2022 International Conference on Unmanned Aircraft Systems (ICUAS)
Dubrovnik, Croatia. June 21-24, 2022

Smart Anomaly Detection and Monitoring of Industry 4.0 by Drones

William Pensec[1], David Espes[2] and Catherine Dezan[2]

2022 International Conference on Unmanned Aircraft Systems (ICUAS) | 978-1-6654-0593-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICUAS54217.2022.9836057

***Abstract*— Nowadays, industry 4.0 can be distributed over a large area. To monitor their processes, they use sensors that periodically gather data on the system. Based on them, operators can detect that an anomaly occurs on the system. However, it is not always easy to know the causes of the anomaly because the operator has no visual information on the system. To help operators to identify the root of the anomaly, drones are very useful because they are fast enough to intervene in large-scale industry and embed a large variety of sensors to offer complementary data (images, video...) that are necessary for the diagnosis. However, drones have to be synchronized with the industrial process to know where the anomaly occurs and to go there in an automated way. We propose a new architecture to automate the displacement of the drone to reach safely the place where the anomaly is located and to confirm it using a deep-learning approach. The drone embeds a small computing system (Raspberry Pi) which communicates with the supervisory control and data acquisition system in order to be aware of anomalies that occur on the industrial process. To function properly indoor or outdoor, the drone is positioned either using a precise positioning system based on ultra-wide band (UWB) or on the GPS. The drone can take pictures of the potentially detected anomaly and thanks to a neural network algorithm, it analyzes the images to confirm or deny the anomaly. The results show an error on the indoor position of about 5 cm, and a precision of about 90% to detect anomalies.**

I. INTRODUCTION

Today, in a context of industrialization and digitization of production, it is important to have automated responses in order to gain flexibility and competitiveness. This type of automation relies on very heterogeneous systems and equipment, whether in terms of the systems or network protocols used and the information to be analyzed. These infrastructures are based on cyber-physical systems. Although these infrastructures ensure a very precise traceability of the production chain, the information collected may not be sufficient to identify a potential problem and its origin. Thus, because of these needs for industry in terms of monitoring, detection and prevention, the term Industry 4.0 was born. This concept brings the notion of Internet of Things for Industries (IIoT) [1], [2] and therefore a need for connection between the different services, production lines, operators, maintenance technicians in order to be the most capable of preventing and repairing breakdowns or potential breakdowns that may occur and that could stop production or represent problems for the personnel.

[1]Univ. Bretagne Sud / Lab-STICC
`william.pensec@univ-ubs.fr`
[2]Univ. Bretagne Occidentale / Lab-STICC
`david.espes@univ-brest.fr`,
`catherine.dezan@univ-brest.fr`

The fields of application of drones can be very varied and can be complementary to other existing recognition systems. For example, drones can be used for infrastructure monitoring or for comprehensive diagnostics that can be performed remotely by operators in the event of a breakdown. Since drones are equipped with cameras and can move freely [3], without interfering with the manufacturing process, they are indispensable assets for taking pictures of the failure and transmitting them to operators for processing, analysis and recovery of the failure. But they can also process them themselves or send them to the cloud for analysis and processing [1]. When the drone moves to the breakdown, it must be able to locate itself precisely in order to get there as quickly as possible while avoiding physical harm to personnel on the way [4], [5]. To do this, the drone must be synchronized with the production line and use failure detection algorithms that would be implemented directly into the system to warn it of an immediate failure or a potential threat.

With the aim of Industry 4.0 [1], [2], the drones will be more and more used. Industrial use of drones come in at around 25% while the production is only at 1.5% of the total production. They offer many possibilities across many areas for the industry. They can be used for area surveillance [6], construction inspection, agricultural monitoring missions... They are generally used outdoors to make the best use of their potential (speed, altitude, sensors). However, they can be adapted for indoor use. In this case, indoor environments present different constraints such as floors, moving or stationary objects. It is necessary that the drone can fly in this environment while avoiding these obstacles (people, objects, machines) [3] although the constraints of buildings make that GPS signals do not pass through the walls [4]. It is therefore necessary to position them differently like Bluetooth Low Energy devices (BLE) [7], ZigBee protocol [8] or with ultrasound [9].

To analyze, in real time, the production process in order to detect or prevent anomalies without impacting the staff safety or introducing heavy devices in the industrial part, cooperation between the devices present in the environment, such as the production line (Industrial Control System (ICS)), the automatons (Programmable Logic Controller (PLC)) and the unmanned aircraft vehicle (UAV) is required. This cooperation makes it possible to ensure various operations such as taking images of a failure in the system or monitoring the environment in order to avoid risks for the personnel or the material as well as analysing these data to find the causes of anomalies.

To make all of these devices cooperate, a new architecture is needed. Our architecture sets up a communication between the ICS and the drones that monitor the environment. This communication between the devices allows the drone to know when and where the anomaly occurs. To properly function indoor or outdoor, the drone cannot only rely on the GPS signals. In indoor, due to building walls, GPS signals cannot be received. So, UWB sensors are added to the existing industrial system to have a precise indoor positioning system. If the UAV is in range of the UWB positioning system, the UAV uses it to position itself due to its very precise accuracy. After reaching its destination, the drone can take one or more images of the anomaly. These images are processed using an embedded neural network algorithm to confirm or deny the anomaly and treat it accordingly.

First, Section II introduces the related research works. Then, Section III presents the environment in which the drone operates. In Section IV, our new architecture is introduced and allows synchronizing the drones with other devices already present in an ICS. Section V presents the results on the displacement of the drone and its ability to safely operate in the system and to precisely confirm anomalies that occur on it. Finally, Section VI concludes this work.

II. RELATED WORKS

Drones can have many uses in industry, including the monitoring of an area, the supervision of industrial processes...

Most of these missions were usually performed by remotely controlled ground robots and more recently by unmanned ground vehicle (UGV). The tasks that they have to achieve are repetitive tasks in general such as painting, welding, assembling pieces e.g., in automotive industry where they will replace workers to improve the benefits of the industry and reduce hardship at work. They can carry or lift heavy loads without presenting any risk to the personnel. They can also check the quality of the manufactured products via the analysis of photos or video in continuous flow or be used in risky environments (nuclear power plant, highly corrosive environments, ...).

Unfortunately, these UGV cannot cover all the current use cases in these industrial environments. This is why the use of UAVs is growing more and more. UAVs are flexible enough to accomplish many tasks, including tasks that are impossible for UGVs. They can be used in swarms and communicate with each other to meet specific needs such as space surveillance or fault detection [10]. They can also be connected to Intrusion Detection Systems (IDS). These IDS allow to detect malicious intrusions in the cyber-physical systems of the industrial environment [11].

As aforementioned, drones can be used in production lines for surveillance, inspection [12] but also to prevent or detect anomalies on this production line and collect data [1], [13], [14]. They embed many sensors to meet the specific requirements of the mission given to them. They also have the possibility to communicate in a swarm composed of several drones. All these features prove to be very effective to meet the needs of operators to do maintenance in an automated way and allow a reliable and fast surveillance of zone without danger for the personnel. unlike UGV, aerial drones have the ability to move freely over machines in the environment. For this, they must move and be able to locate themselves precisely at any time in order to avoid collisions but also to go to a given point quickly and efficiently. In [15], the authors present concepts for the movement and control of UAVs in indoor surveillance. UAVs can be used for agricultural surveillance to reduce the workload of farmers. They perform well because they can move freely and analyze field data via sensors. For example, as shown in [16], it is possible to use drones to analyze fields and treat them with the right amount of pesticide at the right time to avoid inconvenience and health problems. In [12], multiple drones are used to move loads in a synchronized manner. To work properly, it is highly necessary that the position of the drones is exact and that there are no errors on the movement to avoid unbalancing the load.

A camera can be embedded on the drone so that it can take a picture of an event and send it to the cloud [1], [14], a server or even process it itself in order to analyze what is present on the image and detect potential anomalies. It can then act or transmit the processed information to maintenance operators within the industry. There are many possible applications for the use of a drone with a camera. Indeed, they can monitor an area [17], allow navigation without GPS, traffic monitoring, monitoring of the electrical network by image analysis [18] and research and rescue human in hazardous area e.g., due to a natural disaster.

In [19], the authors propose an indoor navigation system, without GPS, using a system based on image capture.

In [3], a solution to automate indoor flight and prevent collision with obstacles is introduced. An IR sensor is used to detect obstacles and to allow the mapping of the environment. This mapping brings teaching opportunities to the drone so that it learns itself to find its way in a new environment and that it automatically detects obstacles of all kinds.

Drones with cameras can also be used for mapping, obstacle avoidance, route planning, all this is summarized in [20].

Finally, drones can take adequate decisions using neural networks in order to analyze the information and act accordingly. In [1], the authors introduce a model designed to monitor the Industrial Internet of Things using an UAV. Its mission is to monitor the production line of a concrete plant. The drone analyzes the collected data using deep learning.

At the best of our knowledge, there are not any papers discussing the cooperation between UAVs and Supervisory Control and Data Acquisition (SCADA) or PLC equipment. [21] refers to the application of UAVs with PLCs and makes an analysis of the different parameters to guarantee for this application e.g., the precision of the flight control, the duration of the loss for the flight control... Our work is the first to introduce an architecture that allows the cooperation between UAVs and the ICS system in order to confirm the causes of an anomaly when one occurs.

III. BACKGROUND

In this part, the environment in which the drone operates is presented. This environment is divided into 3 parts: the industrial platform, the drone and the SCADA server.

A. Industrial platform

The platform is a replica of a pharmaceutical industry complex. This replica is built by Schneider Electric. It is composed of two automatons: M340 and M580.

The first module manages the orders (recipes) dictated by the user. Once an order has been placed, an articulated arm will come and pick up a bottle and a cap and place them on a pallet. Then, this pallet will pass on a conveyor and go to the second module. When the pallet comes back from the second module, the articulated arm takes the stoppered bottle and puts it in a box. Once the box is full, a conveyor takes the box to another location for storage.

The second module takes care of the order processing. The order is processed by RFID chips that are located throughout the production line. The order consists of putting beads in the bottle. The amount of beads to be put in depends on the order.

B. Drone

The drone used in this paper is the Matrice 100 (M100) of the brand DJI. This drone is optimal for the world of research because it has many sensors, a good autopilot, many tutorials and also the possibility of embarking more materials such as a camera, batteries, an embedded computer, etc.

The company DJI provides tools like Software Development Kit (SDK) to access the sensors, autopilot and other features of their drones. These tools can be applied to create an Android/iOS application to manipulate the drone, manage the sensors/devices added to the drone, or to connect one's own embedded computer to the drone via the serial port (universal asynchronous receiver-transmitter (UART)).

We use the tool provided by DJI called Onboard-SDK. This tool allows to connect an embedded computer on the drone and to access to the sensors, to the autopilot and to manipulate the drone remotely in an automatic and programmable way directly from the computer in C++. There are many examples on the manufacturer's website to make this SDK work on the drone of our choice. The choice of the SDK version is determined by the drone. For Matrice 100 UAVs, the last compatible SDK version is the 3.9. Same if the following versions can be compatible too, they are not indicated as such on the manufacturer's website. In order to anticipate any compatibility problem, the version recommended by the manufacturer is used.

Even if the platform we use is static, we use a drone in order to be easier to set up given its autonomy to respond to the tasks requested and thus be more reliable in its work than multiple cameras installed for surveillance. The main goal of this project is to be able to make a continuous analysis in an industry and thus on surfaces which can be very large and thus where it can be difficult to install multiple cameras working together.

C. SCADA server

The SCADA server is quite common and gathers data from all the sensors that are present in the industrial platform or acts on the industrial platform to adapt the process.

The SCADA server is used as an intermediary between the UAV and the industrial platform. The SCADA server communicates with the UAV and informs it when an anomaly occurs. Once the UAV reaches the area where the anomaly occurred, it sends back complementary information (images, diagnosis...) to the SCADA server in order to be analyzed by the operators.

IV. METHODOLOGY

Figure 1 shows the general architecture of our approach. This architecture is composed of 3 distinct parts.

The first part is the industrial platform which is composed of 2 automatons. This industrial platform was built by Schneider. The second part consists of: 1) a SCADA server that periodically gathers data from the sensors connected to the two automatons and stores them in a database and 2) a java application that periodically looks at the database to detect anomalies and pushes them to the drone. The third and last part includes two components. First, the drone that embeds a Raspberry Pi which allows to pilot the whole drone and to link the drone to the java application. Second, the drone embeds a UWB tag that is connected to the Raspberry Pi. This tags allows the drone to be positioned indoor when it is in the range of four UWB anchors.

Thus, the problem that arises is the management of the communication between these 3 parts and their coordination. The goal of our approach is to move the drone over the anomaly that is detected by the java application from the reading of the sensors.

A. Anomaly management

Industrial processes may encounter anomalies during their functioning. The most common anomalies in industrial processes have 13 causes [22].

Using an UAV can confirm 12 of them by:

- Checking that the order of the segments is respected
- Checking that the production order (PO) is executed with the right equipment
- Checking that the PO is executed with the correct material
- Checking that the total expected time is correct
- Checking the duration per segment
- Checking times between segments is correct
- Checking if requests arrive while the PO is not launched
- Checking that the quantity requested is the one manufactured
- Checking if resources are available before launching the PO

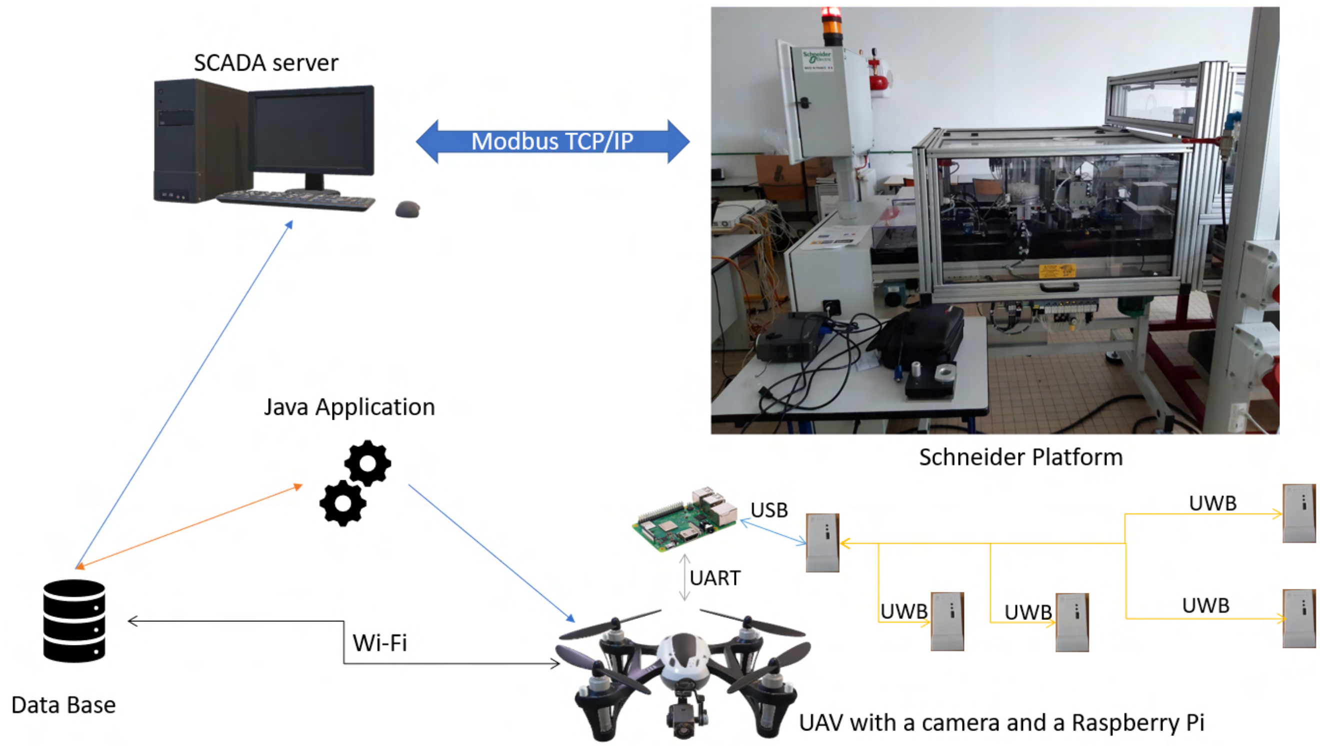

Fig. 1. General architecture

- Checking if equipment is broken down, while sending data
- Checking the launching order of PO
- Checking if equipment is down while continuing to send data

To confirm the cause of an anomaly, it is: 1) necessary to get fine-grained data from the industrial process to know precisely where the anomaly occurs in the process and 2) compare these data to the production order to detect any deviation during the process.

To get fine-grained data, the SCADA server periodically asks for information to the automatons present in the industrial process. These information are stocked in registers by the automatons. These registers can be read as registers "%M" and "%MW". The registers "%M" represent a bit and the registers "%MW" represent an array of bits. The size of these arrays is defined at the design time.

These information concerns all the characteristics of the process that can help to determine the cause of an anomaly. They can be timing information (e.g., time between two segments, duration of an operation...), state information (e.g., an actuator is closed or opened, a sensor is correctly functioning...) and value information (e.g., number of beads in the bottle, number of bottle to deliver...).

When received by the SCADA server, these information are stored in a database. In our industrial process, this database is of type '*Microsoft SQL Server 2014*'. As shown in Figure 2, there are 3 columns representing the date and time of recording, the name of the register saved and the value of the register.

	currDateTime	registerName	registerValue
1	2021-08-19 15:09:49.700	MW150	6400
2	2021-08-19 15:09:59.920	MW150	12800
3	2021-08-19 15:10:10.147	MW150	19200
4	2021-08-19 15:21:56.080	M103	0
5	2021-08-19 15:21:58.280	M103	1

Fig. 2. Values saved in the database

To confirm the cause of an anomaly, it is necessary to have knowledge of the production order. Each time a production order is sent to the industrial process, it is also stored in the database. Finding an anomaly is hence quite easy. The information get from the industrial process are compared to the ones contained in the production order. If they differ too much, an anomaly occurs in the industrial process and it is possible to know exactly where the process fails.

B. Drone

To function properly in either indoor or outdoor, the drone has to be position in all these environments. In outdoor, the drone already embeds a GPS and can use it to be localized. In indoor, the GPS does not function due to the walls of the building. To avoid injuring human operators and to be able to place itself at a given place in the industrial process, a precise positioning system is required. A precision of few centimeters is required. However, the inertial unit of the drone is not precise enough to answer this need and so, another solutions is required.

1) Required equipment and SDK: To position itself in indoor and perform its task, the drone needs complementary elements. An UWB tag and a camera (used in Section IV-C) are connected to a Raspberry PI 3 which is itself connected to the drone.

As the drone is made for developer, DJI provide a SDK (Software Development Kit). This SDK can be installed on a Linux computer and so Raspberry Pi are compatible. The SDK provided is called Onboard-SDK. It can be accessed through the serial port. Thus, we connect the RX/TX pins of the Raspberry Pi to one of the UART port on the drone. From these pins, sensors values (such as speed, acceleration, heading, altitude, etc.) can be read. In order to set up the SDK, it is necessary to use the application provided by DJI: DJI Assistant 2. This application helps to configure the baud rate, the transmission frequency of the sensors and many others parameters.

The baud rate is set to 230400 Bd and the sensors are set to 10Hz to allow a reading every 100 ms.

2) Indoor positioning: To properly position the drone in indoor, the UWB technology offers a very precise location for an acceptable cost [7]. We have chosen the Decawave MDEK1001 cards for the positioning.

The UWB cards can have two functions: 1) the anchors that are positioned to a known position and periodically transmit signals and 2) the tags that listen to these signals and determine a distance that separate them from these anchors. One tag is fixed on the drone and connected to a USB port on the Raspberry Pi. The position of the anchors is fixed at initialization.

To read the distance from the tag to the anchors, commands are sent via the serial port:

- "*les* command:" shows the distance between the tag and the 4 nearest anchors. Example : 1151[5.00,8.00,2.25]=6.48 0CA8[0.00,8.00,2.25]=6.51 111C[5.00,0.00,2.25]=3.18 1150[0.00,0.00,2.25]=3.16

To position the drone indoor, its position has to be approximated. As the distance that separates the tag from an anchor is not exact (due to noise, doppler effect...), trilateration methods cannot be used. To position the drone in 3D, at least four anchors are required.

To approximate the position of the drone, we compare two methods: 1) the Bancroft method [23] that uses as many anchors as needed, and 2) the Newton/Raphson method that only uses four anchors.

The Bancroft method is solved analytically and so is very fast. To improve the precision of the method, it is possible to use the distances of more than 4 anchors. To implement this method, we use the *GPSTk* library proposed by the Laboratory of Space and Geophysics in the University of Texas at Austin [24].

Unlike the Bancroft method, the Newton/Raphson method is iterative. This method is efficient to find numerically an approximation of a function but longer than the Bancroft method. Moreover, this method is limited to distances between four anchors and one tag.

After testing these methods (see Section V-A for more details), all the methods are executed in less than 100ms. The Newton/Raphson method is the method with the best accuracy. On average, the accuracy is about 5 centimeters that is quite acceptable for the indoor positioning of the drone.

3) Indoor displacement: To move indoor, the best choice is to go straight to the destination at a certain altitude. By maintaining a certain altitude ensures that the drone flies above the machine and cannot hurt human operators.

To go straight to the destination, the drone has to know the angle that it has to rotate from its initial orientation to the orientation required to reach the destination. After calculating this angle, the drone can rotate of this angle and flies in straight line to reach the position of the destination.

In actual drones, the compass is linked to the GPS sensor. If the GPS sensor is not functioning because it does not receive GPS signals due to building walls, the compass is not functioning also.

The solution is to use the inertial unit that is present in the drone. The inertial unit of the drone gives a value corresponding to an angle with respect to North. These values are quite stable, so the inertial unit is a good candidate to replace the compass.

The values returned by the inertial unit (see Figure 3) are: North is at 0, East at 0.69, South at 0.9999 or -0.9999 depending on the angle and the initial rotation of the drone (North to South by East: positive values / otherwise negative values), West at -0.86.

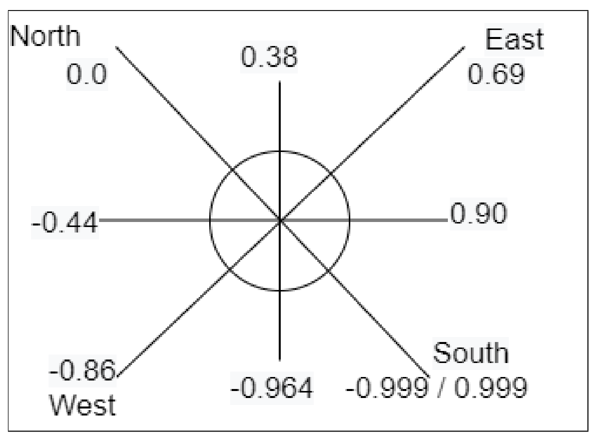

Fig. 3. Diagram showing the sensor data according to the angle of the drone

After calculating the angle to correctly orientate the drone towards the destination, it is necessary to estimate the value to reach in the inertial unit. A linear interpolation is used to find the inertial unit value to reach according to the desired angle (see Listing 1).

Once the inertial unit value to reach is estimated, the drone takes off from its initial position to reach a safe altitude i.e., above machines and human operators. It rotates slowly to reach the estimated inertial unit value and then goes

```
double LinearIteration(double angle, double ang_min,
  double ang_max, double inerUnit_min, double
  inerUnit_max)
{
  return (angle - ang_min) * (inerUnit_max -
  inerUnit_min) / (ang_max - ang_min) + inerUnit_min;
}
```

Listing 1. Linear iteration function

straight forward to reach its destination. On long distances, it is possible to adjust midway the estimated inertial unit value and so the angle to the destination. To increase the safety of human operators, they can carry a UWB sensor on themselves. At destination, the drone descend to the desired height only if all operators' UWB sensors are at a safe distance of the drone.

C. Anomaly confirmation and diagnosis

To confirm an anomaly and make a prediagnosis, the drone can analyze the images itself or send them to the human operator. The last solution is not really efficient because the drone will send a lot of images and increase the workload of the operator.

The first solution seems the best one because it reduces the bandwidth consumption and so increases the lifetime of the drone. Moreover, the drone only sends to the operator images after analyzing them. It can only send one image per type of anomalies detected in order to reduce the workload of the operator.

To detect that an anomaly is present in a image, a convolutional neural network can be used. As it has to be executed on a Raspberry Pi in real-time, the size of the model has to comply with the resources of embedded systems. YoloV5 [25] is a real-time object detection system. It allows to get results very quickly for a given dataset. Once a model is trained, it is possible to detect objects on a video in real time. Moreover, YoloV5 proposes different size of the model. For this reason, this system was chosen in our approach.

Four models are available in YoloV5: S, M, L and X. They are called YOLOv5[smlx]6 in the Fig. 4. To select the model that fits the most to our system, we compare them (see Figure 4) using values provided by the creators of YoloV5. Values are given with the dataset COCO val2017 on 5000 pictures (see [25] for more details). The model S has a very small size so it is very fast. But it gives poor results. The model X is a lot larger than the other models. Same if it has the best accuracy, this one is limited in comparison to the models M and L. However, its computation time is a lot slower in comparison to both of these models (i.e., 60% slower). As a trade-off between accuracy and energy has to be found, the models S and X are not interesting in comparison to the two other models.

Models L and M are quite similar in size and in execution time (and so, in energy). The model L is retained due to its better accuracy.

Finally, the training of the model is done on a server. The parameters used for the training are: `img=640, batch=1, epochs=100`. The characteristics of these parameters are the following:

- img : determine the size of the images. Larger the image is, longer the learning time is.
- batch : the number of images that the neural network takes at each propagation in the network.
- epochs : the number of training sessions. For images and videos this number must be large (more than 100, sometimes it is even recommended to be 500/1000 if needed).

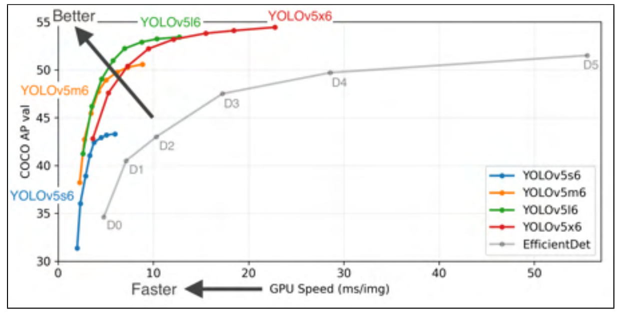

Fig. 4. Comparison of available models

V. RESULTS

To evaluate the performance of our architecture, the continuous monitoring of the platform has to be achieved by the drone. For that, we generate 50 anomalies of two types on the platform. The first anomaly type is a lack of tube on the pallet. The second anomaly type is a lack of cap on the pallet.

If an anomaly is detected via the reading of the registers of the automatons (and thus the sensors) then a message is sent to the drone by the java application. Once the message is received, the drone takes off and is placed above the sensor (the message contains the coordinates of the sensor).

For 20 anomalies, an operator with an UWB tag is close enough of the sensor to analyze the situation and restore the process to a functioning state. Each time, the drone detects the operator and stays at a safe distance of the operator. When this situation occurs, the drone goes back to its home station.

For the 30 other anomalies, no operator are present in the neighboring. The drone descend to the desired height and the Raspberry Pi takes a picture of the sensor that poses a problem and analyzes via the neural network if the anomaly is confirmed or denied.

If the anomaly is confirmed then it is necessary to apply a treatment on the automatons e.g., stop the platform. If the anomaly is invalidated, in this case, everything resumes as it was at the beginning. In both cases, the drone returns to its original position and is ready to take off again if a potential anomaly is detected again.

A. Indoor Positioning

As two methods can be used to position the drone, we compare these methods to determine which one is the most efficient and if these methods respect the constraint to return a position in less than 100ms.

As shown on Figure 5, the drone is approximately positioned in the middle of the anchors for the test. The test is done with 7 anchors of known position and a tag is located on the top of the drone. The position values for each anchor or tag are indicated in this way: X, Y, Z. The distances between the anchors and the tag (dotted line) are distances measured with a meter and the drone position is also measured with a meter in relation to the reference anchor (anchor 0 in $\begin{bmatrix}0 & 0 & 0\end{bmatrix}$).

The computation for the Bancroft method is done with 7 anchors and only the four anchors with the best signal strength are used for the Newton/Raphson method.

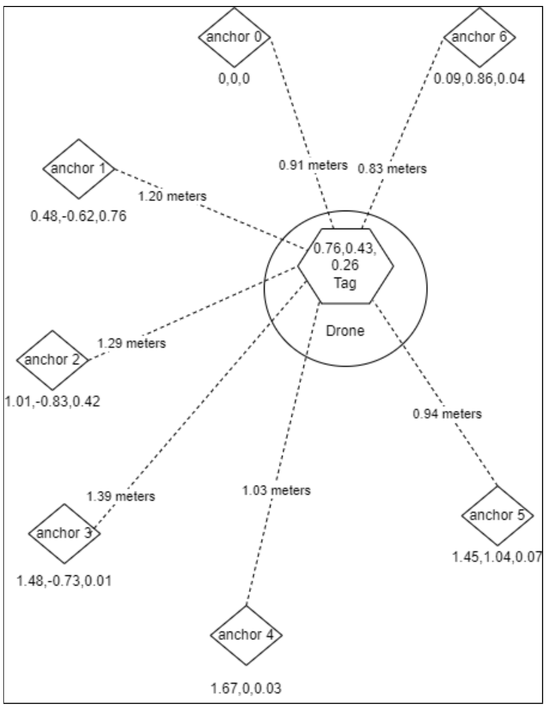

Fig. 5. Diagram showing the positioning of the drone and the anchors with measured distance

The results are shown in Table 2 for 1000 iterations. The execution time is largely inferior to 100ms for both methods. To compare the accuracy of both methods, the error is calculated for each method using the root mean square error (RSME). The values of the result vector are very different according to the methods. The Bancroft method has the lowest accuracy same if it uses 7 anchors. To solve the problem analytically, the Bancroft method does the assumption that the error is the same for each measurement. In reality, it is not the case and so anchors with a lower signal strength have a higher error. These anchors impact the accuracy significantly.

B. Anomaly confirmation and diagnosis

To confirm the anomalies, it is necessary to train the convolutional neural network on a dataset. First, we create a specific dataset that contains labeled images for both anomalies. Once trained, the model is loaded in the Raspberry Pi and the drone executes its mission when an anomaly is detected.

Method	Execution time	Value of the result vector	RSME
Reality	/	0.76 0.43 0.26	/
Bancroft	≅ 840 µs	0.98 -0.11 0.35	0.34
Newton/Raphson	≅ 840 µs	0.78 0.38 0.26	0.07

TABLE I
RESULTS OBTAINED WITH BANCROFT AND NEWTON/RAPHSON

1) Building the data set: In order to make the neural network learn the anomalies it was necessary to take a certain amount of images highlighting the anomalies that we seek to detect.

The dataset is thus composed of 25 images of caps, 25 images of transparent tube and 25 images of tube with a colored paper inside (see Figures 6 and 7).

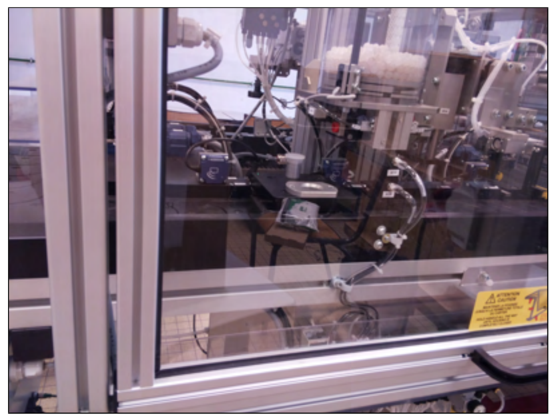

Fig. 6. Sample image: Tube with cap

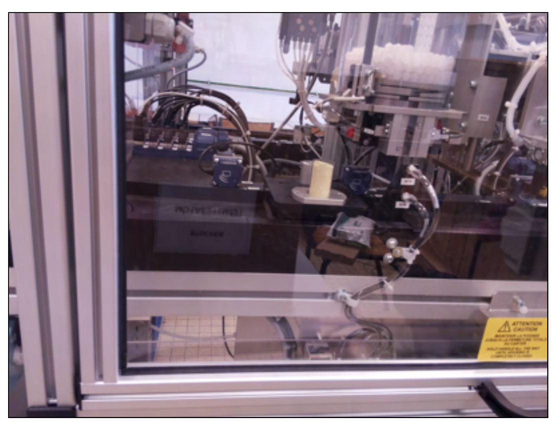

Fig. 7. Sample image: Colored tube without cap

2) Image labelling: Once the dataset is done, it is necessary to label the images. For that, the zone that contains the objects to highlight has to be specified.

In order to label this dataset, that is composed of 75 images, the open source software *labelImg* is used [26]. Its interface allows a human operator to easily create the rectangle to indicate where the object is located in the image. Once the zone that contains the object is known, it is possible to specify a class for this zone. The labeling of the images is quite fast, because the full dataset was labeled in about 20 minutes.

The file created during the labeling process just gives for each image a line in the form: *class centerX centerY width height*. The class is represented by a number (0, 1, ...) depending on the number of classes in the dataset.

3) Anomaly detection: After training the model on a server with 64 Gigabytes of RAM, a Nvidia Quadro P4000 graphics card and an Intel Xeon Gold 5122 processor at 3.6GHZ with 8 cores / 16threads, the model is loaded on the Raspberry Pi. The training of the model with 1 batch and 100 epochs takes about 3.5 hours.

For the 30 anomalies without operators in the neighboring, the drone takes an image of the zone (see Figure 9 for predicted images by the drone).

The normalized confusion matrix that is obtained is shown in Figure 8). For each 30 anomalies, the algorithm perfectly detects the caps and the tubes (blue squares with a value of 1). The FP background value is an element of the background of an image that has been classified as a tube (gray square with a value of 1). What we need to understand from the image 8 is that all the caps or tubes are well detected but in addition to that some parasitic elements are detected as tubes in some images which causes false positives. The diagonal elements represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier. The higher the diagonal values of the confusion matrix the better, indicating many correct predictions. Such images increase the workload of the human operators. Usually, such images have a low probability for the prediction. Most of the time, it is so possible to remove such images.

VI. CONCLUSIONS

For monitoring an industrial control system, drones are a lot more effective than ground robots. They are faster and can fly over the machines or the personnel present in the industry. They can help operators to determine the cause of an anomaly and can inform maintenance operators more precisely.

To confirm the presence of an anomaly and help operators to determine its cause, a new architecture is required for cooperation between the drones and the industrial control system. This cooperation leads to an active monitoring of the process by analyzing the possible failures detected by the system.

This architecture is composed of three elements. First, the SCADA server gets information on the industrial process from the different automatons that it contains. These information and the production orders are stored in a database.

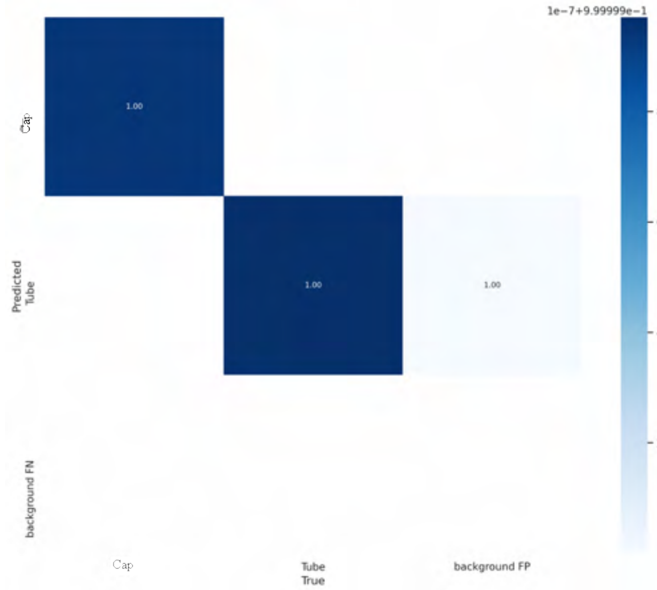

Fig. 8. Confusion matrix

Second, the java application periodically accesses to the database and compare the information obtained from the industrial process to the ones contained in the production order. If they differ too much, the java application alerts the drone that an anomaly occurs.

Finally, the drone embeds a Raspberry Pi 3B+ that allows to execute the SDK provided by the drone manufacturer and gets information on the anomalies from the java application. access the data of the sensors stored in the database. Once the location of an anomaly is known, the Raspberry Pi can control the autopilot of the drone to carry out displacements between a starting point and a point of arrival in an automatic way.

The Raspberry Pi executes a convolutional neural network (i.e., YoloV5) to confirm the anomaly and detect its cause. The model is trained offline on a dataset. The drone can take images of the area where the anomaly occurs and can feed them to the neural network. When the drone confirms an anomaly, it sends the image to the SCADA server in order to help operators to understand why the process fails.

To test our architecture, we interconnect the drone to a real industrial platform. The drone moves indoor safely and takes images of the area of concern with great precision. We train the neural network to confirm two anomalies and the results are outstanding. All the anomalies are correctly classified and the pictures that are sent to the operators are of great help for them to understand their causes.

REFERENCES

[1] M. Salhaoui, A. Guerrero-González, M. Arioua, F. J. Ortiz, A. El Oualkadi, and C. L. Torregrosa, "Smart industrial iot monitoring and control system based on uav and cloud computing applied to a concrete plant," in *Sensors*, vol. 19, 2019.

[2] D. Mourtzis, J. Angelopoulos, and N. Panopoulos, "Uavs for industrial applications: Identifying challenges and opportunities from the implementation point of view," in *Procedia Manufacturing*, vol. 55, pp. 183–190, 2021.

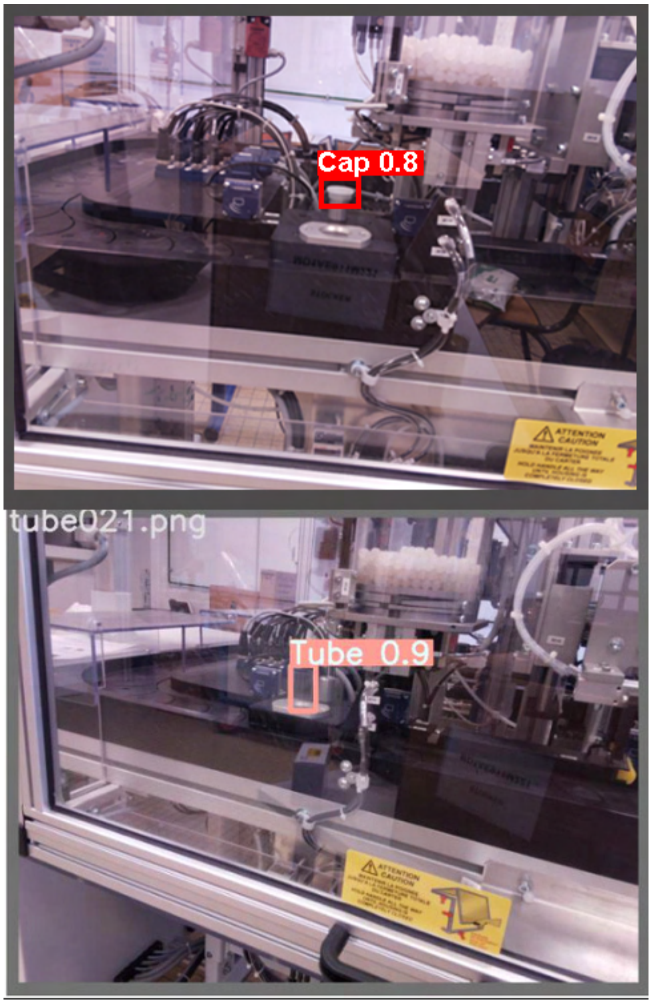

Fig. 9. Example of prediction made by learning

[3] J. M. S. Lagmay, L. Jed C. Leyba, A. T. Santiago, L. B. Tumabotabo, W. J. R. Limjoco, and N. Michael C. Tiglao, "Automated indoor drone flight with collision prevention," in *TENCON 2018 - 2018 IEEE Region 10 Conference*, pp. 1762–1767, 2018.

[4] P. Suwansrikham and P. Singkhamfu, "Indoor vision based guidance system for autonomous drone and control application," in *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, pp. 110–114, 2017.

[5] Y.-R. Chen and Y.-C. Chen, "Path planning in large area monitoring by drones," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, pp. 295–299, 2018.

[6] S. Khruahong and O. Surinta, "Develop the framework conception for hybrid indoor navigation for monitoring inside building using quadcopter," in *2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, pp. 1–6, 2019.

[7] B. Yu, L. Xu, and Y. Li, "Bluetooth low energy (ble) based mobile electrocardiogram monitoring system," in *2012 IEEE International Conference on Information and Automation*, pp. 763–767, 2012.

[8] T. H. Nasution, I. Siregar, and M. Yasir, "UAV telemetry communications using ZigBee protocol," *Journal of Physics: Conference Series*, vol. 914, p. 012001, oct 2017.

[9] D. Dhawan Soni and D. Dhawan Soni, "Project of implementation of an ultrasound positioning system and other sensors in an industrial UAV." https://upcommons.upc.edu/handle/2117/186298, june 2018. Accessed: 13/02/2022.

[10] H. Yu, T. Xie, and B. Wilamowski, "Recent advances in industrial control," in *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4626–4631, 2011.

[11] J. Whelan, T. Sangarapillai, O. Minawi, A. Almehmadi, and K. El-Khatib, "Novelty-based intrusion detection of sensor attacks on unmanned aerial vehicles," in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, p. 23–28, Association for Computing Machinery, 2020.

[12] R. Babaie and A. F. Ehyaei, "A novel uav cooperative approach for industrial inspection and monitoring," *The International Power System Conference (PSC)*, 2016.

[13] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, "A uav system for inspection of industrial facilities," in *2013 IEEE Aerospace Conference*, pp. 1–8, 2013.

[14] M. Salhaoui, A. G. Gonzalez, M. Arioua, J. C. M. Molina, F. J. Ortiz, and A. E. Oualkadi, "Edge-cloud architectures using uavs dedicated to industrial iot monitoring and control applications," in *2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, pp. 1–6, 2020.

[15] R. Kristiansen, E. Oland, and D. Narayanachar, "Operational concepts in uav formation monitoring of industrial emissions," in *2012 IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 339–344, 2012.

[16] X. He, "Rapid development of unmanned aerial vehicles (uav) for plant protection and application technology in china," *Outlooks on Pest Management*, 2018.

[17] S. K. Phang, M. A.-A. Hassan, Z. Y. Wong, Z. Y. Ng, and Y. L. Lai, "Development of autonomous uav systems for low light surveillance applications using night vision camera," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 10, no. 13, pp. 1379–1391, 2018.

[18] E. Karakose, "Performance evaluation of electrical transmission line detection and tracking algorithms based on image processing using uav," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–5, 2017.

[19] F. Wang, S. K. Phang, J. J. Ong, B. M. Chen, and T. H. Lee, "Design and construction methodology of an indoor uav system with embedded vision," *Control and Intelligent Systems*, vol. 40, no. 1, p. 201, 2012.

[20] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, "A survey on vision-based uav navigation," *Geo-spatial Information Science*, vol. 21, no. 1, pp. 21–32, 2018.

[21] K. Xu, "Application analysis of PLC in UAV field," *Journal of Physics: Conference Series*, vol. 2037, p. 012106, sep 2021.

[22] S. Alem, D. Espes, E. Martin, L. Nana, and F. De Lamotte, "A hybrid intrusion detection system in industry 4.0 based on isa95 standard," in *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–8, 2019.

[23] E. S. Agency, "Bancroft method - navipedia." https://gssc.esa.int/navipedia/index.php/Bancroft_Method, Jul 2014. Accessed: 30/08/2021.

[24] Space, G. Laboratory, and U. of Texas at Austin, "Gpstk - github." https://github.com/SGL-UT/GPSTk. Accessed: 30/08/2021.

[25] Ultralytics, "Yolov5 - github." https://github.com/ultralytics/yolov5. Accessed: 31/08/2021.

[26] Tzutalin, "Labelimg. git code (2015)." https://github.com/tzutalin/labelImg. Accessed: 31/08/2021.