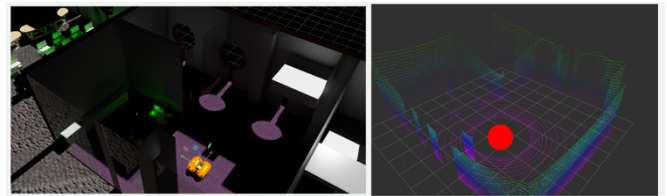
으로 Tag 의 배터리 소모를 줄이고, 시스템 확장성이 뛰어난 장점이 있어 도입하였다.

유저가 Web UI 에 접속하여 키보드와 같은 I/O 장치를 통해 Husky UGV 로봇을 조작할 때는 정의된 비트맵에서의 목표와 시작 위치의 좌표를 기준으로 A* 알고리즘을 적용하여 최단 경로로 움직이게 하였다.

Product 에서 Husky 가 계산한 측과 Odeometry 는 UWB, Infra Layer, Omniverse 의 측과 Odeometry 값과 다르기 때문에 이를 방법으로 초기해주어야 한다.

1. Align Two Coordinates():
2. for step < 3:
3. Husky Go front
4. Push HuksyOdeometry(Relative Position, Orientation) to HOarray
5. Push UWBDData(Absolute Position) to UWBarry
- 6.
7. Check gradient made by HOarray is 0
8. GetHOarrayUWBdata()
9. diffRadian = Get Radius between gradient of UWBarry and UWB map
10. setHuskyPos -= diffRadian

Husky 의 측과 Odeometry 가 현실과 가상에서 일치된 후 목표 UWB 좌표간 차이를 기준으로 이동하며 카메라, 라이다, IMU Product 에서 센싱되는 데이터를 스트리밍하는 종합 데이터 수집 파이프라인의 전송부를 만듦으로써 Product Layer 의 구현을 완료했다.



(그림 4) Data Collection From Omniverse

Infra Layer 는(Fig 5) MSA 구조를 지향하는 Omniverse 기술을 채택함으로써 URDF, USD, Teleoperator 와 같은 추가적인 기능, 객체 추가 및 Virtual Environment 확장이 용이하게 하였다 [7].

통신 데이터 처리는 Kafk, DDS 를 통해 지정된 QoS 를(예: 최대패킷전송시간-200ms) 만족하도록 처리하여 IoT 환경에서 발생하는 패킷 유실을 방어하였다 [8].

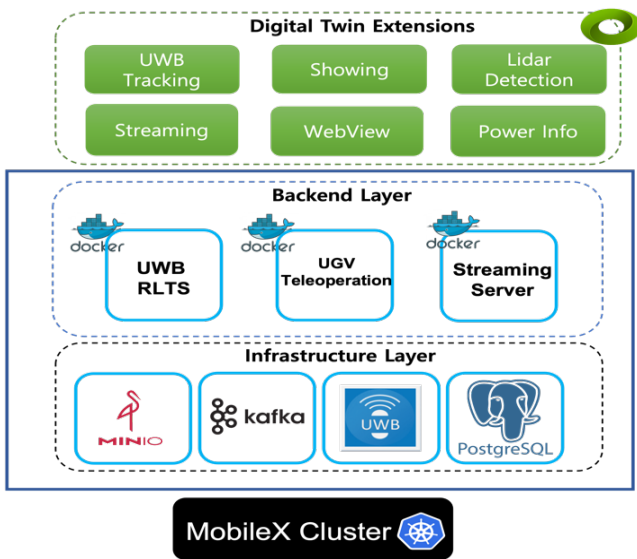
모든 통신 데이터와 센서 데이터, 제어 데이터는 MINIO 와 Postgres 에 저장된다. 데이터의 특징에 따라 고용량의 데이터는 MINIO 에 저장되고 트랜잭션이 중요한 데이터는 Postgres 에 저장되어 데이터 영속성과 시스템 장애 로그 기록을 책임진다. 이를 통해 데이터 검색, 시스템 관제 및 장애 시 분석을 위한 데이터를 안전하고 시스템의 에러 내성을 높일 수 있다.

UWB 정보 센싱 시스템은 컨테이너화 하여 배포함으로써 무중단 운영, 장애 탐지 및 좌표 추적 객체

의 추가를 유연하게 할 수 있도록 하였다.

Omniverse 내에서는 다양한 서비스가 MSA 형태로 유지 관리되고 있다. 이는 내부에 UWB 센서, Husky 원격 제어, Lidar 센서, 전력 센서, 스트리밍, 웹 뷰 (WebView) 등의 기능을 지원 및 구현하여, Physical/Virtual 을 통합하고 CPS 시뮬레이션 기능을 인프라와 로직상 분리하여 통합 및 관리 수 있었다.

그 외에도 React 와 같은 여러 서비스를 만들어 Infra Layer 내 배포하였으며, 이처럼 다양한 기능과 다양한 프레임워크를 각각 컨테이너화 하여 성공적으로 통합하였다. CIP 구조를 통해, K8s 클러스터에 배포되어 양쪽 데이터의 저장 및 처리, 운영 자동화를 달성하고, 추후 추가되는 요소들에 대해 유연하게 확장할 수 있었다.



(그림 5) Data Collection From Omniverse

5. 결론

본 논문은 CIP 구조로 설계된 가상과 현실의 Mirroring 을 구현했다는 점, 센서의 종류와 개수에 관계없이 수평적으로 Product 를 추가하는 것이 가능함을 확인 할 수 있었다는 점, 마지막으로 Camera, Lidar, IMU 데이터를 가상과 현실에서 동시에 수집하여 가공하며, 장애 대응과 운영 자동화가 가능한 Infra Layer 구성했다는 점, 손쉬운 제어가 가능한 비트맵 제어 Control Layer 를 만든 점에 의의가 있다.

실험을 통해 수집 UGV 의 가상 및 현실 자율주행 차량 데이터는 추후 UGV 논문함에 있어 데이터로 인한 병목현상을 해소 할 수 있도록 데이터를 배포하기 위한 노력이 진행중이다.

다음 논문에서는 데이터를 더욱 효과적으로 저장, 검색하는 파이프 라인을 구축하며 저장된 센서 데이터간 시간 오차를 보정을 추가하고 이를 통해 과거 재현이 가능한 데이터 베이스 및 파이프 라인을 구축

할 계획이다. 이를 통해 반복가능한 시뮬레이션 환경을 조성하고, 테스트 비용의 절감한 논문 환경에 기여 할 계획이다.

Acknowledgment

본 논문은 2023 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 논문임 (No. 2021-0-01176, 클라우드 기반 융합형 자율주행 지능학습 데이터 생성/제공을 위한 데이터 수집가공 핵심 기술 개발; No. 2019-0-01842, 인공지능대학원지원(광주과학기술원))

참고문헌

- [1] Platzer, A, Logical foundations of Cyber-Physical Systems. In Springer eBooks. 2018
- [2] N. Nikolakis, V. Maratos, and S. Makris, “A cyber physical system (CPS) approach for safe human-robot collaboration in a shared workplace,” Robotics and Computer-Integrated Manufacturing, vol. 56, pp. 233–243, 2019.
- [3] 정보통신기획평가원, 디지털 트윈 기술 K-로드맵-디지털트윈 기술 발전 5단계, 2022.02.06 [보고서].
- [4] J. M. Davila Delgado and L. Oyedele, “Digital Twins for the built environment: learning from conceptual and process models in manufacturing,” Advanced Engineering Informatics, vol. 49, p. 101332, Aug. 2021
- [5] M. Klumpp, “Innovation Potentials and Pathways Merging AI, CPS, and IoT,” ASI, vol. 1, no. 1, p. 5, Jan. 2018, doi: 10.3390/asi1010005.
- [6] A. Hoenig, K. Roy, Y. T. Acquaah, S. Yi, and S. S. Desai, “Explainable AI for Cyber-Physical Systems: Issues and Challenges,” IEEE Access, vol. 12, pp. 73113–73140, 2024, doi: 10.1109/ACCESS.2024.3395444.
- [7] Van Eenbergen, J., & Sawicki, P(NVIDIA), A Deep Dive into Building Microservices with Omniverse, GTC Session, 2021
- [8] T. Ribeiro De Souza, R. Owen, D. Mills, M. Reuter, A. Clements, and W. O’Mullane, “Replacing DDS with Apache Kafka as middleware technology for the Rubin Observatory control system,” in Software and Cyberinfrastructure for Astronomy VIII, G. Chiozzi and J. Ibsen, Eds., Yokohama, Japan: SPIE, Jul. 2024, p. 44.
- [9] G. Piras, S. Agostinelli, and F. Muzi, “Digital Twin Framework for Built Environment: A Review of Key Enablers,” Energies, vol. 17, no. 2, p. 436, Jan. 2024