Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

A HYBRID SIMULATION MODELING WITHIN A DIGITAL TWIN FRAMEWORK OF AGV-DRONE SYSTEMS

Rupesh Bade Shrestha[1] and Konstantinos Mykoniatis[1]

[1]Dept. of Industrial and Systems Eng., Auburn University, Auburn, USA

ABSTRACT

Integrating Autonomous Guided Vehicles (AGVs) and multi-copter drones, or AGV-drone systems, can potentially enhance efficiency and manipulation in intralogistics operations. While AGVs provide reliable, safe, and energy-efficient ground-based movement, drones offer rapid aerial mobility, making their collaboration promising. However, implementing this system is challenging as it involves collaborating with indoor drones. In particular, the testing phase can be unsafe and expensive because of the high probability of crashes due to the lack of technological advancement in this domain. Primarily, the communication protocols become complicated once the system scales. A digital twin (DT) methodology can help mitigate these risks and improve scalability. This study presents the simulation aspects of a work-in-progress DT for the AGV-drone system. Here, as a work-in-progress DT, the hardware-in-the-loop feature intends to help communicate within the system and build a DT testing environment.

1 INTRODUCTION

AGVs are autonomous transportation systems used for the horizontal movement of material using magnetic tape as their path (Vis 2006). Autonomous Mobile Robots (AMRs) are similar to AGVs but can plan travel while preventing collisions with environmental obstacles (Shrestha et al. 2024). This study uses the term AGV and can also apply to AMR.

While an AGV can manipulate objects in 2D space, a drone can provide 3D manipulation for the AGV. For that, a drone can be placed on top of AGV during navigation, and the drone will handle the manipulation tasks (Li et al. 2024). This will increase the manipulation field for AGV. On the other hand, drones can be self-sufficient in carrying loads from one place to another. However, in assembly plants, they will need to travel through many no-fly zones where people can be walking or working. Delivery through no-fly zones is a considerable challenge when using drones, as it will need to face many safety challenges (Fehling and Saraceni 2022). The noise and wind of these drones also generate mental stress for the workers when the drones fly on top of these workers (Yeh et al. 2017). For these reasons, using only indoor drones might not be a good option. Drones can rest on the AGV to skip no-fly zones, then start delivering the cargo whenever the AGV reaches the flight zone requiring delivery. Using multiple drones can make the job more effective in terms of delivery time and workload balance. Also, Bányai (2023) claims that using both AGVs and drones can be more efficient in terms of energy than when used individually.

The AGV-drone system is a new concept with high potential in intralogistics operations. The configuration of AGV and drone for this system is shown in Figure 1. However, if anybody wants to implement this system, it can be hazardous regarding safety during its testing phase as it involves using indoor drones (de Miguel Molina et al. 2018). Indoor drones encounter unique challenges that outdoor drones do not, and they have yet to reach the same technological maturity level as outdoor drones (Hell and Varga 2019). This can also increase the project's cost if the drone crashes every time we try to test new delivery strategies for intralogistics operations. Also, according to Munasinghe et al. (2024), the main challenge in using the

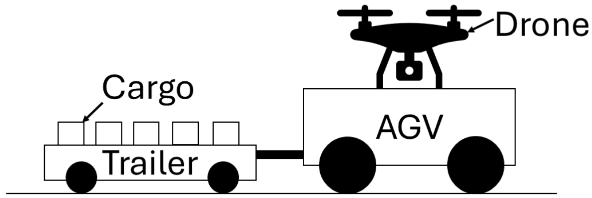

Figure 1: Configuration of AGV and drone in AGV-drone system.

AGV-drone system lies in the communication technology, coordination mechanism, and safety during the testing and operational phase.

Using Hardware-in-the-loop (HIL) simulation and DT can solve the problems mentioned above (Fathy et al. 2006; Barbie and Hasselbring 2024). HIL simulation integrates real and virtual prototyping by incorporating precise physical replicas of selected subsystems, enabling a bidirectional exchange of information between the physical components and the simulation model to accurately emulate the overall system (Fathy et al. 2006).

A DT is a virtual and dynamic representation of a physical product, process, or system, with the near-real-time bidirectional data exchange between its virtual and physical space (Grieves and Vickers 2017; Trauer et al. 2020; Mykoniatis and Harris 2021). If the data exchange is only one-directional, from physical space to virtual space, then it is called digital shadow (Grieves and Vickers 2017). DT enables real-time monitoring, simulation, and optimization based on near-real-time operational data. A DT is also considered one of the pillars of Industry 4.0 (Jiang et al. 2021). DTs integrates Internet of Things (IoT) and Artificial Intelligence (AI) to build smart, cyber-physical environments (Radanliev et al. 2022). According to Yao et al. (2023), DTs have three main functions: (1) data integration of required features between physical objects and virtual models with high fidelity, (2) physical and virtual components mutually exist and evolve together throughout the whole life cycle of the physical component, and (3) optimization, prediction and real-time control of physical components.

Grigoropoulos and Lalis (2020) have used a DT methodology on outdoor drones that improves operational safety and reliability through both pre-deployment testing and continuous runtime monitoring. Their system is designed to issue an alert if the DT's observed behavior deviates from the physical drone's behavior. Similarly, Martínez-Gutiérrez et al. (2021) developed a DT for AGVs to lower expenses by modeling its dynamic operations in industrial plants and forecasting potential problems in a Smart Manufacturing environment. These works suggest that the DT methodology can significantly help build an AGV-drone system.

DTs have recently gained popularity in the product planning and development life cycle. Huang et al. (2022) propose a digital-twin framework for product planning, validation, and manufacturing processes. They claim that the DT can accelerate the product development process. This makes it possible to conduct product validation in virtual space in a less expensive way than traditional product validation.

Similarly, Chandrasekaran and Rajesh (2025) and Lo et al. (2021) claim that DTs can effectively support the conceptual design, detailed design, design verification, and redesign phases of product development. In the case study of a medium-duty truck frame assembly provided by Chandrasekaran and Rajesh (2025), the application of synthetic DT data enabled a 50% reduction in design iterations, from six to three, while achieving a 9% weight reduction without compromising structural integrity. Lo et al. (2021) also provide some real-world examples in Maserati, Aurus, and Airbus, where using DT in the product development phase significantly reduces the development time. They claim that using DT in the product development phase can change the approach from a designer-driven approach to a "DT-driven" one. However, they found its use to be limited during the early conceptual design phase because of the limitations of its physical prototype.

By definition, although HIL simulation and DT look similar, HIL simulations are used particularly for prototyping, whereas DTs are used before and during operation. However, suppose we keep building the real-time interaction between physical and virtual components, prediction, and optimization on top of the HIL simulation. In that case, we can get a DT of that system, which will have a higher value in the long term because of its use even in the operational phase. This way, the HIL simulation can be called a DT prototype. Barbie and Hasselbring (2024) propose the development of DTs with the incremental process of DT prototypes. Instead of first developing the physical product and then making a DT of it, the methodology of building a DT prototype of a basic-level product with the minimum features, testing it, and then incrementing the prototype gives more benefits. According to them, product requirement updates and design flaw fixes can be done during the development phase with this methodology. This makes the DT prototype evolve continuously in a small increment until the final full-functioning DT is ready. They call this "Continuous Twinning". As the objective of this study aligns with this methodology, this study plans to build the DT of the AGV-drone system using this methodology.

While studies like Pantelidakis et al. (2022) and Katsigiannis and Mykoniatis (2024) build the DT from the existing product or system, Barbie and Hasselbring (2024)'s DT is about integrating the DT prototype from the beginning phase of the product or system development. If we first build the whole AGV-drone system, it can be expensive and unsafe because of the high chances of drone crashes during the testing phase (de Miguel Molina et al. 2018). Hence, the approach of building the whole AGV-drone system without a DT is not practical. Also, without a DT, scaling the communication protocol between drones and AGVs will be time-consuming when we want to add the number of drones (Kapteyn et al. 2021). The strategy shift will also require a change in the algorithm if we do not integrate DTs.

This study follows Emmert-Streib (2023)'s terminology, who claims the DT as solely the virtual component, and the DT System encompassing both physical and virtual components. In this study, a hybrid simulation model composed of Discrete Event Simulation (DES) and Agent-Based Modeling (ABM) handles the simulation aspects of the DT. Combining at least two kinds of simulation methods can be called a hybrid simulation (Mykoniatis and Angelopoulou 2020). DES best models a process-centric behavior where entities must compete to use resources (Caro and Möller 2016). ABM works better at modeling the complex model behavior and interactions (Dubiel and Tsimhoni 2005). At the same time, DES and ABM are similar in terms of time progression. ABM uses the DES method of time advancement, which is one of the reasons the creator of Simio claims ABM as a special case of DES (Law and Kelton 2015). The other reason is that DES models are starting to be capable of representing the agent's behaviors with entities (Brailsford 2014). However, the difference between DES and ABM exists mainly in the modeling perspective and objectives. They were also developed independently from each other (Macal and North 2014). So, although some DES modeling platforms have started giving agents' behaviors to the entities, we can still call the integration of DES and ABM a hybrid simulation approach.

Using a hybrid simulation of DES and ABM facilitates the benefits of both worlds. The initial development of the AGV-drone system only utilizes a single drone, so there is no need for queues. So, the whole model can be built with the ABM approach. However, this study considers queues for scalability so that we can use the same model with little updates if we want to use multiple drones. In this case, queues are required because it can be unsafe to land or pick up cargo with multiple drones simultaneously, considering the landing space and cargo space are limited in AGV. This demands the need for DES.

This study focuses on building the simulation part of the DT for the AGV-drone system. This is the unique contribution of this study because no work has been done to improve the coordination mechanism using DT technology. The study also focuses on building the simulation model with HIL features as a part of DT that defines the communication protocol in the AGV-drone system for intralogistics operation. The study is limited to explaining the communication protocols between an AGV, a drone, a simulation model, and an assembly cell. In this system, the AGV transports both the drone and the cargo from the home position to the assembly cell. Upon arrival, the drone takes off from the AGV, picks up the cargo, and delivers it to the designated assembly location. Once the delivery is completed, the drone returns and lands on the AGV, ready for the next cycle. This approach leverages the strength of both platforms: the AGV handles long-distance ground transportation with multiple no-fly zones, while the drone enables rapid and flexible delivery from the AGV. Combining these capabilities reduces overall delivery time, improves effectiveness, minimizes drone battery consumption (since the drone is not used for the entire journey), and ensures safer navigation by limiting drone flight to controlled areas. The basic layout of this system is shown in Figure 2.

The study provides a theoretical foundation; however, its scope is limited and does not encompass the complete operational life cycle of the AGV-drone system. For real-world deployment, it is essential to analyze and validate the system across the entire process, from pickup to delivery and return. However, this study does not present validation for the system's safety. Also, the study does not focus on the return of an AGV and a drone to the home position.

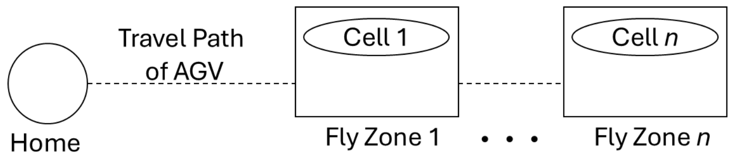

Figure 2: Basic layout of system.

2 SYSTEM INTERACTION

The communication protocol or system interaction between the virtual and physical components in the DT framework is shown in Figure 3. The main components of the DT system are AGV, drone, manufacturing cell, and simulation model. The study uses the User Datagram Protocol (UDP) method through WiFi for the initial testing of communication protocols. UDP is a simple, lightweight, connectionless protocol that enables applications to send messages with minimal overhead (Postel, Jon 1980).

The initial study assumes that the AGV uses a camera's object detection feature to track the drone's arrival near the AGV and the drone's landing. Photo-resistors are placed on the trailer to track the presence of cargo. When the cargo is on the trailer, it blocks the light from reaching the photo-resistor. Once the drone picks up the cargo, the space becomes empty, allowing light to reach the photo-resistor. This signals to the system that the cargo has been successfully picked up. Similarly, each manufacturing cell uses a camera to track the drone's arrival and the cargo's delivery. The drone also relies on its camera to navigate and land precisely on the AGV, guided by ArUco tags. ArUco tags are special visual markers that help drones and other systems pinpoint their exact location, handy for precise landings (Khazetdinov et al. 2021). To track its movement in the X, Y, and Z directions, it uses a 9-axis IMU sensor. The drone itself is equipped with a gripper for picking up the cargo.

Cameras and ArUco tags were selected to provide low-cost, robust visual tracking for navigation and landing. At the same time, photoresistors offer a simple, reliable mechanism to detect cargo presence without requiring complex load or visual sensors. Using a 9-axis IMU sensor further supports localization in GPS-denied environments, aligning with the goal of indoor operation. The data from the 9-axis IMU sensor will be corrected using the LIDAR sensor to get the precise displacement data. This combination of technologies reflects a design choice favoring modularity, low cost, and compatibility with real-time DT integration. Ultimately, the system is structured to demonstrate coordinated autonomy between ground and aerial agents in a hybrid logistics scenario, while also supporting scalability and experimental flexibility in a DT context. Here's how the entire system interacts:

2.1 The Start() Operation

The system starts by initializing the variables and agents in the simulation model. This also involves the establishment of a WiFi connection between all the components of the DT system.

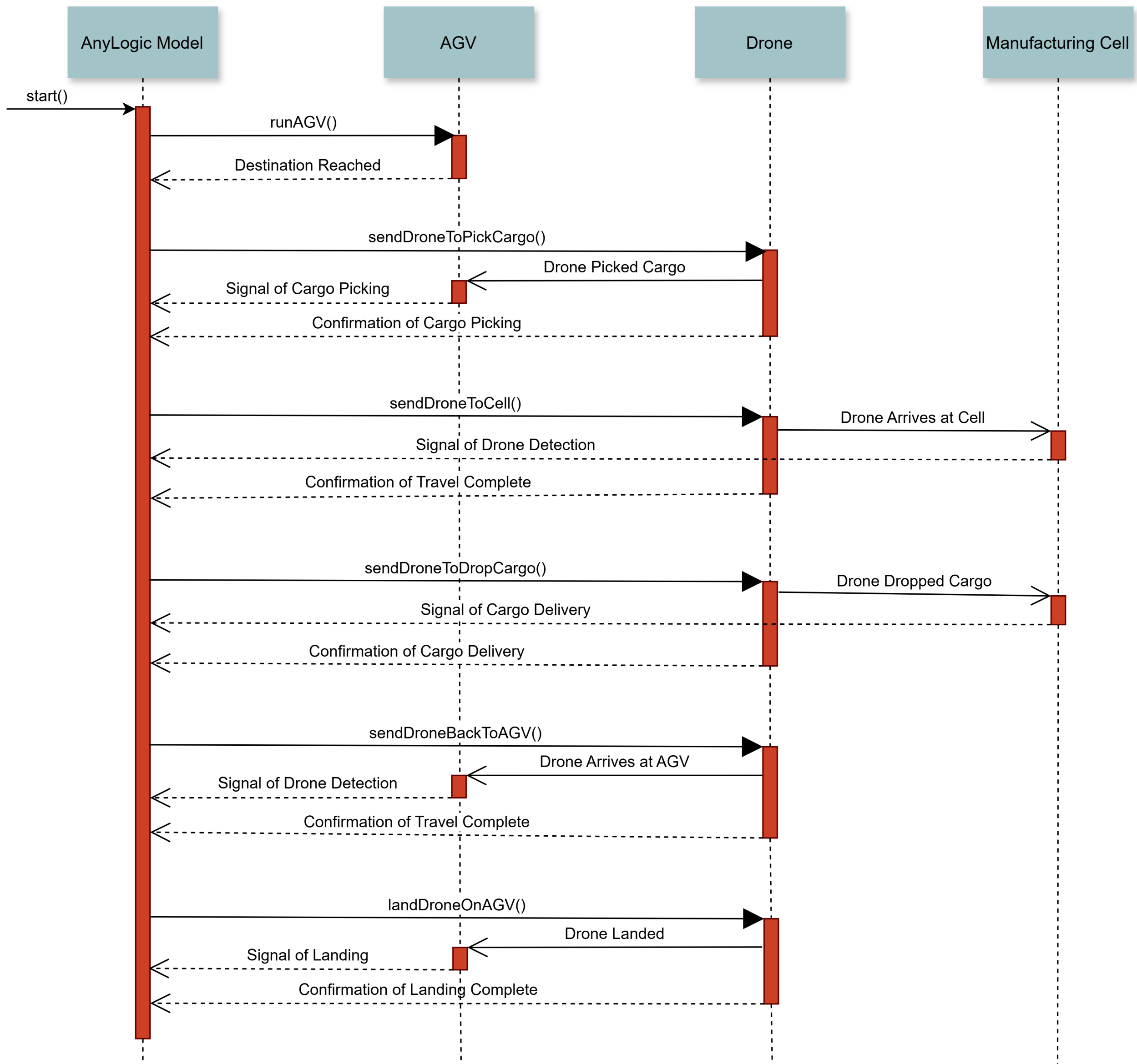

Figure 3: Sequence diagram for the work-in-progress DT of AGV-drone system.

2.2 The RunAGV() Operation

Here, the simulation model sends a message to the AGV to start heading to the assembly cell. An AGV moves towards the manufacturing cell while sending its displacement data to the simulation model using an encoder sensor in the wheel. The simulation model mirrors the position of AGV, and when AGV reaches its destination, this operation stops. Then, the system becomes ready to run other operations. The AGV uses a Hall effect sensor to detect fluctuations in the shape of the magnetic path. This helps to track the arrival at the destination. Since the system will know the precise arrival time of the destination, it can dispatch the drone for delivery with perfect timing. This significantly simplifies the coordination between the AGV and the drone.

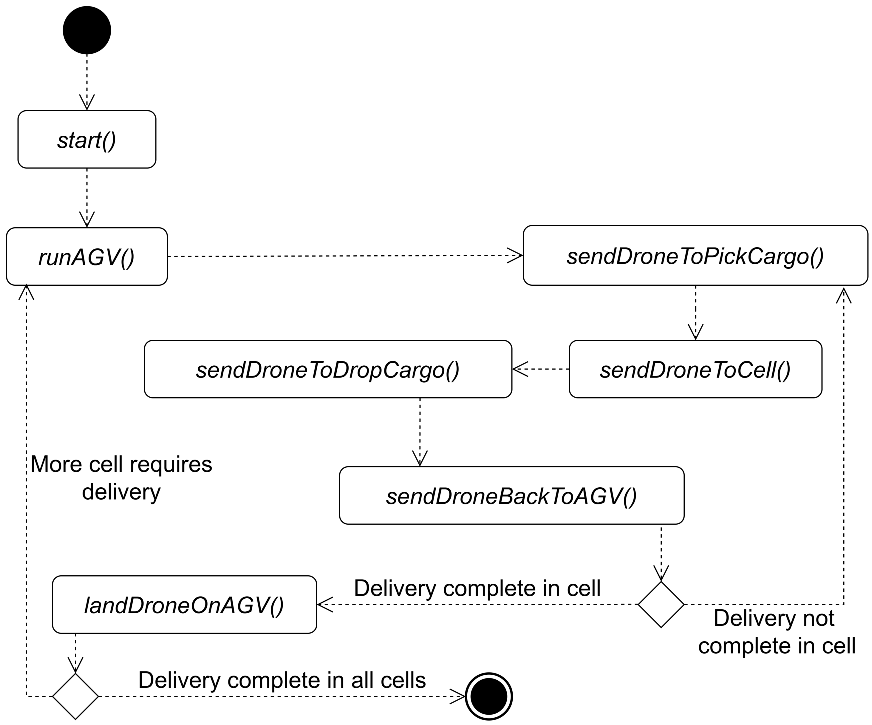

Figure 4: Activity diagram for the work-in-progress DT of AGV-drone system.

2.3 The SendDroneToPickCargo() Operation

The simulation model sends a message to the drone to pick up cargo. Here, the drone positions itself near the cargo (localizes) and picks it from the trailer using its gripper. The photo-resistor sensor in the AGV detects the cargo being picked up and sends the message to the simulation model. At the same time, since the drone finishes picking up the cargo, it sends a task completion message to the simulation model. This confirmation is essential to make the system robust. If the simulation model receives a message from both AGV and drone, then this operation stops, and the system becomes ready to run other operations.

2.4 The SendDroneToCell() Operation

The simulation model sends a message to the drone to head for the delivery location of that assembly cell. Here, the drone moves toward the delivery location of the cell while sending its X, Y, and Z coordinates to the simulation model using the 9-axis IMU sensor to mirror the drone's position. This operation stops when the drone reaches its destination, and the confirmation is also sent by the manufacturing cell after the camera in the cell detects the drone.

2.5 The SendDroneToDropCargo() Operation

The simulation model sends a message to the drone to drop the cargo in the delivery area. Here, the drone localizes near the delivery location and delivers the cargo. Once the delivery is complete, the drone sends a task completion message to the simulation model. A camera in the cell detects the cargo and sends the confirmation message to the simulation model after receiving the cargo. This confirmation makes the system more robust. If the simulation model gets a message from both the drone and the cell, then this operation stops, and the system becomes ready to run other operations.

2.6 The SendDroneBackToAGV() Operation

The simulation model sends a message to a drone to return to the AGV. Here, the drone moves toward the AGV while sending its X, Y, and Z coordinates to the simulation model using the 9-axis IMU sensor to mirror the drone's position. This operation stops when the drone reaches its destination physically and virtually, and the confirmation is also sent by the AGV after the camera in the AGV detects the arrival of the drone.

2.7 The LandDroneOnAGV() Operation

The simulation model sends a message to the Drone to land on top of the AGV. Here, the drone first localizes itself to the AGV and then lands on the AGV. The drone sends the task completion message to the simulation model. AGV detects the drone's landing through its camera and sends the confirmation message to the simulation model. This operation stops after the simulation model receives a task complete message from both AGV and drone. Then, the system becomes ready to run other operations.

If the manufacturing cell requires more cargo, the operation repeats from `sendDroneToPickCargo()` operation to `sendDroneBackToAGV()` operation as shown in Figure 4, until the delivery is complete. Similarly, if the system has multiple cells, this algorithm repeats from the `runAGV()` operation until the delivery is complete in the last cell.

The design of the AGV-drone system was guided by the need to balance energy efficiency, safety, and delivery precision in constrained environments. The AGV is used to transport the drone and cargo over longer distances, skipping no-fly zones, conserving the drone's limited battery for short-range, and precise deliveries within manufacturing cells. This also reduces airspace congestion and safety risks associated with long-range drone flight indoors.

3 SIMULATION MODEL FOR AN AGV-DRONE SYSTEM

The simulation model consists of four main parts, i.e., communication agent, AGV agent, Cell agent, and Drone agent. The study uses AnyLogic 8.9.3 Personal Learning Edition to build the model.

3.1 Communication Agent

A communication agent handles the communication between the physical system and other parts of the virtual model. The study uses the UDP method through WiFi for the initial testing of communication protocols. This method focuses more on sending the data quickly rather than reliably (Postel, Jon 1980). The communication agent receives the messages from the physical system, passes them to the decision logic, and then sends them to other model agents to update their state or block based on that logic. Also, when the corresponding state or block change occurs in the other agents of the model, those agents send the message to the communication agent. Then, the communication agent sends the message to the physical system to start a particular operation or correct specific actions (usually when the drone is trying to align itself with the landing zone of AGV and the cell).

3.2 AGV Agent

The AGV agent controls the movement of the AGV from the home position to the manufacturing cell position. The study uses the ABM approach to model the internal behavior of AGV as shown in Figure 5. It consists of three states, i.e., *Delivering* state, *AGVTravel* state, and *RobotWaiting* state. *AGVTravel* state and *RobotWaiting* state come under the *Delivering* state as shown in Figure 5. In the *AGVTravel* state, the AGV agent mirrors the physical AGV using its displacement data. For this data, the physical AGV sends the encoder sensor data to the communication agent, which then sends it to the AGV agent. Once the AGV agent reaches specified coordinates in the model and gets the confirmation from the communication agent, the state changes to the *RobotWaiting* state. A physical AGV sends this confirmation message by detecting its magnetic path using a Hall effect sensor. This state triggers the cell agent to create the drone entity from the *sourceDrone* block.

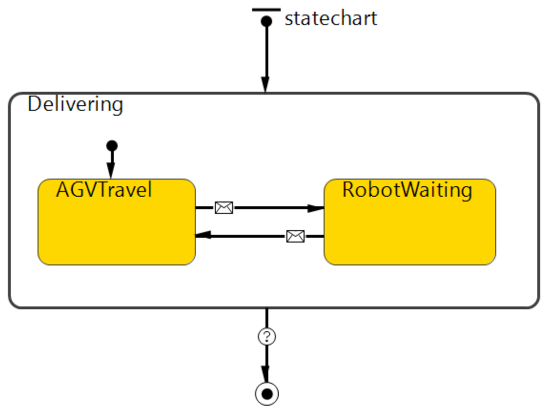

Figure 5: State chart of AGV agent.

Once the replenishment tasks in the cell are complete, the cell agent sends the message to the AGV agent through the communication agent, and the state of the AGV agent goes back to *AGVTravel* state if there is a need for replenishment or delivery in more cells. Once the delivery is complete in all the cells, the AGV agent exits the *Delivering* state.

3.3 Cell Agent

The cell agent controls the movement of drones within the manufacturing cell. The study uses the DES approach to model process-centric behavior as shown in Figure 6. Once the *sourceDrone* block receives the signal from the AGV agent, it creates the corresponding quantity of entities and binds them to the individual drone agent. This binding is done to connect the entities to the respective drone agents. This connection helps the state change in targeted drone agents, which affects the holding of individual entities in the queue blocks. The queue blocks are placed before hold blocks to accommodate multiple drones, where hold blocks do not let any entity pass through the queue until the *unblock()* function is called for that block. However, the initial phase of developing the DT will only focus on a single drone.

Entities being blocked because of a hold block represent a particular activity being done by a drone. In the *LocalizingAndPicking* hold block, the drone picks up the cargo from the trailer. The drone travels to the cell in the *Traveling1* hold block. In *LocalizingAndPlacing* hold block, the drone delivers cargo to the station of the cell. After the model receives the confirmation from the physical system, the drone travels back to AGV in *Traveling2* hold block. Then, the drone localizes, lands on the AGV, and sends the message to the AGV agent if the replenishment is complete in this cell. Otherwise, the drone goes to the trailer and picks up the cargo. These tasks can occur in the *LocalizingAndPicking* block. *DemandCheck* block decides whether the replenishment is complete.

3.4 Drone Agent

The drone agent controls the internal behavior of drones. The study uses the ABM approach to model this internal behavior of drones. This agent consists of three states: *Idle*, *NotAligned*, and *Moving* state as shown in Figure 7. The *Idle* state represents the drone not doing any task. The *NotAligned* state represents that the drone is trying to localize itself to the target, to pick cargo, or to place the cargo. The *Moving* state represents the drone traveling to its destination. The *Moving* state mirrors a physical drone's X, Y, and Z positions. For this data, the physical drone sends the 9-axis IMU sensor data to the communication

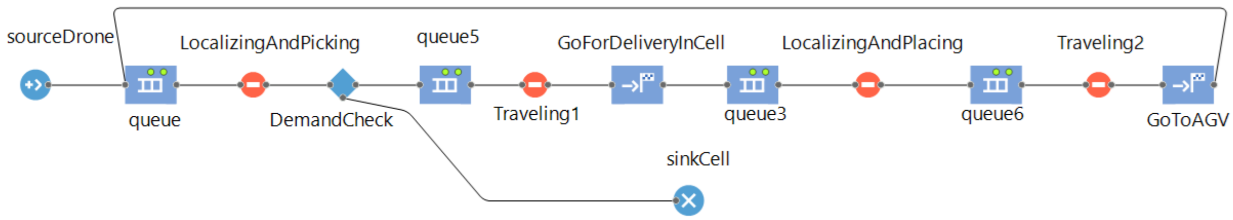

Figure 6: DES logic of cell agent.

agent, which then sends it to the drone agent. Once the drone agent reaches the specified coordinates in the model and gets the confirmation from the communication agent, the state changes to the *Idle* state.

The entities in the cell agent can only move from a queue block to a hold block if the drone agent is in an *Idle* state. The state of the drone agent is controlled by the data received from the communication agent and the blocks of the cell agent. While the entity remains in *LocalizingAndPicking*, and *LocalizingAndPlacing* hold blocks, the drone agent stays in the *NotAligned* state. While the entity remains in *Traveling1* and *Traveling2* hold blocks, the drone agent stays in the *Moving* state.

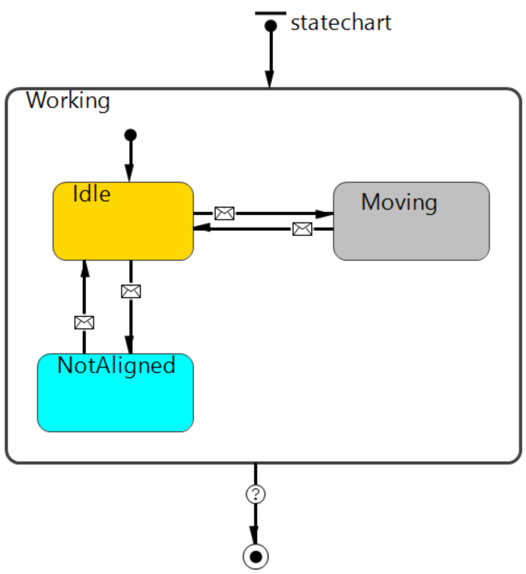

Figure 7: State chart of drone agent.

To verify the model, the study uses Arduino UNO R4 WiFi to test the communication protocol with a simulation model. It is done by checking the state changes in the simulation model when an Arduino sends different messages through different buttons. The study uses LEDS to test the transfer of correct messages from the simulation model to Arduino. The study also uses animation to verify if our model works as expected.

4 DISCUSSION

Once developed, the performance of the AGV-drone system will be evaluated using multiple metrics that reflect both operational and practical value. These include: (1) total material delivery time from home position to the assembly line, (2) drone battery usage per cycle, (3) position and landing accuracy based on deviation from target coordinates, and (4) task execution success rate, representing the reliability of cargo pickup and delivery. These metrics assess the system's responsiveness, precision, and operational efficiency in manufacturing. Tracking these values in both the simulation and physical prototypes enables validation of the DT and supports real-time performance optimization.

In the current design, the drone returns to the AGV after completing a delivery to maintain a compact, coordinated system with minimal infrastructure. This approach allows the AGV to carry and recharge the drone during transit, navigate through no-fly zones, simplify battery management, and reduce the need for additional landing or docking zones within the facility. In this setting, having the drone return to a fixed docking station would require traversing multiple no-fly zones, posing safety and compliance challenges. However, such alternatives may be viable and advantageous in other facility layouts or logistics setups. Depending on the system configuration and design, alternative strategies, such as the drone returning via a different AGV, can be used. These trade-offs are essential when evaluating deployment strategies across diverse manufacturing environments.

We also recognize that the drone's cargo capacity inherently limits the weight and size of materials it can deliver. This system is intended for small, lightweight items such as tools, sensors, documents, or small components required at the point of use. In high-mix, low-volume manufacturing environments, such as electronics or specialty automotive parts, where timely and precise delivery is prioritized over high-volume transport, a small drone is appropriate and feasible. Larger drone use would require facility redesign, which may not be cost-effective. Thus, the AGV-drone system is envisioned as a targeted solution in plant areas where constraints and delivery needs align.

Although the current simulation model primarily facilitates coordination between the AGV and drone, it is developed as the foundation for a fully functional DT system. As the system evolves, the model will support predictive analysis (e.g., estimating delivery delays), testing and optimizing control strategies (e.g., AGV-drone task assignments), and anomaly detection. These advanced capabilities will be enabled once sensor feedback and physical integration are in place, moving the model beyond a passive mirror to an active decision-support system.

5 CONCLUSION AND FUTURE WORKS

This study presents a hybrid simulation model with HIL features for an AGV-drone system that works for intralogistics operations. In other words, the study presents the work-in-progress DT for the AGV-drone system, testing its logic and coordination. In a proposed AGV-drone system, a drone sits on top of an AGV, which navigates through no-fly zones, such as human work areas or unsafe flight regions. When the AGV reaches a designated safe fly zone, the drone takes flight, performs logistics operations, and returns to the AGV. This architecture improves maneuverability and enhances energy efficiency in intralogistics operations.

The main objective of this study is to design a scalable communication system for the AGV-drone system. With this objective in mind and prioritizing safety during the early testing phase, the study develops a simulation model with HIL features as the foundation for a DT. The simulation component of the DT system is modeled using a hybrid approach that combines ABM and DES.

This study represents an initial phase in developing a coordinated AGV-drone system, focusing on concept validation through simulation and early-stage DT modeling. The implications of this study enable the continued development of the AGV-drone system and its DT, ultimately improving future logistics operations. However, the current work does not yet specify all the sensors required for a complete DT system, and it doesn't validate the hybrid model with physical testing. While literature suggests that continuously evolving DT prototypes can improve safety during early development, this study does not verify such improvements.

Future work will include integrating sensor data, physical prototype testing, cost-benefit analysis, and lifecycle-based evaluation to ensure the system's practical applicability and robustness. We also plan to integrate the proposed hybrid simulation model with the real AGV, drone, and manufacturing cell. We will also build the complete DT by continuously evolving the DT prototype with appropriate sensors. The DT model will expand beyond coordination to support real-time performance optimization, predictive analysis, and anomaly detection. We will also test the system with multiple drones and delivery strategies, using advanced coordination algorithms. Furthermore, we will evaluate alternative return strategies, explore drone sizing and layout trade-offs, and assess scalability challenges in more complex environments such as heavy manufacturing. Because of the scalability of the system, the complexity in coordination and concerns about safety arise. This long-term road map aims to ensure that the AGV-drone system remains adaptable, safe, and practically viable in real-world operations.

To reach this ultimate goal, we will continue to apply DT techniques across multiple stages of the system development process, advancing from simulation-only testing to real-time deployment and system-wide optimization.

ACKNOWLEDGMENT

This research was funded by Grant W52PLJ-20–9-3045, awarded to the Interdisciplinary Center of Advanced Manufacturing Systems (ICAMS) by the US Army, from funding from the Industrial Base Analysis & Sustainment Program of the Industrial Base Policy Office of the Secretary of Defense, United States.

REFERENCES

Bányai, T. 2023. "Energy Efficiency of AGV-Drone Joint In-Plant Supply of Production Lines". *Energies* 16(10):4109.

Barbie, A., and W. Hasselbring. 2024. "From Digital Twins to Digital Twin Prototypes: Concepts, Formalization, and Applications". *Institute of Electrical and Electronics Engineers Access* 12:75337–75365.

Brailsford, S. C. 2014. "Modeling Human Behavior- An (ID)entity Crisis?". In *2014 Winter Simulation Conference (WSC)*, 1539–1548 https://doi.org/10.1109/WSC.2014.7020006.

Caro, J. J., and J. Möller. 2016. "Advantages and Disadvantages of Discrete-Event Simulation for Health Economic Analyses". *Expert Review of Pharmacoeconomics & Outcomes Research* 16(3):327–329.

Chandrasekaran, M., and P. Rajesh. 2025. "Conceptualisation of Digital Twin-Enabled New Product Design Framework for the Automotive Industry". *Journal of Engineering Design*:1–41.

de Miguel Molina, M., V. Santamarina Campos, M. Á. Carabal Montagud, and B. de Miguel Molina. 2018. "Ethics for Civil Indoor Drones: A Qualitative Analysis". *International Journal of Micro Air Vehicles* 10(4):340–351.

Dubiel, B., and O. Tsimhoni. 2005. "Integrating Agent Based Modeling into a Discrete Event Simulation". In *2005 Winter Simulation Conference (WSC)*, 1029–1037 https://doi.org/10.1109/WSC.2005.1574355.

Emmert-Streib, F. 2023. "Defining a Digital Twin: A Data Science-Based Unification". *Machine Learning and Knowledge Extraction* 5(3):1036–1054.

Fathy, H. K., Z. S. Filipi, J. Hagena, and J. L. Stein. 2006. "Review of Hardware-in-the-Loop Simulation and Its Prospects in the Automotive Area". In *Modeling and Simulation for Military Applications*, edited by W. K. Schum and A. F. Sisti, Volume 6228, 117–136. Kissimmee: Society of Photo-Optical Instrumentation Engineers.

Fehling, C., and A. Saraceni. 2022. "Feasibility of Drones & AGVs in the Last Mile Delivery: Lessons from Germany". *SSRN preprint SSRN: 4065011*.

Grieves, M., and J. Vickers. 2017. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems". *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*:85–113.

Grigoropoulos, N., and S. Lalis. 2020. "Simulation and Digital Twin Support for Managed Drone Applications". In *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, September 14^{th}–16^{th}, Prague, Czech Republic, 1–8.

Hell, P. M., and P. J. Varga. 2019. "Drone Systems for Factory Security and Surveillance". *Interdisciplinary Description of Complex Systems (INDECS)* 17(3-A):458–467.

Huang, S., G. Wang, D. Lei, and Y. Yan. 2022. "Toward Digital Validation for Rapid Product Development Based on Digital Twin: A Framework". *The International Journal of Advanced Manufacturing Technology*:1–15.

Jiang, Y., S. Yin, K. Li, H. Luo, and O. Kaynak. 2021. "Industrial Applications of Digital Twins". *Philosophical Transactions of the Royal Society A* 379(2207):20200360.

Kapteyn, M. G., J. V. Pretorius, and K. E. Willcox. 2021. "A Probabilistic Graphical Model Foundation for Enabling Predictive Digital Twins at Scale". *Nature Computational Science* 1(5):337–347.

Katsigiannis, M., and K. Mykoniatis. 2024. "Real-Time Tracking of Production in Assembly Operations Using Agent-Based Modeling and Digital Twin Techniques". In *2024 Winter Simulation Conference (WSC)*, 3011–3021 https://doi.org/10.1109/WSC63780.2024.10838980.

Khazetdinov, A., A. Zakiev, T. Tsoy, M. Svinin, and E. Magid. 2021. "Embedded ArUco: A Novel Approach for High Precision UAV Landing". In *2021 International Siberian Conference on Control and Communications (SIBCON)*, May 13th–15th, Kazan, Russia, 1–6.

Law, A. M., and W. D. Kelton. 2015. *Simulation Modeling and Analysis*. 5 ed. New York: McGraw-Hill Education.

Li, Y., M. Zhao, J. Sugihara, and T. Nishio. 2024. "Cooperative Navigation System of AGV and UAV with Autonomous and Precise Landing". In *2024 IEEE International Conference on Mechatronics and Automation (ICMA)*, August 4th–7th, Tianjin, China, 1477–1483.

Lo, C., C.-H. Chen, and R. Y. Zhong. 2021. "A Review of Digital Twin in Product Design and Development". *Advanced Engineering Informatics* 48:101297.

Macal, C., and M. North. 2014. "Introductory Tutorial: Agent-Based Modeling and Simulation". In *2014 Winter Simulation Conference (WSC)*, 6–20 https://doi.org/10.1109/WSC.2014.7019874.

Martínez-Gutiérrez, A., J. Díez-González, R. Ferrero-Guillén, P. Verde, R. Álvarez, and H. Perez. 2021. "Digital Twin for Automatic Transportation in Industry 4.0". *Sensors* 21(10):3344.

Munasinghe, I., A. Perera, and R. C. Deo. 2024. "A Comprehensive Review of UAV-UGV Collaboration: Advancements and Challenges". *Journal of Sensor and Actuator Networks* 13(6):81.

Mykoniatis, K., and A. Angelopoulou. 2020. "A Modeling Framework for the Application of Multi-Paradigm Simulation Methods". *Simulation* 96(1):55–73.

Mykoniatis, K., and G. A. Harris. 2021. "A Digital Twin Emulator of a Modular Production System Using a Data-Driven Hybrid Modeling and Simulation Approach". *Journal of Intelligent Manufacturing* 32(7):1899–1911.

Pantelidakis, M., K. Mykoniatis, J. Liu, and G. Harris. 2022. "A Digital Twin Ecosystem for Additive Manufacturing Using a Real-Time Development Platform". *The International Journal of Advanced Manufacturing Technology* 120(9):6547–6563.

Postel, Jon 1980. "User Datagram Protocol". RFC 768, Internet Engineering Task Force.

Radanliev, P., D. De Roure, R. Nicolescu, M. Huth, and O. Santos. 2022. "Digital Twins: Artificial Intelligence and the IoT Cyber-Physical Systems in Industry 4.0". *International Journal of Intelligent Robotics and Applications* 6(1):171–185.

Shrestha, R. B., E. Hungerford, and K. Mykoniatis. 2024. "Planning a Material Replenishment Through Autonomous Mobile Robot in an Assembly Plant Using a Hybrid Simulation Approach". In *2024 Winter Simulation Conference (WSC)*, 1197–1208 https://doi.org/10.1109/WSC63780.2024.10838769.

Trauer, J., S. Schweigert-Recksiek, C. Engel, K. Spreitzer, and M. Zimmermann. 2020. "What is a Digital Twin?–Definitions and Insights from an Industrial Case Study in Technical Product Development". In *Proceedings of the Design Society: DESIGN Conference*, edited by D. Marjanović, M. Štorga, S. Škec, and T. Martinec, Volume 1, 757–766. Cambridge: Cambridge University Press.

Vis, I. F. 2006. "Survey of Research in the Design and Control of Automated Guided Vehicle Systems". *European Journal of Operational Research* 170(3):677–709.

Yao, J.-F., Y. Yang, X.-C. Wang, and X.-P. Zhang. 2023. "Systematic Review of Digital Twin Technology and Applications". *Visual Computing for Industry, Biomedicine, and Art* 6(1):10.

Yeh, A., P. Ratsamee, K. Kiyokawa, Y. Uranishi, T. Mashita, H. Takemura, *et al*. 2017. "Exploring Proxemics for Human-Drone Interaction". In *Proceedings of the 5th International Conference on Human Agent Interaction*, October 17th–20th, Bielefeld, Germany, 81–88.

AUTHOR BIOGRAPHIES

RUPESH BADE SHRESTHA is a PhD student in the Department of Industrial and Systems Engineering at Auburn University. He holds a Bachelor of Engineering in Industrial Engineering from the Institute of Engineering, Thapathali Campus, Tribhuvan University. He is a Research Assistant at the Interdisciplinary Center for Advanced Manufacturing Systems (ICAMS) and a Teaching Assistant for the Simulation course. His research interests are Modeling and Simulation, DTs, Model-Based Systems Engineering, Robotics, Industry 4.0, and Lean Systems. His email address is rzb0117@auburn.edu.

KONSTANTINOS MYKONIATIS is an Assistant Professor in the Industrial and Systems Engineering department at Auburn University. He holds a B.Eng. in Production Engineering and Management from the Technical University of Crete and an M.S. and Ph.D. degrees in Modeling and Simulation from the University of Central Florida (UCF). Prior to joining Auburn University, he worked as an Assistant Professor at Wichita State University and as a simulation instructor at the Industrial Engineering and Management Systems at UCF. In addition to his academic positions, he has held professional engineering positions as a modeling and simulation professional in the Business Analytics and Industrial Engineering Division of Universal Orlando Parks and Resorts, the Institute for Simulation and Training, the Institute for Advanced Systems Engineering, and the Public Power Corporation S.A. Hellas. His primary research interests are in the areas of hybrid modeling and simulation, advanced manufacturing systems, robotics, and automation. His email address is kmykoniatis@auburn.edu.