COVID-19: Simulated robots to pick items due to workforce shortage in the retail sector

Jonathon Carter
School of Computing
University of Portsmouth
Portsmouth, United Kingdom
UP896692@myport.ac.uk

***Abstract*—During COVID-19 many areas of the industry have had to change and evolve in order to adapt with everyday life. The risk of infection from COVID-19 is down to human transmission and therefore minimising human to human interaction is essential. Robots have been modified and created in order to battle these issues in all areas of life.**

***Index Terms*—COVID-19, retail, industry, logistics, robotics, youBot, CoppeliaSim, Lua, colour detection, object sensing, SOMA, AMR, AVG, ASRS, artificial intelligence and machine learning.**

I. Introduction

The retail industry during COVID-19 has had a major impact upon its staff. With the risk of transmission being down to human to human interaction it is inevitable that staff may get infected. With the measures of social distancing and PPE, staff have been able to minimise this [1]. With all of these measures in place staff shortage can arise due to the domino effect of infection.

Specifically during the pandemic online shopping has been a good way of mitigating the spread of the virus due to people not leaving their homes to go and purchase goods. Yet due to the staff shortage that occurs due to the virus it has put a strain on the retail side of the industry as shown in figure 1. Therefore with the aid of robots in the retail sector, the pressure of meeting the consumer demand for online ordering can be met. By using robots to complete staff tasks not only can they fill in due to a workforce shortage but can also reduce the transmission of COVID-19 [2].

This summary will be addressed in this report with technical analysis of the current solutions and literature. Further on will a theorised design and implementation through CoppeliaSim be concluded focusing on this issue.

II. Literature

The focus of this review is to investigate the solutions that are available in the field in addition with updated literature in the area to help in design and developing a solution to the specified problem.

First will be looking into existing implementations that have already been adapted to tackle the areas of concern mentioned in the problem. Then moving onto a proposed solution which can be further developed upon to give this review a concise conclusion to the issue.

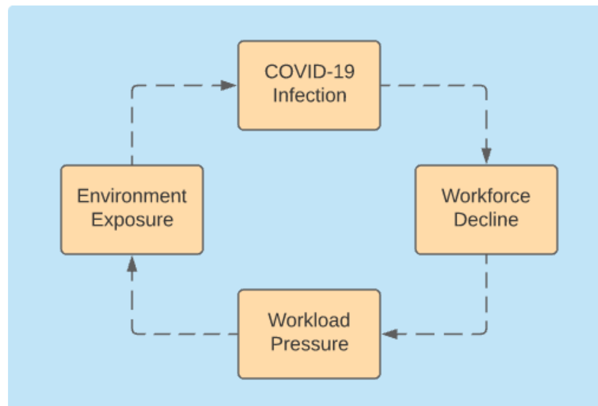

Figure 1. COVID-19 retail cycle of workforce shortage and industry pressure.

A. Existing Examples

There are many areas in retail that use different types of robots for specific types of jobs. As this literature is focusing on a problem with workforce shortage it is important to consider multiple areas and roles that may fall under a workers job description.

Soft manipulation (SOMA) is a type of technology that allows for robots to grasp objects in a natural manner which “reduces uncertainty and reject disturbances during grasping” Deimel and Brock [3]. By using this solution the robot is able to pick up and handle delicate objects such as fruit and vegetables that can be then placed in its desired position while not clasping the object too hard to cause bruising as shown in figure 2. This is done through moving to the desired pose position, moving in a straight line towards the object, grasping the object and moving upwards on the Z-axis. From here the robot is able to move to its second desired position and repeat the steps in a backwards fashion [4].

Autonomous Mobile Robot (AMR) is a type of robot that can independently move through its environment. They use sensors, artificial intelligence and machine learning to plan a path to navigate through the environment [6]. Due to the nature of these AMR’s they are very versatile and can be used for identifying, collecting and moving objects from one position to another whilst avoiding obstacles [7]. AMR’s can be used in the retail environment through assigning a specific object to

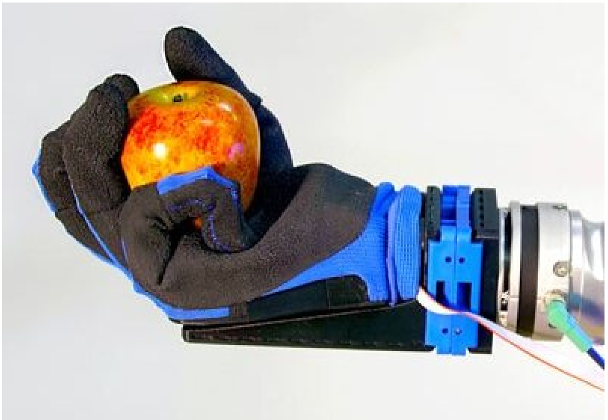

Figure 2. SOMA hand used by Ocado Group [5].

collect (this could be a bulk box or a singular item), retrieving and dropping in the specified drop off point. Whilst all of this being done with the robot's algorithms that plots its course and mitigates obstacles it comes up against as seen in figure 3.

Figure 3. Example AMR robot: "Fetch and Freight" [8].

Automated Guided Vehicles (AVG) is a less advanced version of the AMR. AVG rely completely on transport of goods from point A to B, the route taken can be unidirectional or bidirectional and is usually controlled by "traffic control" style computer/system which regulates flow when more than one vehicle is in the system [9]. A great innovative use for AVG's is the ability to combine it with UV-C disinfection technology. This type of robot is called a UVD or MIT UV robot that when programmed to follow the intended route can make its way around an area or complex disinfecting as it goes. The downside to this is UV-C light is harmful so would need to be done at distance or during the out of work hours [10]. This can be incredibly helpful in the fight against COVID-19 [11].

Automated Storage and Retrieval Systems (ASRS) is a warehouse automation technology. It is solely to buffer, store and fetch items on demand. Similar to the AVG, ASRS uses

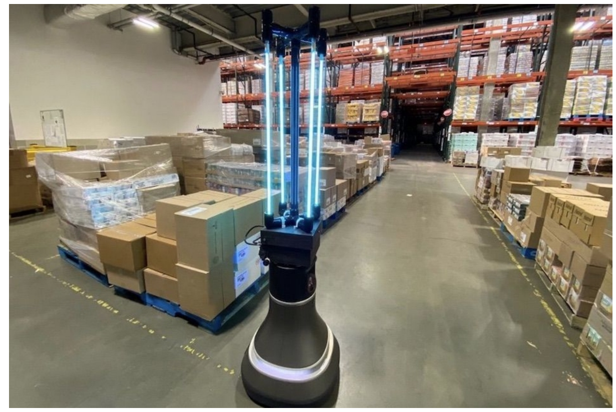

Figure 4. MIT UV robot used in a food bank to disinfect stock [12].

a "traffic control" style computer/system whose sole purpose is to completely manage every aspect of the robot(s) tasks. This can include thousands of metrics of data and provide split precision in its actions on the floor. It has many benefits that can be brought to companies such as an efficient use of floor space, very efficient, accurate and increased yield [13]. A very good current example of this is the Ocado Hive where hundreds of robots move about a grid like system picking items to be further sorted into customers baskets [14] as seen in figure 5.

Figure 5. "The Hive" a large collection of robots sorting goods [15].

By using this type of system human to human transmission of COVID-19 can be mitigated and therefore is a great innovation of technology. By completely automating this process Ocado has used SOMA technology to pack consumers bags to be then transported to the customers door via worker. In addition, in the case of a Goods-to-Person model workers can be spread across the workplace in their own stations packaging items [16]. However due to the efficiency of an ASRS system companies will most likely opt for a complete automation model.

B. Proposed Solution

The issue is clear in requiring items to be collected, the literature explored a range of examples that nicely tie in mobile robots with collection and distribution being focus points. Hence for this solution the KUKA's youBot is a great option to run in a simulated environment. This is due to youBot being very versatile with 5 degrees of freedom on its arm, 2 finger gripper and maximum freedom of movement on its base (x, y, Θ) [17].

Figure 6. KAKU youBot to be used in simulations.

The solution will be modelled inside of CoppeliaSim where the youBot can be completely customised. The youBot has open interfaces meaning adding sensors that are directly programmable will allow it to respond to many obstacles, specifically the ability to distinguish different colours or shapes. Furthermore the flexibility of movement and freedom provided with the arm and base will allow the youBot to diversely complete tasks.

This robot is a good choice to be working with due to the possibilities and customisation it can provide. Research is still being completed to date with the use of the youBot. An important example of this is the use of testing fuzzy logic models focusing around the base of the robot with its complete freedom of (x, y, Θ) [18].

III. Approach

In order to progress on the proposed solution this section will be focusing on the design and development of that. This is where the scenario can be accurately recreated resembling the real world example. Furthermore by doing this effectively the scenario can be thoroughly tested before deploying in the real world situation.

A. Software & Hardware

The simulation environment will be within CoppeliaSim, this will allow for the complete 3D modelling and replica environment to be cloned into the application. The youBot is already well implemented into this software as a model which will allow for quick and concise programming reducing time in getting the proposed solution running to tackle the issue.

The youBot as stated earlier in II-B consists of a multi directional mobile base with a 5 axis robotic arm. This is very agile for the environment that it needs to work in allowing it to move to the needed position and then carry out the set tasks. Sensors that are being attached to the youBot include proximity and vision. This is primarily for:

- Making sure the youBot does not collide with any object or being.
- Allow the youBot to sense what colour object is being handled to select the next task to be completed.

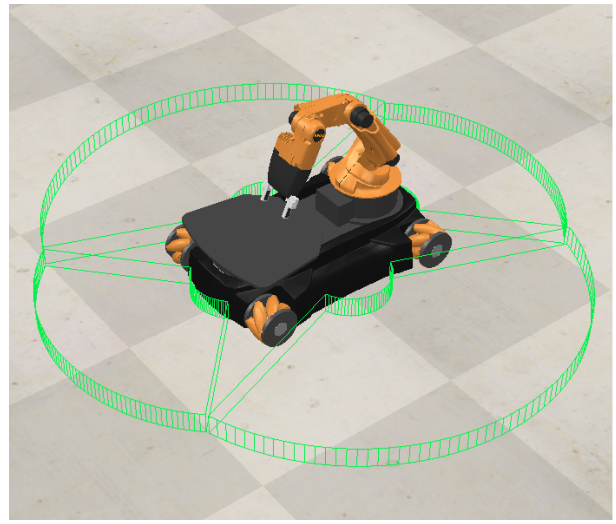

Figure 7. KAKU youBot with proximity sensors attached.

In the environment which will be discussed in the next section III-B there will again be a proximity and vision sensor. These sensors will be primarily for:

- Making sure the objects that are being provided to youBot are in an orderly fashion.
- Providing a visual inspection in the simulation targeting the object's colour before it is handled by the youBot.

B. Environment

Due to retail having two sides to a store, a browsing side for the customer and one side for the storage or solutions. Hence for this scenario the youBot is working on the latter side where items are sorted. The simulation consists of a backroom where objects are brought in via conveyor belt, this is to simulate any items that have been pre-picked by staff on the shop floor.

From here objects are collected from the conveyor belt and distributed into objects basket, this is to help with picking at a later stage. The other task that it needs to perform is collection of three different objects and place them into a sorted order basket. This is then to be processed by staff into the next stage.

As this environment is the storage of the retail shop or building, people will be expected to be walking around. Hence it is important to make sure that the youBot does not collide with a person. However since the area is not a busy zone as it is known to be in use by youBot the severity of a person being collided with is low.

C. Objects, Determinism and Navigation

In order for youBot to know how to handle an object it is necessary to have the vision sensor added. In this simulation

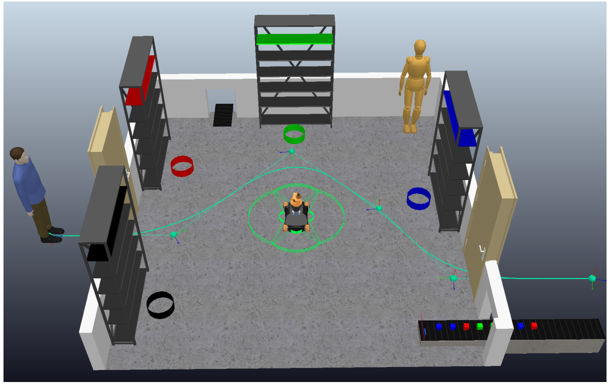

Figure 8. Simulation environment, path for human model to take during the run.

the objects are cubes which are coloured differently, this is to demonstrate objects that require sorting determined by what they look like. Hence the strongest route for the simulation is to have them coloured in a red, green and blue (RGB) approach.

Within CoppeliaSim you are able to use vision sensors to capture an image of what the sensor can see in its path. From this image you can specify how much of the image you want to view all the way down to 1 pixel. This is where you can extract the RGB values on a scale of 0-1, with this you can determine the colour of the object. This is the pseudocode in figure 9 to determine exactly what colour an object is, even down to very similar RGB values an object is able to be sorted. Even if the object is deemed to be greyscale or doesn't fit into the colour bracket specified, it can be sorted.

```
-- where passed image coded {r,g,b}
image = sim.getVisionSensorImage(sensor)

red = image[1]
green = image[2]
blue = image[3]

if (red > green) and (red > blue) then
    -- this is red
elseif (green > red) and (green > blue) then
    -- this is green
elseif (blue > red) and (blue > green) then
    -- this is blue
else
    if (red == blue) and (red ~= green) then
        -- this is classed magenta (red)
    elseif (green == red) and (green ~= blue) then
        -- this is classed yellow (green)
    elseif (blue == green) and (blue ~= red) then
        -- this is classed cyan (blue)
    else
    -- greyscale
end
```

Figure 9. Pseudocode for differentiating RGB values to colour.

After the object type can be determined, it is down to youBot to deposit said object in the correct specified location. In this case youBot plots a course to the dummy that is corresponding to the object type where it is dropped off and the process starts again until the conveyor has been sorted correctly.

The youBot that has proximity sensors added to it will make sure that it does not collide with any people. With the sensors they have been added in a 360° manner to give the maximum coverage around the base. This is to guarantee that any people that come into the path of youBot cause it to stop, wait for a certain amount of time, check if it's clear and carry on (as seen in figure 10).

```
-- get information from sensor
-- x , y not used however are needed to get objectName
objectPresent, x, y, objectName = sim.readProximitySensor(sensor)

-- if objectName is not equal to nil (it is detected)
if (objectName ~= 'nil') then
    if (objectPresent > 0) and (sim.getObjectName(objectName) == 'human') then
        -- halt youBot and wait till human is out of the way
    else
        -- ignore as detection it is not an issue
    end
end
```

Figure 10. Pseudocode for differentiating human detection and halting.

The simulation calculates the first point, that being the robots current position and then the second point which is the target position. From here the robot is able to navigate effectively and quickly to its destination. Effectively point to point is a path based system where the robot travels along the straight line to that new position (figure 11).

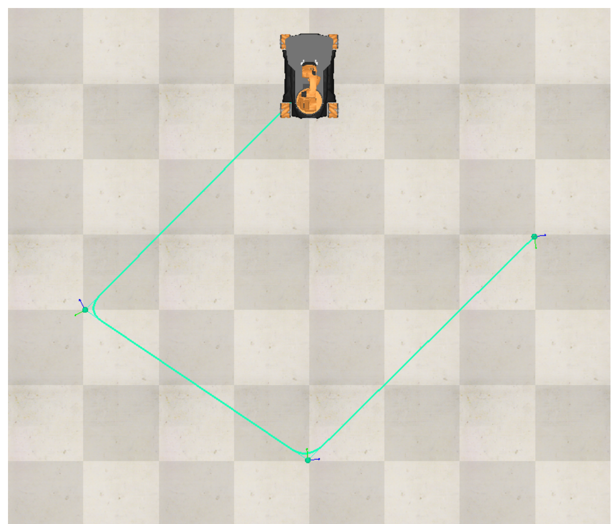

Figure 11. Example of youBots path within the simulation, stopping off at each point.

D. A Detection Method In Industry

Industry is heavily using artificial intelligence and machine learning to determine the environment around a robot as previously mentioned in section II-A. For example Tesla is developing and constantly testing their autopilot for autonomous driving where the user would not have to control the vehicle unless necessary. By using artificial intelligence and machine learning the vehicle is able to differentiate between vehicles and pedestrians, furthermore the vehicle is able to alert its driver of obstacles or dangers in the blind zones due to the cameras and sensors in those areas [19].

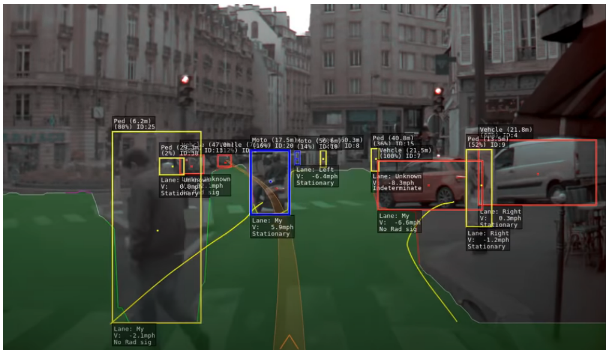

Figure 12. Paris streets in the eyes of Tesla Autopilot [20]

IV. Results Analysis

With the simulation made to meet the requirements of the proposed solution it is imperative that some testing to make sure some objectives for the solution are met. This testing provides an accurate analysis which can then be further relied on if taking the solution further into a physical form. Along with this is the basis of how the simulated flow steps are worked through which will be explained further in this report IV-D.

A. Testing Objectives

The following objectives were curated to make sure the proposed solution can be applied quickly into physical form, whilst meeting requirements and safety precautions.

- Collision: youBot detecting a human must stop immediately.
- Collection: youBot should get a signal to collect objects, go to the location and store them temporarily on the robot.
- Delivery: youBot should determine the colour of the object and deliver to the objects destination.

B. Testing Results

By determining the position of the object and then having the data handled to the youBot to move to it allows it to be self-sufficient when being left to itself. For this the example in figure 13 is a successful console log of this data being sent to youBot. After youBot has loaded its objects onto the

```
"Obj:","OAHV7","Position:",{1.6118005514145, -2.0550787448883, 0.16245968639851}
"Obj:","M6GV6","Position:",{1.6095445156097, -2.0551269054413, 0.16246253252029}
"Obj:","91YV4","Position:",{1.6097540855408, -2.055055141449, 0.16246224939823}
```

Figure 13. Console log, Read: Obj,{Name},Postion,{X,Y,Z}

temporary storage it picks up the first object, measures its colour. From there it can determine where to travel to and deliver the object. The output in figures 14 shows a successful decision based on the colour of the object it is viewing. If youBot detects a grayscale object it is determined that this object cannot be sorted into that of the other colours and so is placed on a conveyor to take to its destination. This figure 15 shows that the robot is able to determine that the item is of

```
"Obj's loaded on youBot:",{"OAHV7", "M6GV6", "91YV4"}

"Obj:","OAHV7","Colour Type:","blue"
"Obj:","M6GV6","Colour Type:","blue"
"Obj:","91YV4","Colour Type:","red"
```

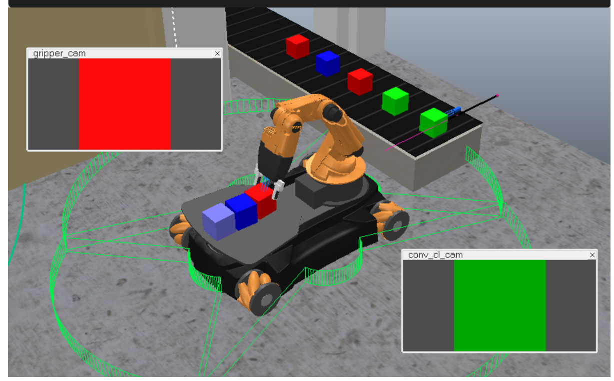

Figure 14. Console log and simulation image showing colour recognition test. Gripper cam and conveyor cam showing the objects infront of them.

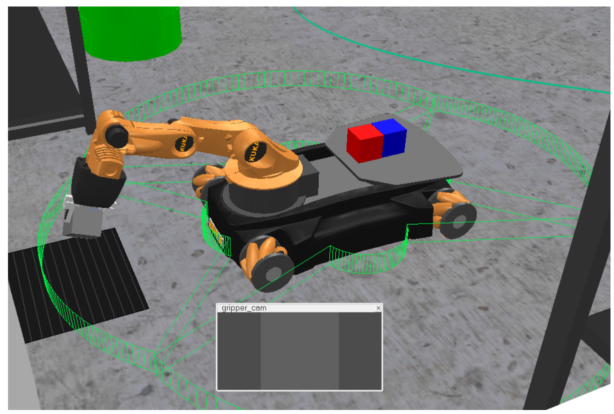

Figure 15. Simulation image showing greyscale object being dropped off to exit conveyor in the test.

the greyscale type and places it in that location. Furthermore youBot is also programmed to be specifically told if it should sort items into one area for all colours, this is helpful for employees to quickly get similar items but not of the same type. Figure 16 shows this being a success. The final test for youBot was to automatically halt when detecting a human object. This has been done successfully as shown in figure 17.

C. Testing Summary

Test Objective	Sim 1	Sim 2	Sim 3	Overall
Position of object	pass	pass	pass	100%
Colour measurement	pass	pass	pass	100%
Greyscale measurement	pass	pass	pass	100%
Specific object delivery	pass	pass	pass	100%
Human collision avoidance	pass	pass	pass	100%

D. Simulation Flow

In this section is a quick overview of the decisions made for the simulation to work effectively and efficiently as shown in

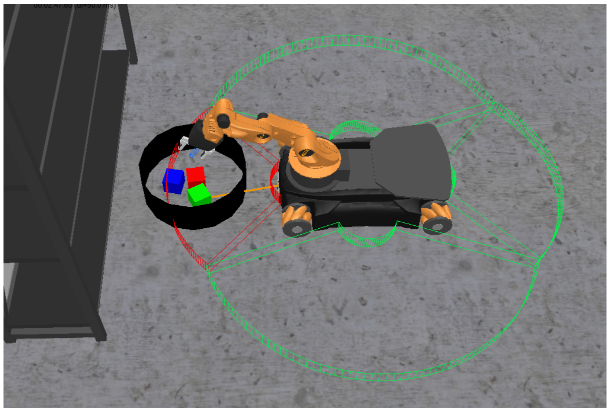

Figure 16. Simulation image showing different RGB objects sorted into the same basket during the test.

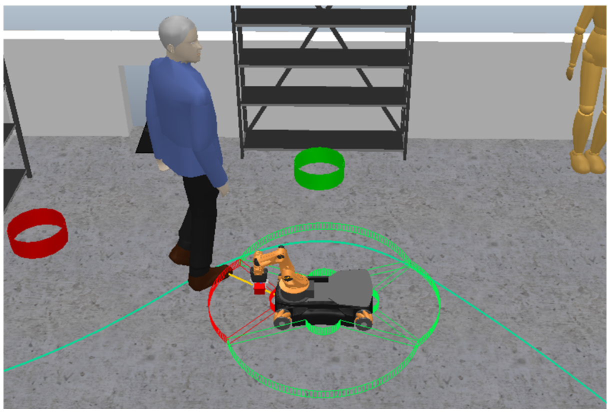

Figure 17. Simulation image showing youBot detecting and halting when a human is present in its proximity sensing range.

the flowchart figure 18. This gives a more of an understanding behind the simulation and for a user to determine if this proposed solution fits their own issue, which if it does it can be taken into consideration for their own project.

V. Discussion and Conclusion

A. Performance

The overall solutions performance was a great success, with the youBot being able to deliver all of the given objects as mentioned in section IV. The other objectives of no collision, sorting by colour type and sorting as one have been met. There were minor errors seen which do vary simulation to simulation however these are mentioned in the following issue section V-B.

B. Issues

Due to the object sizes, when storing them on the youBot platform the blocks may get clipped slightly during collection and drop off. This means that there is a small percentage chance that the youBot is unable to collect the item off of the platform which may result in the object being dropped from the youBot. This can be mitigated by precisely calculating the exact angle that the object is needed to be picked up

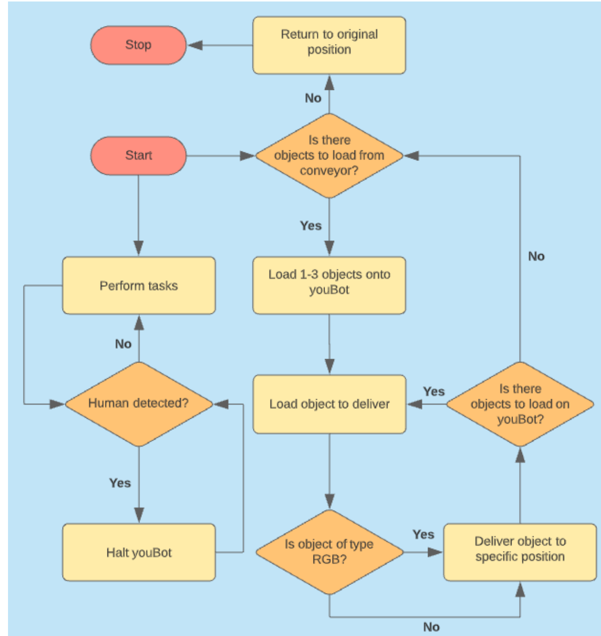

Figure 18. Flowchart of the simulation.

and placed. In a real world situation a gripper that is more adaptable to picking up object variations.

Another issue faced in the simulation was the detection of what is human and what is not human. Through programming and use of the CoppeliaSim API the different objects were able to be filtered due to the baskets being “clippable”. This would not be the case in a real world situation.

C. Improvements

Though with the success of the simulated solution improvements can still be made. With the use of CoppeliaSim youBot is an obvious choice due to the flexibility and integration that it has in CoppeliaSim whilst also being a heavily educational background. Therefore a system of two or more “Fetch and Freight” robots as shown in section II-A figure 3 may be more efficient than a single youBot carrying out the sorting.

This current solution runs off a point to point example. By implementing fuzzy logic into the solution the youBot would be able to navigate around “unknown” environments freely and avoid collisions by deciding a path around the obstacle rather than halting completely [21]. Fuzzy logic is based upon degrees of truth rather than boolean true or false. Hence allowing for more decision making and learning.

Lastly, through use of the camera in this solution, objects were able to be determined through colour. In a real world scenario users may want the robot to understand what it is handling. UHF-RFID tagging can be done to track the whereabouts of items [22]. This could also let the robot determine how hard a grip/pressure to use when collecting the item which in turn could allow the robot to handle soft items as mentioned in section II-A.

D. Conclusion

In summary the solution was a success, it closed the issues that the problem outlined and performed well in simulation. This report has a mixture of literature that is relevant in combating the problem whilst providing insight information to aid in factors of the proposed solution. The youBot performed the objectives that were needed and passed testing successfully meaning the simulation can be applied to a real world concept. Programming hurdles were met along the way with solutions and improvement suggestions. This report adds a perspective upon the research and so with confidence can be a contributing factor to future work in this area.

References

[1] L. Morawska, J. W. Tang, W. Bahnfleth *et al.*, 'How can airborne transmission of covid-19 indoors be minimised?' *Environment international*, vol. 142, p. 105 832, 2020.

[2] S. Singh, V. K. Dalla and A. Shrivastava, 'Combating covid-19: Study of robotic solutions for covid-19,' in *AIP Conference Proceedings*, AIP Publishing LLC, vol. 2341, 2021, p. 020 042.

[3] R. Deimel and O. Brock, 'Soft hands for reliable grasping strategies,' in *Soft Robotics*, Springer, 2015, pp. 211–221.

[4] M. F. Martins, P. Sotiropoulos, M. Kowalik and G. Deacon, 'Systematic evaluation of compliant under-actuated soft manipulators in an industrial context-the ocado use case,' 2016. [Online]. Available: http://clem.dii.unisi.it/~malvezzi/graspquality/wp-content/uploads/2016/05/final_06-MM_poster.pdf.

[5] O. Group, *How ocado and soma consortium partners made major breakthroughs in robotic manipulation*, 2019. [Online]. Available: https://www.ocadogroup.com/technology/blog/how-ocado-and-soma-consortium-partners-made-major-breakthroughs-robotic.

[6] N. Sariff and N. Buniyamin, 'An overview of autonomous mobile robot path planning algorithms,' in *2006 4th student conference on research and development*, IEEE, 2006, pp. 183–188.

[7] Intel, *Autonomous mobile robot (amr) overview: Types and use cases-intel*, Sep. 2021. [Online]. Available: https://www.intel.com/content/www/us/en/robotics/autonomous-mobile-robots/overview.html.

[8] E. Ackerman, *Fetch robotics introduces fetch and freight: Your warehouse is now automated*, Apr. 2015. [Online]. Available: https://spectrum.ieee.org/fetch-robotics-introduces-fetch-and-freight-your-warehouse-is-now-automated.

[9] R. Gaskin and J. Tanchoco, 'Flow path design for automated guided vehicle system,' *International Journal of Production Research*, vol. 25, no. 5, pp. 667–676, 1987.

[10] S. Kurn, *Uvd robots*, May 2019. [Online]. Available: https://www.sychem.co.uk/infection-control/uvd-robots/.

[11] Y. Shen, D. Guo, F. Long *et al.*, 'Robots under covid-19 pandemic: A comprehensive survey,' *Ieee Access*, 2020.

[12] R. Gordon, *Csail robot disinfects greater boston food bank*, Jun. 2020. [Online]. Available: https://news.mit.edu/2020/csail-robot-disinfects-greater-boston-food-bank-covid-19-0629.

[13] K. J. Roodbergen and I. F. Vis, 'A survey of literature on automated storage and retrieval systems,' *European journal of operational research*, vol. 194, no. 2, pp. 343–362, 2009.

[14] T. Scott, *How many robots does it take to run a grocery store?* Jul. 2021. [Online]. Available: https://www.youtube.com/watch?v=ssZ_8cqfBlE.

[15] O. Group, *Technology behind the ocado smart platform*, 2020. [Online]. Available: https://www.ocadogroup.com/technology/powering-osp.

[16] B. Welsman, 'Automated order picking systems: Bringing the goods to the person,' *MHD Supply chain solutions*, vol. 40, no. 2, pp. 32–36, 2010.

[17] Kuka, *KUKA youBot Research & Application Development in Mobile Robotics*. 2010. [Online]. Available: https://www.generationrobots.com/img/Kuka-YouBot-Technical-Specs.pdf.

[18] F. Valdepeña, D. M. Vargas, A. Luna-Alvarez and M. Á. Ruíz-Jaimes, 'Navigation of an omnidirectional robot based on takagi-sugeno logic model type-i,' in *2021 Mexican International Conference on Computer Science (ENC)*, IEEE, 2021, pp. 1–7.

[19] S. Ingle and M. Phute, 'Tesla autopilot: Semi autonomous driving, an uptick for future autonomy,' *International Research Journal of Engineering and Technology*, vol. 3, no. 9, pp. 369–372, 2016.

[20] greentheonly. 'Paris streets in the eyes of tesla autopilot.' (Sep. 2018), [Online]. Available: https://www.youtube.com/watch?v=_1MHGUC_BzQ.

[21] M. Faisal, R. Hedjar, M. Al Sulaiman and K. Al-Mutib, 'Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment,' *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 37, 2013.

[22] F. Shamsfakhr, L. Palopoli, D. Fontanelli, A. Motroni and A. Buffi, 'Robot localisation using uhf-rfid tags for industrial iot applications,' in *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*, IEEE, 2020, pp. 659–664.