# Abstract

Industry 4.0 is the result of rapid technological advancement, dictated by Moore’s Law. Industry 4.0 seeks to enhance on Industry 3.0's automation by allowing devices to interact with one another, commonly known as the Internet of Things (IoT). Due to the widespread growth of IoT, various data from sensors is made accessible as what’s known as big data. Coupled with improvements from deep learning and cloud computing, these data can be stored and processed in the cloud or used for training machine models to make decisions. Resulting in an intelligent system that makes decisions without human involvement. Industry 4.0 has created an opportunity for a future where smart factories can leverage some of the most cutting-edge developing technology to automate and enhance many processes. Unmanned Aerial Vehicles (UAVs) is one such example. In Industry 4.0, UAVs have been deployed to perform task in smart factories that performs automatable and tedious tasks. Hence, this report aims to covers the usage and capabilities of UAVs in Industry 4.0. More specifically, the development and design of an intelligent UAV system for Industry 4.0.

# UAVs in industry 4.0

In Industry 4.0, autonomous UAVs are used to achieve a wide range of missions. Missions include warehouse operations – Inventory management, indoor intra-logistics, and inspections and surveillance[1]. Manufacturing – Inspection and maintenance [2]. For Warehouse management, UAVs are equipped with RFID scanners and Cameras for QR code and deployed to perform stock taking. Additionally, they have capabilities to transfer inventory from one location to another and lastly, check for pallet placement and detect theft. For Manufacturing, UAVs carry out inspection on equipment using infrared sensors to detect anomalies and cameras to deploy computer vision to detect cracks.

# Classification of UAVs

Generally, there are 4 categories of UAVs. Multi-rotor drones consist of a flight controller and 3 to 8 propellers. The flight controller collects data from the inertia measurement unit (IMU) and performs sensor fusion to accurately collect orientation and heading data. By applying a field of mathematics, control theory, the flight dynamics can be modelled for stable flight. Due to the lower cost and better maneuverability compared to fixed winged UAVs, multi-rotor drones are commonly used in smart factories. Figure 3A shows the categories of drones.

zDiagram

Description automatically generated

Figure 3A: Categories of UAVs [3]

Additionally, the level of aerial autonomy can be further classified based on inputs, capabilities reactions and decisions. Currently, drones have reached Level 4A of autonomy, where is senses and navigates with the assistance of an external computer. Figure 3B shows the categories aerial autonomy.

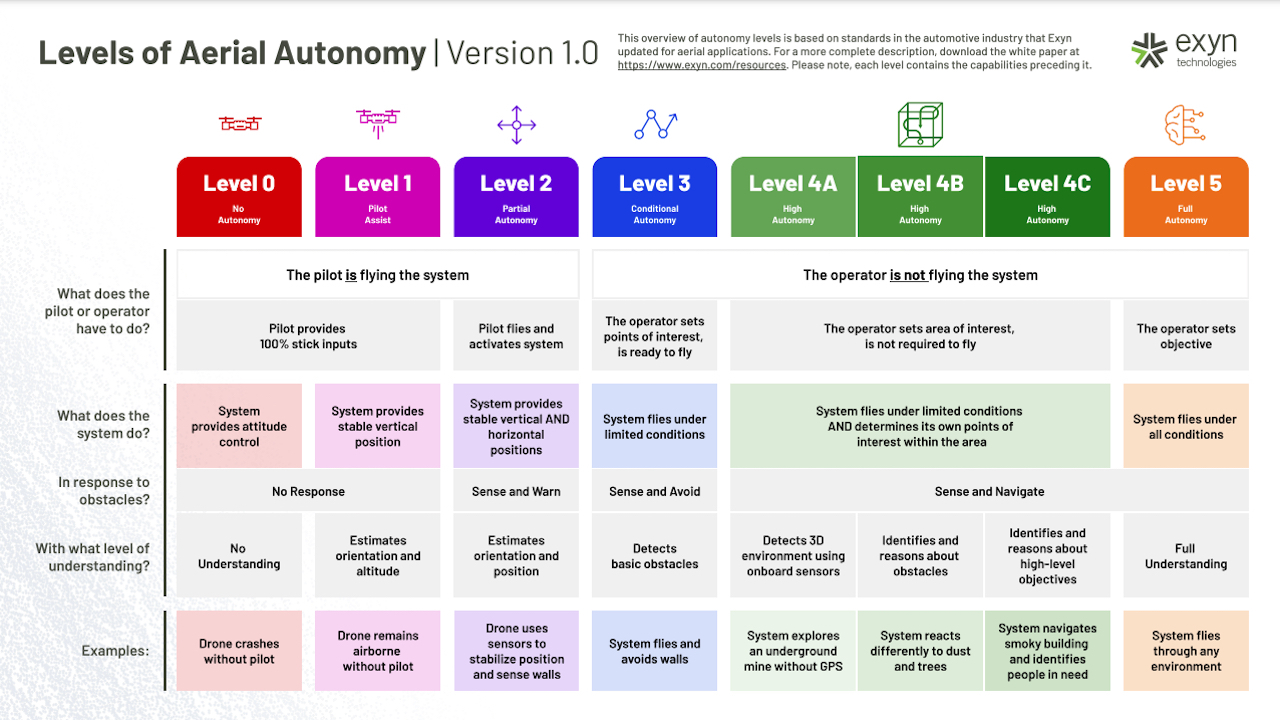


Figure 3B: Levels of autonomous UAV[4]

In this report, more emphasis will be placed on the development of Level 4A UAVs in smart factories.

# Function principles of UAVs

In smart factories, drones are IoT devices that communicate together and by transmitting information, it results in a collective perception. Although mapping may be completed, the environment may not be deterministic due to human interaction in the system. Hence, UAVs operate in a Partially Observable, Stochastic, Collaborative, Multi-agent, Dynamic, and Continuous environment. Thus, smart factories UAVs must be goal based, utility and learning agents. Agents are things that senses and act on the environment. Goal based agents have sensors to localize their position and aims to reduce their position to their goal. Utility agents aims to minimize the total costs which could be path time or power consumption, or various parameters based on pareto optimal points lastly, by learning from the wide range of data, the agent could increase performance over time. By having sensors to detect objects and indoor localization or stereo cameras or Light detection and ranging (LIDAR) devices to perform simultaneous localization and mapping (SLAM), the UAV should be able to run search algorithms such as A\* to reach the end goal based on current state, as well as update its current state based on its perception from sensors. Figure 4A shows how SLAM is implemented. Figure 4B shows a graphical representation of A\* algorithm.

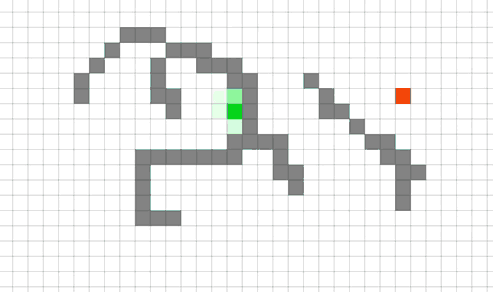
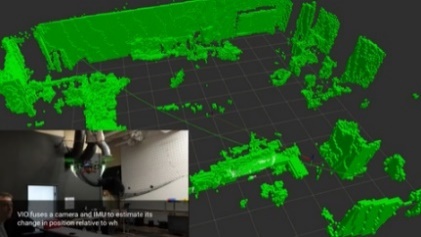


Figure 4A: Point cloud reconstruction of SLAM [5] Figure 4B: Graphical representation of A\* Algo[6]

# Infinium Scan

Infinium Robotics is a company that develops collaborative drones for stock taking in smart factories [7]. The system consists of a camera and RFID reader equipped UAV, an unmanned ground vehicle (UGV) and a ground control system (GCS). UAVs are tethered to the UGV using a cable that transmit power and data where the tethering connection acts like a prismatic joint (1DOF).



Figure 4D: Infinium Scan Drone[8]

Firstly, mapping is done to collect obstacle data and free space. By modelling the dynamics of the UAV/UGV, a configuration space is created. Figure 4D shows a 3D plot of configuration space based on a 2D, rotating robot.

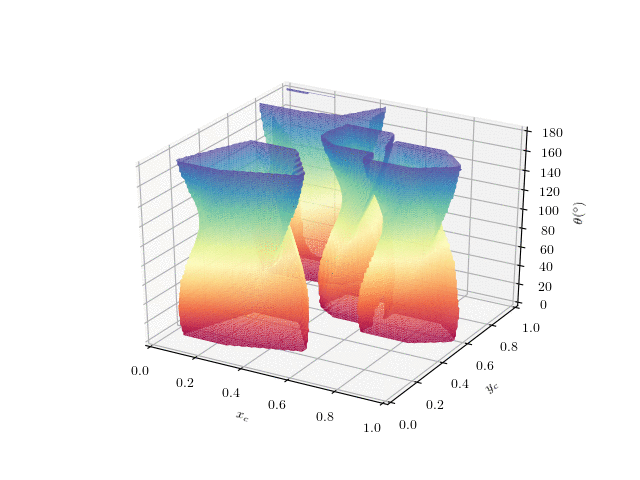


Figure 4D: Visualisation of configuration space on a 2D rotating robot [9]

With this data, a ground control system (GCS) completes a search algorithm of minimum path cost for the UAVs to possibly scan the items for stock taking. This is done by calculation of minimum total path costs of UGVs while preventing collision. To calculate the height of the UAVs, inverse kinematics (IK) based on the final position of the drone is used to calculate the required elevation from the UGVs. The waypoints are then transmitted over to UGVs. Localization is done on the UGVs, and by travelling to the selected waypoints, the position of the UAVs may be maintained by relative position sensor. This allows the UAVs to reach their final waypoint to complete scanning. Finally, the scanned data is uploaded to the warehouse management system (WMS).

# Challenges with UAVs

In smart factories, drones compete with existing solutions like cranes, conveyors, and robotic arms. These solutions are mounted and positioned to handle tasks in an assembly line. By having a fixed location, the end effectors of the robotic arms can be calculated through IK to get the relevant joint angle to reach the intended position and orientation. While drones can reach desired orientation and position, there is a much wider positional tolerance compared to robotic arms due to disturbances from turbulences from other drones or poorer data from localization techniques [10]. Therefore, tightening of a screw will take longer time compared to a robotic arm. Additionally, safety and noise may be a concern. Doors, people, and equipment limit the movements of UAVs. Lastly, the power consumption of keeping a drone airborne with payload is higher compared to robotic arms. These factors result in robotic arms being the primary factor of manufacturing.

# Proposal for UAVs in smart manufacturing

Although UAVs are harder to implement compared to robotic arms, there are still inherent advantages of drones over robotic arms. Firstly, drones can be cheaper than robotic arms. Secondly and most importantly, there is a law of diminishing return for increasing the range of robotic arms. Robotic arms have limited range length due to limited link length and motor sizing. As the length of the arm increases, the cantilever system requires a thicker base and larger motor. Whereas drone range are limited by their localization capabilities and flight time due to batteries.

With these advantages, the proposed UAV system would be implemented in a smart factory for stock taking, labelling and material handling. This would bridge the gap of material transport and production lines by solve labor issues where workers must reach higher areas in warehouses to collect and transport the needed inventory.

# System description

Like Infinium Scan, the system would include a GCS to communicate with UAVs. The mission of the UAVs would be stock taking, labelling, storage, and delivery of inventory. The UAVs will be equipped with flight controller with a companion computer like Jetson AGX Xavier. The flight controller would include an IMU which has gyroscope, magnetometer, accelerometer and barometer. Coupled with an indoor GPS, extended Kalman Filter (EKF) can be applied for sensor fusion to get better positional data. The companion computer would have LIDARs, cameras, RFID scanners, and a robotic arm/gripper attached to it. The LIDAR would be used for SLAM and the camera for object detection and recognition. The arm connecting to gripper could have 1 DOF to retrieve and deliver objects. In addition to further simplify the problem for computing, human intervention is not allowed. This results in a fully observable, and deterministic environment.

A picture containing indoor

Description automatically generated

Figure 8A: Gripper connected to UAV

Inputs to the GCS will be from the WMS where commands are given to transport an inventory. The GCS will assign a drone for the task and create paths in the form of waypoints. To pick and grab items, the drone reaches the final waypoint, detects, and performs classification to search for the object. Next, hand-eye coordination for robotics is implemented using stereoscopic vision to minimize the distance between the gripper and inventory. After picking the package, the GCS calculates the waypoints to deliver the object and the process is repeated.

In summary, the UAV mission includes mapping, multi-agent path finding (MAPF), object detection and classification, inventory handling and labeling.

The knowledge of the UAV should include:

1. How to navigate the warehouse
2. How to handle collisions
3. How to optimally plan paths for fleet of UAVs
4. How to identify inventories and weight
5. How to correctly grab and retrieve the objects
6. How to label the inventory

The rules and facts include:

1. No two drones can be at the same position at any time (collision)

The other rules and facts are listed in the pseudo code.

Table

Description automatically generated

# Learning approach

Suitable tasks for learning approach includes MAPF, object detection and classification, inventory handling and labeling.

For MAPF, it is a NP-hard to obtain an optimal solution [11]. However, it may be possible to implement Genetic Algorithm (GA). In addition, we could apply Prioritized Planning [12] and Improved Conflict Based Search (ICBS) [13] This would allow us to solve the multi-objective problem. Prioritized planning implements Cooperative A\*[14], and priorities agents with largest estimated costs to plan first. This results in fewer obstacles due to collisions for the toughest paths, which can improve total computational costs.

# References

[1] L. Wawrla, O. Maghazei, and T. Netland, “Whitepaper-Applications of drones in warehouse operations Whitepaper Applications of drones in warehouse operations,” 2019, Accessed: Feb. 20, 2022. [Online]. Available: www.pom.ethz.ch

[2] M. Javaid, I. H. Khan, R. P. Singh, S. Rab, and R. Suman, “Exploring contributions of drones towards Industry 4.0”, doi: 10.1108/IR-09-2021-0203.

[3] M. Javaid, I. H. Khan, R. P. Singh, S. Rab, and R. Suman, “Exploring contributions of drones towards Industry 4.0,” *Industrial Robot*, 2021, doi: 10.1108/IR-09-2021-0203/FULL/PDF.

[4] “Exyn Drones Achieve Autonomy Level 4.” https://www.exyn.com/news/exyn-drones-achieve-autonomy-level-4 (accessed Feb. 20, 2022).

[5] “SLAM: The L is for Localization – the M is for Mapping | ModalAI, Inc.” https://www.modalai.com/blogs/blog/slam-the-l-is-for-localization-the-m-is-for-mapping (accessed Feb. 20, 2022).

[6] “ysshah95/Astar-Algorithm-for-ACME-Robotics.” https://github.com/ysshah95/Astar-Algorithm-for-ACME-Robotics (accessed Feb. 20, 2022).

[7] “Indoor Unmanned Aerial Systems | Infinium Scan.” https://infiniumrobotics.com/infinium-scan/ (accessed Feb. 20, 2022).

[8] “Dexion Infinium Scan Drone - Dexion.” https://www.dexion.biz/product/scan-drones/ (accessed Feb. 20, 2022).

[9] “ethz-asl/amr\_visualisations: Simple python animated examples for the planning segment of the Autonomous Mobile Robots lecture slides.” https://github.com/ethz-asl/amr\_visualisations (accessed Feb. 20, 2022).

[10] O. Maghazei and T. Netland, “Drones in manufacturing: exploring opportunities for research and practice,” *Journal of Manufacturing Technology Management*, vol. 31, no. 6, pp. 1237–1259, Nov. 2020, doi: 10.1108/JMTM-03-2019-0099/FULL/PDF.

[11] H. Ma, G. Wagner, A. Felner, J. Li, T. K. Satish Kumar, and S. Koenig, “Multi-Agent Path Finding with Deadlines \*”.

[12] M. Cap, P. Novak, A. Kleiner, and M. Selecky, “Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 835–849, Jul. 2015, doi: 10.1109/TASE.2015.2445780.

[13] J. Li, A. Felner, E. Boyarski, H. Ma, and S. Koenig, “Improved Heuristics for Multi-Agent Path Finding with Conflict-Based Search,” 2019.

[14] “Cooperative Pathfinding | Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment.” https://ojs.aaai.org/index.php/AIIDE/article/view/18726 (accessed Feb. 24, 2022).

**Function** DRONE-SELECTOR([*weight*]) **returns** a drone model

**if** *weight* = *Heavy* **then return** *DroneA*

**else if** *weight = Light* **then return** *DroneB*

**else return** *DroneC*

**Function** DRONE-MOVE-GRIP([*location,status*]) **returns** a Grab, Drop or Reposition

**if** *position* = *WaypointRadius*

**if** *status = InventoryPresent* **then return** *Grab*

**else if** *status = NoInventory* **then return** *Release*

**else return** *Reposition*

**Function** DRONE-UPDATE-SPACE([*droneID,status*]) **returns** a Grab, Drop or Reposition

***for*** *each droneID*

**if** *status* = *InventoryPresent* **then return** *IncreaseConfigurationSpace*

**else if** *status = NoInventory* **then return** *DecreaseConfigurationSpace*

**Function** DRONE-LABEL([*location,status*]) **returns** a Label or Reposition

**if** *position* = *WaypointRadius*

**if** *status = LabelRequired* **then return** *Label*

**else return** *Reposition*

**Function** CHECK-BATTERY([*droneID,batterystatus*]) **returns** a State or Reposition

***for*** *each droneID*

**if** *batterystatus* = *low* **then return** *Charge*

**else if** *batterystatus* *= full* **then return** *ChargeComplete*

Algorithm: Proposed GA method.

***while*** *systemrun = True* ***do***

**if**