2017

LULEÅ TEKNISKA UNIVERSITET

Mobile Multi-Robot Coordination for Logistics Automation

PROJECT IN ELECTRICAL ENGINEERING

JAIME VICENTE GARCIA REPRESA

LULEÅ UNIVERSITY OF TECHNOLOGY

Contents

1. INTRODUCTION

The purpose of this project is to show a type of Automated Storage and Retrieval System, which has become very used in recent years, using robots as Automated Guided Vehicles.

First, there is a comparison between the common systems used in warehouse showing his advantages and disadvantages. The case of study chosen is that of a medium size company that is growing steadily and wants to improve his logistic area. Its decision of implement one system or another depends on the field of work of the company, the quantity of storage needed or the time goods are stored.

Next the characteristic of storage systems are presented form a theoretical point of view. Also the different software used to command the equipment of a warehouse are schematized for identifying the level at which they work. In the solution proposed, the equipment used consists of robots, so the main technologies that will be used are presented. Then a breve description of the system is showed with the help of a UML diagram, and the equipment where the system will work is presented with its interfaces.

After presenting all the components of the system, the solution is implemented with learning equipment for mobile robots, which enables fast prototyping and safe execution. A demonstration is performed and recorded of the system normal operation.

2. PROBLEM DESCRIPTION

2.1 Problem statement

In this project the initial conditions start from a classical warehouse of a small/medium size company, where all actions are manual, there is no process automatized and every product is sorted by an employee, keeping the record by software, typically in a Warehouse Management System. The company wants to improve efficiency and reduce the cost and size of the warehouse, so it sends this request to an engineering team.

The warehouse is part of the building where the company carries out his operation, so its dimensions cannot be increased, although if the dimensions are decrease the empty place left could be used for other purposes that the company requires.

The company is willing to spend his savings in this project, however is not willing to hire more people to manage the warehouse, so people would have to be trained if there are manual task, or for monitoring the system. As a remark, the budget to implement the solution is out of the scope of this project.

2.2 Proposed solution

A fully Automated Storage and Retrieval System (AS/RS) is appropriated for large companies with a big stock, where the rate of operations is more crucial and the greater speed significantly affects the delivery time. The typical structure used is the Unit load AS/RS, A large automated system designed to handle unit loads stored on pallets or other containers, with other types been variations depending on the storage density (Deep-lane AS/RS), size of the unit load (Mini-load AS/RS) or the direction of the aisle (Vertical lift storage modules). But all this systems share some characteristics:

- High initial investment to install the system
- Need a rack structure compatible with the unit loads in used
- The distribution of the storage is in aisles, so each crane machine controls a fixed area of the store/retrieve products, and a pickup and deposit stations to bring the product out of the storage system.
- Provide high Storage density, System throughput, and relatively high accessibility.

As we analyze these requirements it can be seen that they don't match very well the needs of a small/medium size company.

Figure 1. Automatic Vertical Storage AS / RS automatic take & delivery system

The company is prepared to pay an initial investment but this payment may exceed their predictions. At first we don't know if the company already has a rack structure but it will probably not possess a pickup and deposit stations for transporting the unit loads, which could filled the empty space created thanks to the high storage density. The high system throughput may be too high for this size of company, showing low level of utilization during its lifetime, a symptom of over-designed. Finally we could add that the availability of the system is not so high. To the computer failures and maintenance we need to add that is case of failure of a machine of storage/retrieval, the area where it was working will not be available to the rest of the machines, so that area is unusable.

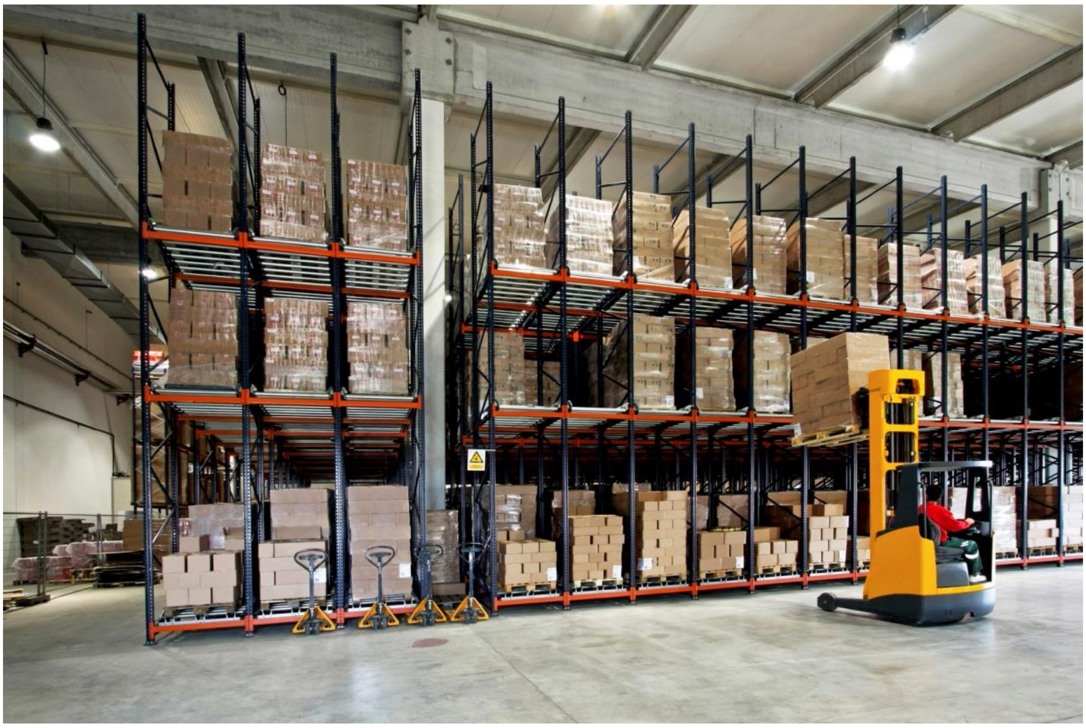

Figure 2. Warehouse with forklift

In the case of a small company, we could improve the efficiency with a Man-on-board AS/RS or with a fleet of forklift to assist the operators, they are the most expensive picker-to-stock equipment alternative, but are less expensive than a fully automated system, although the operation still manual. The operator commands the storage/retrieval machine or forklift, so it keeps the necessity for a rack structure, and the operators need to be instructed in the use of such machine. There will be also an initial investment, but much lower than the fully automated system. Advantages of this system are that is not so fixed, it can be upgraded with just another rack, another machine if more storage capacity is needed, or with an operator and machine if system throughput is what we are aiming for. Also the availability is not so compromised in case of failure of a machine, as operators may come with a temporary solution. Still we have some field to improve, increasing productivity and decreasing cost and size, by using robots to perform the task that humans would do, when these tasks are repetitive and commanded by computer.

Figure 3. Driverless Forklift

The last option is an AS/RS operated by Automated Guided Vehicles (AGV). This is the less expensive of the fully automated systems, as only the shelves will need to be replaced, in order to integrate the rack structure for the robots. Also some reference lines will be needed on the floor to help the guide system of the AGV, but this is easily changed, making this kind of system very flexible and capable to adapt to the needs of any small/medium size company. In addition, this system can work side to side with human operators thanks to their reactivity, and the collision avoidance system, that may be able to help them get back on track again after an unexpected change happened, like an obstacle in their way to the racks. It has the same advantages of the Man-on-board system, with less fixed cost, but more initial investment. However, it may not achieve the Storage density of other types of fully automated AS/RS system, as the height is restricted by the robots range, but it can achieve the highest System throughput. This is the reason why recently many companies are choosing this type of storage system in their pick-pack-and-ship warehouses.

We assume that the company has decided to install an AS/RS system operated by AGV robots, and has saved enough to pay its full implementation.

3. LITERATURE REVIEW

Storage is an essential function in an automation system. The material storage system allows materials to be stocked for a specified period of time, waiting for a customer or for another process of the automation system.

There are two main facilities used for storage: Warehouses and Distribution Centers. They play a critical role in the flow of goods from manufacturers to consumers, because they serve as giant routing centers in which pallets of products from different manufacturers are split and the items are redirected into outgoing transports to costumers, and sometimes in the case of Warehouses the product goes back into the automation system [1].

According to Baker and Canessa [2] warehouses are also very significant from a cost perspective representing about 22% of the logistics costs in USA and about 25% in Europe. With this critical impact in the performance of the supply chain network and with high logistics costs research aimed to optimize warehouse operations has been continuously performed but is still necessary.

To characterize the performance of a storage system we use different measures [3]:

- **Storage capacity**: Defined as the total volumetric s space available and the total number of storage compartments in the system available for holding items or loads. The physical capacity of the storage system must be greater than the maximum number of loads that can be stored, to allow for additional, emergency, storage requirements.
- **Storage density**: Defined as the volumetric space available for actual storage relative to the total volumetric space in the storage facility. Aisle space and wasted overhead space are two examples of unutilized storage space.
- **Accessibility**: Defined as the capability to access any desired item or load stored in the system. Density often has an inverse relationship with accessibility to the storage facility.
- **System throughput**: Defined as the hourly rate at which the storage system receives and puts loads into storage, and/or retrieves and delivers loads to the output station. Storage system must be design to meet maximum throughput requirements, as variations in the levels of throughput should be expected throughout the working day. A dual command cycle reduces throughput by combining storage and retrieval functions. The ability to perform such a cycle is dependent on demand and scheduling issues.
- **Utilisation**: Defined as the proportion of time that the system is actually being used to perform storage/retrieval operations compared with the time it is available. If system utilisation is too low, then it is probably over-designed; if it is too high, then there will be no allowance for rush periods or system breakdowns.
- **Availability**: Defined as the proportion of time that the system is capable of operating compared with the normal schedule shift hours. System breakdowns cause downtimes in the system.

With the new storage systems what they try to improve is mainly storage density and system throughput, as the machines need less space to perform the operations than the humans and they are capable of doing it in less amount of time, thanks to the fast communications between the machines and the central computer controlling the warehouse.

There are different software programs used in an automated warehouse, and so Logistics execution software (LES) can be classified into two areas: Warehouse Management system (WMS), and Transportation Management System (TMS).

Even though WMS and TMS are classified as execution software, they do not have capability to interface or control facilities in the warehouse. They manage order level execution, and the WMS software also helps in the integration of other host system as Enterprise Resource Planning (ERP) and Supply Chain Management (SCM). The control of automatic facilities requires control software such as Warehouse Control System (WCS), Material Flow Controller (MFC), equipment Management System (EMS) and Equipment Control System(ECS). Sometimes it happens that the terms WMS and WCS are used as synonyms, however it's clear that each software pursues different objectives [4].

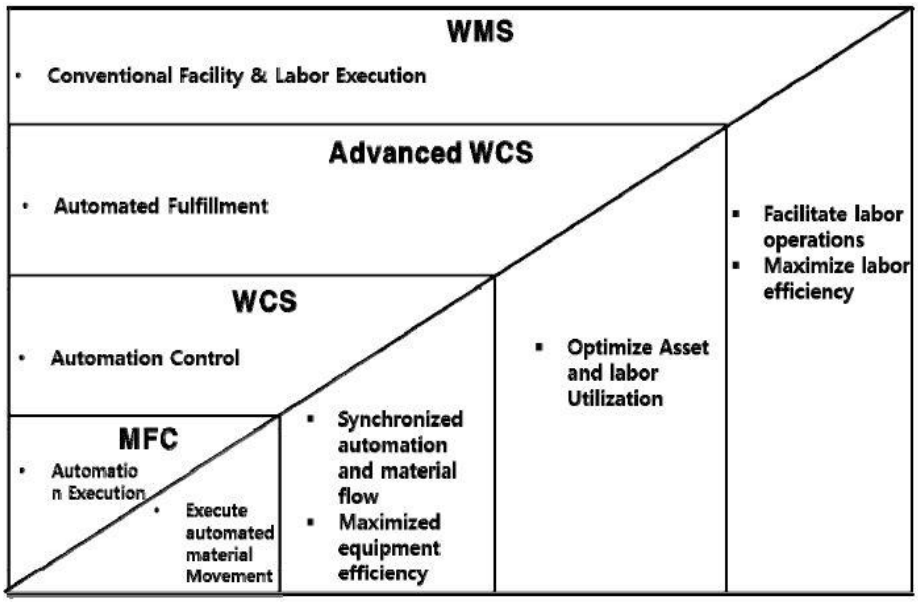

Figure 4. Different software used in warehouse

In out solution we proposed a storage system composed of AGV. Each robot will have his ECS integrated inside, the computer that controls the sensors of the warehouse will be the MFC and so the WCS will be in charge to coordinate the system telling each AGV where to go knowing where each pallet is, thanks to some tracking technology like RFID.

Looking now at the AGV, they have been used to move material within warehouses since the 1950s. They have been used primarily to transport very large, very heavy objects like rolls of uncut paper or engine blocks. For many decades, industrial robots were confined to performing extremely precise and repetitive tasks, and were separated from human workers. The confluence of inexpensive wireless communications, computational power, and robotic components are making autonomous vehicles cheaper, smaller, and more capable. However, robots are still incapable of tasks that require fine manipulation or improvisation, so it is useful to devise ways for robots to collaborate with humans more effectively [1].

Most of the AVG use the same kind of technologies to achieve their task. They have wheels with encoders to measure the distance traveled, some equip a GPS to track them, and others have restrictions in their range of movements, like moving in a grid, or use some type of marks on the floor for orientation. Leaving aside their ability to move, they can have many other skills, which come with extra equipment. There are many different actuators for operations in industry, but as the focus is in logistics, the main tool use to move materials inside a warehouse is the forklift.

The main technologies used in the AVG robots for this project are: rotatory encoders in each wheel, to move between the different stations, an inductive sensor in the front, to follow the metal lines laid at the entry of the stations, a diffuse light sensor in the front but moved aside, to detect the colored marking on the floor signaling the end of the station.

A rotatory encoder is an electro-mechanical sensor that measures the rotation or angular position of a shaft. Its output is an electric signal, digital or analog, so it's a type of transducer. There are two main types of rotatory encoders, absolute or incremental. The ones used in the robots of this project are incremental and so it is the focus. Incremental encoders have no sliding contacts and are not subject to wear. A light-emitting diode, a slotted metal disc, and a photo-diode array form a photoelectric circuit. An internal logic produces two square-wave signals phased at 90° to each other from the output of the photo-diodes, with or without a reference impulse.

Figure 5. Incremental Encoder

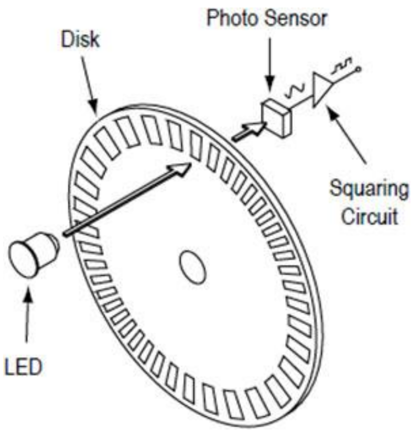

Figure 6. Encoder system

Although nowadays improvements in Hall Effect sensors make them more appealing than rotatory encoders, due to price and resolution, they still the industry standard technique [5].

Inductive proximity sensors are used for non-contact detection of metallic objects. Their operating principle is based on a coil and oscillator that creates an electromagnetic field in the surroundings of the sensing surface. The presence of a metallic object, act as actuator and causes a reduction of the oscillation amplitude. The rise or fall of such oscillation is identified by a circuit that changes the output of the sensor. The operating distance of the sensor depends on the actuator's shape, size and nature of the material, so different results are obtained for aluminum or steel.

Figure 7. Inductive sensor

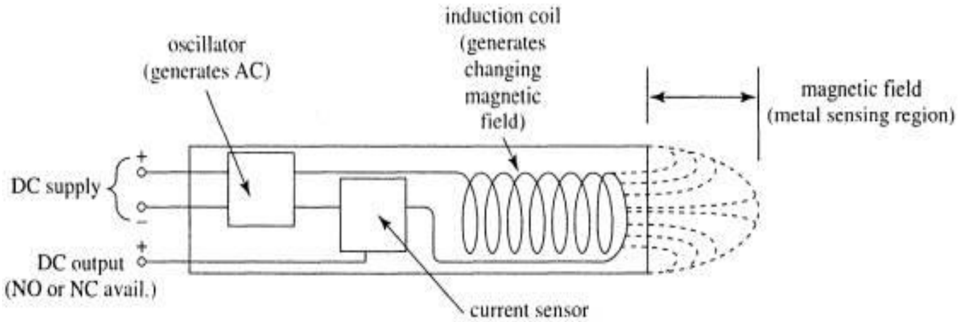

Figure 8. Inductive sensor system

Light diffuse sensors are a type of photoelectric sensor. They detect a change in light intensity, either non-detection or detection of the sensor's emitted light source. The differences between the photoelectric sensors are the type of light and the method by which the target is detected. Photoelectric sensors are made up of a light source, LED, a receiver, a phototransistor, a signal converter, and an amplifier. The phototransistor analyzes incoming light, verifies that it is from the LED, and appropriately triggers an output.

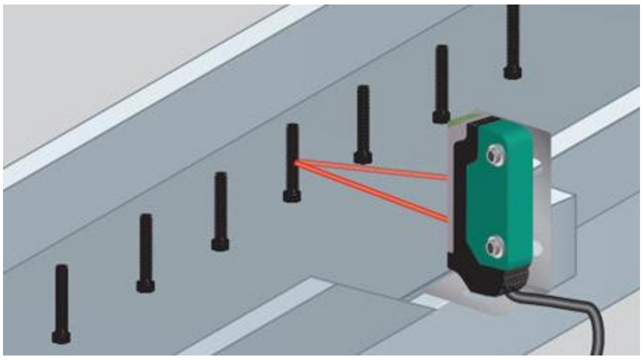

Figure 9. Diffuse light sensor model

In diffused mode sensing the transmitter and receiver are in the same housing. Light from the transmitter hits the target, which reflects light at arbitrary angles. Some of the reflected light returns to the receiver, and the target is detected. Because much of the transmitted energy is lost due to the targets angle and ability to reflect light, diffused mode results in shorter sensing ranges than retroreflective and thru-beam modes, but does not need any secondary device, nor reflector or receiver.

4. SYSTEM DESCRIPTION

According to the proposed solution, it is know what type of technology is going to be implemented, but what is the scope of this AS/RS system.

4.1 Interface

The company wants to completely automate the warehouse, so this system must be able to fulfill the task previously done by human workers. It needs to take the merchandise arriving from suppliers or from the company supply chain, place it where the Warehouse Control System commands and when the time comes, deliver it to the transport that will take it to the final costumer. Then we can try to describe our system through a UML – Use Case Diagram:

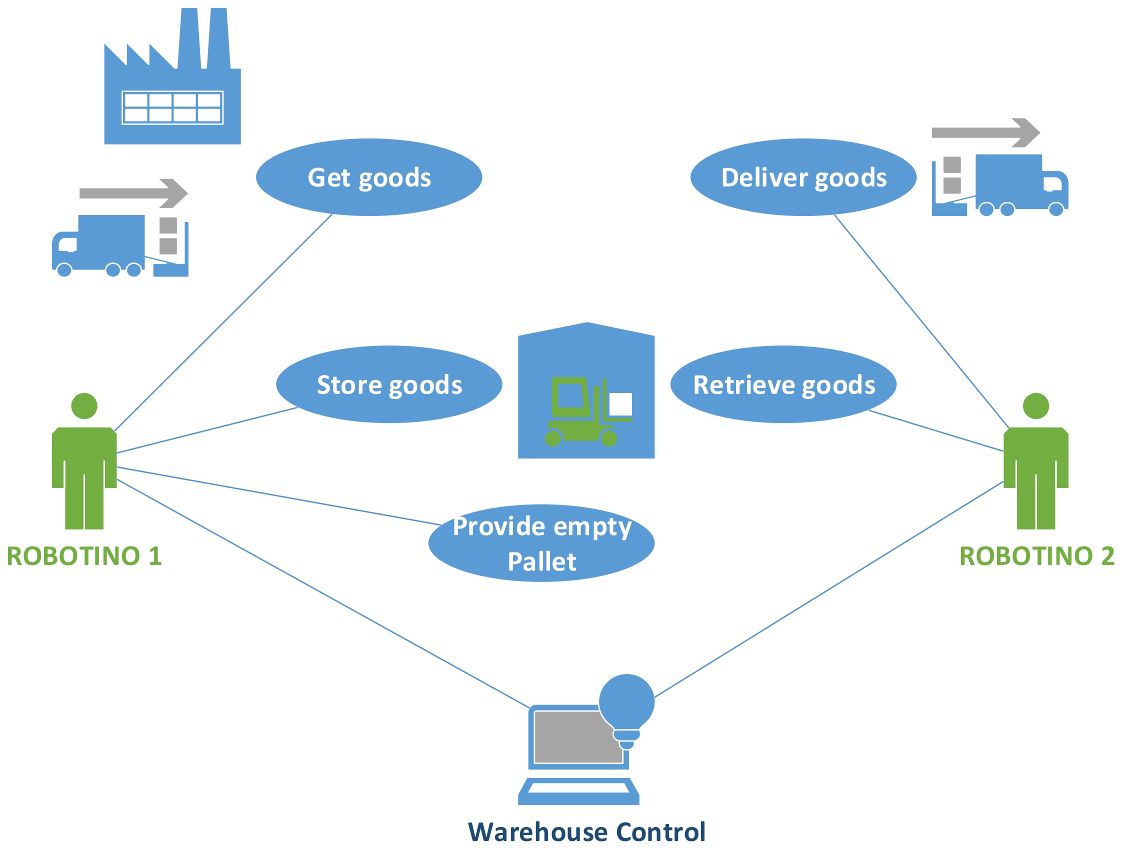

Figure 10. Use Case Diagram

Use Case Description Table

USE CASE	DESCRIPTION
Get goods	The new merchandise arriving into the warehouse has to be picked by Robotino 1
Store goods	The good had to be placed where the warehouse control commands by the Robotino 1
Retrieve goods	Robotino 2 has to take the goods from the designated place by the warehouse control
Deliver goods	The customer orders are shipped once the goods are delivered to the output transport by Robotino 2
Provide Empty Pallets	Robotino 1 is in charged to supply to the previous arranging station with pallet to stuck the goods and be safely transported by the Robotinos

The limitations to our systems come in the way the goods are delivered, the actors being the robots enforces that the loads will be properly arranged in pallets that will enable the forklifts installed in the robots to transport them safely. When the goods are delivered, an intermediate step will be required to pack the goods in pallets and put them in the delivery decks of our system.

4.2 Equipment

This system will be implemented in the ProLog Factory from FESTO. This platform contains its own supply chain, with a variety of stations depending on the desired final product, so they can be mixed according to customer needs. At the end of the supply chain it contains a commissioning station where the products are organized in pallets following customer orders, and once a pallet is complete, the robot arm will place it in the entry deck of the logistics area.

The logistics area is the environment of our system. It is provided with two AVG robots, Robotino 1 and Robotino 2, a high bay rack, where the goods are stored, two input slides, the first is where empty pallets come into the Prolog Factory, and the other is for pallets with goods from the commissioning station, and 4 output slides, one for empty pallets to reach the commissioning station and the other 3 are for pallets from the high bay rack to be delivered to transport outside of the Prolog Factory.

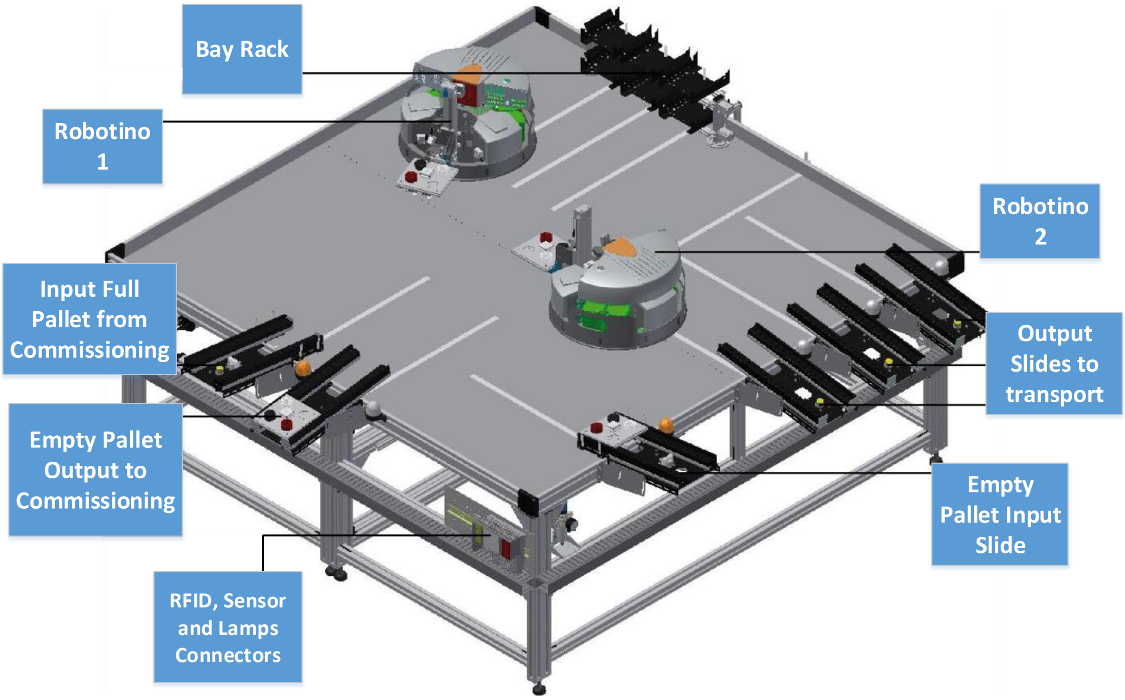

Figure 11. Logistic Area

5.IMPLEMENTATION

The AVG robots in the Prolog Factory, called Robotino, are controlled by an industry-standard PC system, with an Operating System based on Ubuntu Jaunty. Therefore the Robotinos can be programed in various languages including C, C++, Java, .Net, Matlab, LabVIEW and Microsoft Robotics Developer Studio. The company that produces these robots, FESTO, additionally has developed his own graphic programming environment for fast prototyping, Robotino View. The third version of this software, Robotino View 3.1.2, will be the one in used through this project.

The program comes with a Workbook as tutorial to get used to his functions. After completing these exercises we have knowledge of the basic movements of the robots, so we can start to travel between stations. It also explains the sensors that come by default with the robot, the bumper sensor around the robot for collisions, the Infrared sensor for distances between the robot and its surroundings, the inductive sensor for the aluminium tapes on the floor used as guide lines, and the diffuse light sensor for final approach to station. In the tutorials it is suggested to obtain a function of the values read by the analog sensors. As the Infrared sensors are of little use for the project, the graph presented corresponds to the inductive sensor. Later, based on this value a PID controller would be implemented.

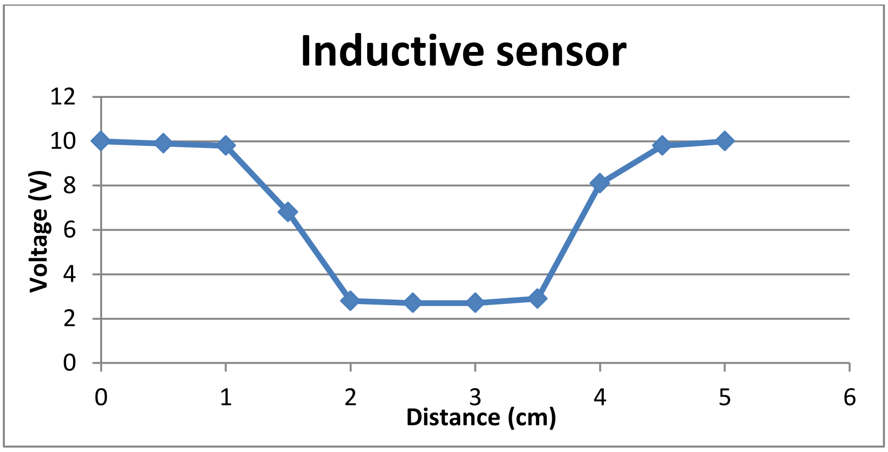

Figure 12. Inductive sensor values

5.1 Movement tuning

To adjust the movements of the robots, turn 90º or move forward 20 cm, the first strategy was based on an internal clock. The velocity was set up constant, in a function block that takes as commands velocity in x and y directions, so the distance travelled was a transfer function of the time elapsed. The main issues with these movements were: at the start, when the robot usually slips, the resolution, which improved with lower speed, but even then the repeatability was low, and the way error keep adding after each step.

Next strategy for creating paths was to read the internal encoders on each of the omniwheels. Looking at the help of the software we see that each motor block has as output the number of ticks read by the encoder, since power up or “reset position”, and that the encoder produces 2000 ticks per round. Instead of using this information, Robotino View already has a block where it maps the ticks of each encoder to a position in an x-y coordinates system, as the one use to set the velocity of the robot. This block, called Odometry, gives as the displacement of the Robotino with respect to an origin coordinates that are introduced arbitrarily at the same time the command to reset is sent. As before, if the robot slip it will induce an error, but the repeatability is improved and the errors do not sum up if the origin is reset.

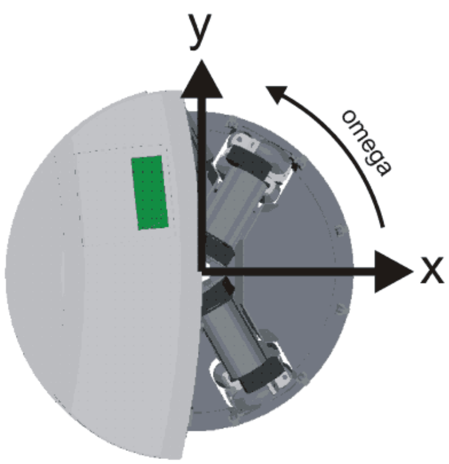

Figure 13. Robotino relative coordinate system

Once basic movements were done correctly across the area, the process of approaching the station accurately was next. To reach the station, the robot has to follow an aliminium tape and stop once it reach a black tape, perpendicular to the metal one, that will trigger the forklift.

However, the robot does not have to follow the center of the aliminium tape, it has to follow the right edge to be able to control it. As there is only one inductive sensor installed in the Robotino, it is not able to recognize which side of the tape he is in. The values read by the sensor are equal on both sides, as shown by the chart above. So we have to assure this in the previous step, always reaching the tape form the right side.
Another constraint is the orientation of the robot when the black tape is reached. In the input slides, there is a bar who can correct the orientation of the forklift before picking up any pallet, but in the output slides there is not. Thus, if the orientation is not almost perpendicular, when the forklift descends the pallet will fall out of the slide path, this is unacceptable.
Putting together these requirements, the PID controller needs to have a fast response when reaching the tape at the start, to not cross the middle of the tape where it will become uncontrollable, besides it needs low deviations at the end, so it reaches an orientation almost perpendicular to deliver the pallet correctly. The formula used for the PID is a rectangular approximation of the analog controller:

$$u(t) = K_p e(\mathrm{t}) + K_i \int d_t + K_d \frac{de}{dt} \qquad \text{Analog PID controller}$$

$$u(n) = K_p e(\mathrm{n}) + K_i T \sum_i e_i + K_d [e(n) - e(n-1)]/T \quad \text{Digital PID}$$

Where u is the output of the controller, e is the error, K_p K_i and K_d correspond to the coefficients of the proportional, integral and derivative respectively. T is the sampling frequency, which is unknown, nevertheless the PID controller is introduce in Robotino View by a Lua script with the option of executing the algorithm only when the values read by the sensor change, so we can simplify it.

With the Robotino accurate movements reached, the tests procced using the forklift of the robots carrying the pallets. This module is attached as an extra to the robots; it comes with two end of stroke sensors at the upper and lower point of the forklift range. The forklift also has the freedom to rotate upwards thanks to a shaft perpendicular to its movement, so if while it is descending hits something under it, the motor will not push it down, rather the forklift will rotate. However it has a top when trying to rotate downwards so it can take weight. Problems come with this module when trying to reach a mid-range position, as this motor does not come with an encoder and a constant velocity is not precisely reached. It shows variations depending on load and battery charge.

Another issue arrived when the robot tried to bring down the empty pallets. Due to the high slope of the slide, the pallet and the teeth of the forklift will collapse in such a way that the pallet got stuck and the forklift with the pallet on it will rotate upwards. This problem was hard to resolve from a coding perspective, changing the position at which the pallet was dropped or moving the robot at the same time the forklift was going down where some of the tests performed. Their purpose was to reach a balance between the slide at the pallet when brought down so they will not collapse, but as this was not safe, other mechanical solutions were tested. First the nut tightening the bolt, teeth of the forklift, was set loose a little bit, but the vibrations during movements produced too much noise. After a comparison with the other output slides, the change in slope was observed, and with this adjustment the problem was solved.

5.2 Communication

As the Robotino is equipped with a standard PC, we could think of many ways to communicate with him. It even has by default a web interface that will open when we introduce his IP address in a web browser. Between the communications available it was clear that wireless had advantages over cables. Moreover it has to be taken into accounts which are present in the software, as there the possibilities are less. Robotino View comes with an OPC client, a UDP server and with Data Exchange Server and Client. As this project will only include programs from Robotino View, the Data Exchange is the connection recommended and the simplest to implement.

For the Data Exchange modules, it is requires a server into one of the programs and arbitrary number of clients to exchange data over an arbitrary number of communication channels. The communication channels are created at server side, so it has to be deployed before the clients. The data of the channels is broadcasted to all clients, they are the ones who choose which information to receive, which channels listen to, and what information to send, through what channel send data. Server and clients have equal rights when exchanging data, so the ones who first write into a channel is the owner, and no one else should write on it, or the final data contained in the channel would be unpredictable. It is similar to a producer–consumer problem, where depending on the channel Robotino 1 could be the producer and the server the consumer or vice versa. This situation prompted an increase in the number of channels needed.

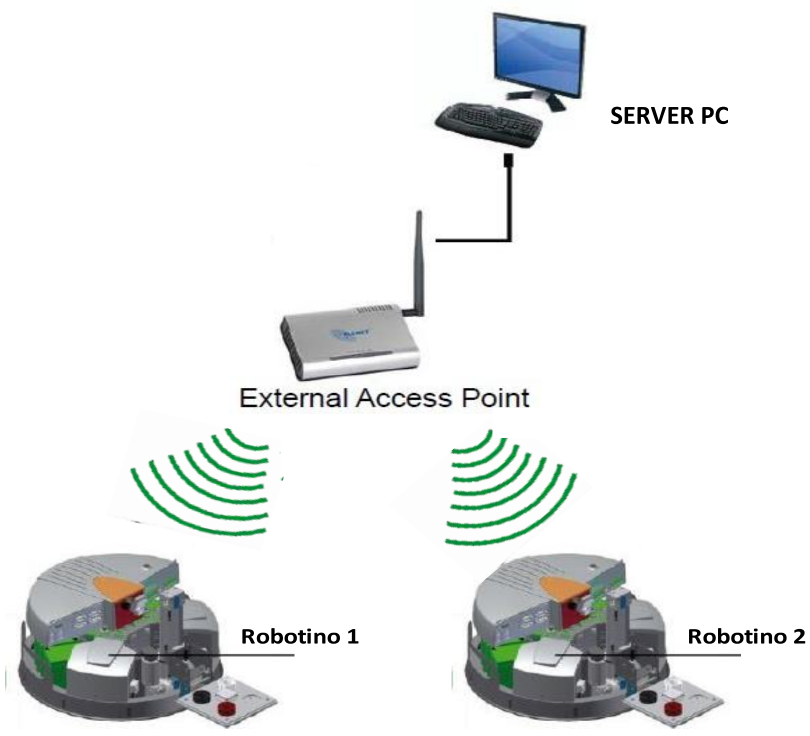

Figure 14. Network architecture

The architecture of this communication is based on the one proposed by FESTO in his manuals. This project uses an independent Wi-Fi network, provided by the router, external access point in the scheme. First, the router is configured to have his own network. Then, a USB stick is connected to the Robotinos for enabling wireless communication, and through the web browser interface they can be configured so that they will connect to the desired network at power up. With the computer also connected to this router, Wi-Fi or Ethernet, the system is complete.
So as wrap-up the connection order is: router to provide network, server to provide channels and clients to exchange data. This order will be the one followed at the boot up of the system, and the programs mapped to each device.

5.3 Programming

For very actor in the system a program in Robotino View is developed. Each Robotino will run a program, as they come with a version of Robotino View preinstalled in the Linux operating system. Although for debugging, the practice was to stablish a communication with the Robotino and run the program in a local PC. This was problematic when the communication was lost, as the Robotino will continue with the last subprogram that was in execution. A better practice consists of having the programs recorded into the devices, and if there is a communication failure, they can progress to a safe step.

Looking at the program for the Robotinos it can be seen some templates that pop out as the operations are repetitive. At the start of the program the initial conditions are stablished, and both programs proceed to a listening phase. Robotino 1 will start as soon as the Warehouse control commands him, while Robotino 2 will also wait for Robotino 2 to finish storing the pallets, or a collision could arrive as they share the same area of work.
Following the orders, a leaving station subprogram starts, where the robots will follow the aliminium guide line on the floor until they are far away from the stations to have all his range of movement available. Then, they will proceed to the station where they will pick a pallet. The only difference between different stations is the coordinates introduce before reaching the destination guide line. Once they detect the guide line from the left side, the Robotino controller will come forward till the black tape, which signals the right position for picking up pallets, or dropping them after little adjustment.
At the same time the pallet is picked up, or dropped, the Robotino will send this information through the wireless communication, and will listen for a new order from the warehouse control. The other main course of action is in Robotino 1 with the empty pallets, where after receiving the orders the first action is to pick up the pallet and the move to the dropping station. When the pallet is dropped it will automatically come back to the starting position. This is similar to the other operations of storing pallets or taking them to the output transport slides. When all the pallets are delivered, Robotino will go back to its starting position to start another branch of the program. This helps reset the errors and stop the system in a stable state.
For the simulation performed during the demo the robots executed all the possible branches going through every state once, and at the end the programs will terminate. However the loop could be closed for them to enter in an infinite loop and execute non-stop the orders of the warehouse control.

6. CONCLUSION and FUTURE WORK

The demo was executed several times, and a video of it was recorded from different points of view. The execution of the program during the video records was not from start to end. Thanks to the communication the subprograms associated to each order can be executed separately. This can be fragmented even more if needed for an operation.

The critical moment of the program comes when storing the pallets, as the forklift does not go all the way to the end, but reaches a medium position. The tolerance for inaccuracy is too small, and if not adjusted properly before the execution of the program it may fail and the emergency stop has to be used. This problem best solution is when the addition of an extra sensor at the middle position. As there were none in stock, an adjustment algorithm with the battery voltage was proposed.

For the future of this work, many options are available. The work area of the robots has many sensor and actuator that were not used, so the system has potential for improvement with extra information. This sensor and actuator are usually controlled with an industrial controller, as a PLC, that could be added next to the I/O connectors. Also an extra power supply will be needed, for the sensors and for the PLC.

If this was available in the future, to add value to the system, the industrial controller will have to communicate with the robots. The best option seems for the industrial controller to execute the warehouse control software and so command the robots, with the information of his sensor and the data form the other stations or a SCADA system. Interaction between the robots and a PLC is something that could be added to the programs made in Robotino View, through the extra modules the software provides.

BIBLIOGRAPHY

[1] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," AI Mag., vol. 29, no. 1, pp. 9–20, 2008.

[2] R. Peixoto , L. Dias , M. S. Carvalho , G. Pereira and C. A.S. Geraldes, "An automated warehouse design validation using discrete simulation", *2016 IEEE 19th International Conference on Intelligent Transportation Systems*

[3] D. O'Sullivan "Unit 9. Automated Storage Systems", National University of Ireland, Galway

[4] D. W. Son , N. U. Kim , Y. S. Chang and W. R. Kim C., "Design of Warehouse Contol System For Automated Warehouse Environment", *2016 5th IIAI International Congress on Advanced Applied Informatics*

[5] D. R. Chris Mechefske and M. Timusk, "Dynamic Sensor Calibration: A Comparative Study of a Hall Effect Sensor and an Incremental Encoder for Measuring Shaft Rotational Position", *2016 IEEE International Conference on Prognostics and Health Management*

[Figure 1] "Almacén automático vertical / de toma y entrega automáticas AS/RS "Jiangsu Union Logistics System Engineering Co. Ltd., 2017. *Automatic Vertical Storage AS / RS automatic take & delivery system*.
Available at: http://img.directindustry.es/images_di/photo-g/81083-8225476.jpg

[Figure 2] "Which Came First the Pallet or the Forklift?" G. Donadio, 2017. *Man-on-board AS/RS.* Available at: http://powerpalletinc.com/wp-content/uploads/2015/07/iStock_000008043090Medium.jpg

[Figure 3] "Potentially Volatile Environment Calls for Extra Features on Driverless Forklift Trucks-ForkliftNet.com", Forkliftnet.com, 2017. *Driverless Forklift*
Available at: http://www.forkliftnet.com/news/newsimg/201105090907026424.jpg

[Figure 4] "The next generation of warehouse control systems", D. Adams, Material handling & Logistics conference, 2011, *Different software used in warehouse*

[Figure 5] "Incremental Encoders for GR/G Motors", Dunkermotoren GmbH, *Incremental encoder* RE 30-2-50 Technical Description

[Figure 6] "The Basics of How an Encoder Works", Clearwater Tech, *Incremental Encoder basis.* Available at: http://www.clrwtr.com/How-Encoder-Works.htm

[Figure 7] TECH-FAQ, "Inductive Proximity Sensor", 11 March, 2016. Available at: http://www.tech-faq.com/proximity-sensor.html

[Figure 8] "Inductive Proximity Sensors", pc-control.co.uk 2008. Available at: https://www.pc-control.co.uk/images/proxy.jpg

[Figure 9] "What's the Difference Between Light ON and Dark ON Modes for Photoelectric Sensors?" M. Turner, Pepperl+Fuchs, Jul 16, 2013. . Available at: http://blog.pepperl-fuchs.us/blog/bid/290513/What-s-the-Difference-Between-Light-ON-and-Dark-ON-Modes-for-Photoelectric-Sensors