# DevB Bot: An implementation of an automous delivery robot for restaurant setting.

1st Keith. M. Machina
*Computer Science Department*
*name of organization*
Kent, USA
kmachina@kent.edu

2nd Derron Ward
*Computer Science Department*
*Kent State University*
Kent, Ohio
dward34@kent.edu

***Abstract*—The field of robotics is growing tremendously and tons of innovative applications are being witnessed everyday. Some of the eye catching applications of robotics that have transformed the landscape include: rescue robot, customer service, ploughing robots in the field of agriculture, tele-surgery robots in medicine, packing and unpacking robots in warehouses and many more applications. One of the key topics that arises at the mention of robotocs is motion planning which is defined as, an approach to determining the best and most efficient path to a target or a goal state. Achieving efficient motion planning is a debatable topic as there's no specific way to do it. Some of the multiple strategies to motion planning techniques that have been suggested in the past include: use of computer vision, reinforcement learning, neural networks, inverse kinematics just to mention a few. This paper will discuss delivery robots and go in depth to unravel details, such as navigation, about implementation of a delivery robot in a restaurant setting.**

## I. Introduction

Delivery robot as an application in robotics has greatly transformed the landscape of delivery. Over the years, there's been a lot of human intervention but now such tasks are being taken over by robots. Robots are now able to get items to the desired customer with ease. Automaton is key issue that arises in robotics as most of the robots are still tele-operated, a good example being tele-surgery where there has to be expert on the other end performing the operation. Automation is quite broad and a contextual issue with multiple solutions to it.

In ROS, move base API is suggested as a means to efficient navigation that happens on the go as the robot is moving. At the heart of move base API is Action API that provides implementation of simple actions through the help of multiple topics that perform various navigation activities. For example, the topic, move base simple goal is a topic that is specifically used to publish a navigation goal. Other topics within action API include move base/status that provides info about set goal being achieved or not, move base/feedback containing information about a current position, move base/result and many others that contribute to autonomy when used. It is worth noting that there are multiple strategies to achieving automation such as sensor based strategies that achieves automation through use of sensors, rule-based strategies which involves applying multiple rules, learning strategies that involve using of techniques such as machine learning, reinforcement learning just to mention a few.

Automation being a key component in robotics, this paper will discuss the details about implementing an automatic delivery robot within a restaurant setup.

## II. Related Works

A major issue in robotics application at the moment is lack of automation capabilities therefore developers and researchers are trying to come up new automation. It is foreseen that automation will revolutionaize the world in many ways. In [1] researchers claim that use of automatic delivery robots has a domino effect on issues such as carbon emission. Since autonomous delivery robots are electric, carbon emission will drastically reduce over the years as there's a a lot of carbon being emitted in the process of delivery.

## III. Methodology

### *A. System Overview.*

The system comprised of the following pieces: turtle bot burger, an graphical interface, and an object detection model. The section below will go over each component and what role was played.

*1) Turtle Bot Burger.:* The turtle bot burger is a 3 layered robot comprising of various component. First layer of the turtle bot held together the wheels of the robot as well as housing the battery. Second layer of the turtle bot houses the open CR board of the burger. The open CR is considered as the "brain" of the robot. Third layer houses the raspberry pie. The raspberry pie contains various ports such as the HDMI and USB meant for different purposes. Through the raspberry pie HDMI port, connecting to a monitor and displaying is made possible. The fourth layer of the burger robot is home to the Lidar sensor, which stands for Light Detection and Ranging. The lidar sensor seeks to find the distance to an object by constantly emitting a laser beams, waits, and measures the time it takes to get the light reflected back. This enables the sensor generate maps since obstacles within the space are recognised and mapped. Through the Lidar sensor, 2D and 3D images can be generated. Attached camera on the robot was to aid real time object detection.


Fig. 1. Turtlebot robot


Fig. 2. QT interface

*2) Object Detection model.:* In order to implement autonomous capabilities object detection had to be implemented. This step involved a series of steps, namely: data collection, model training and model deployment. In the data collection step, all objects that the robot would encounter in the test bed had collected in form of images. The step would be to annotate the images, which is a step that involves putting a bounding box round the object of interest. Essentially what happens is that the position of these bounding boxes will be used during model training to determine where the object is located. Annotation of these images was done with the help of roboflow software, an open source tool. It was also at this step that image pre-prossessing took place. Pre-prossessing is an imperative step in object detection that involves making adjustments to the images in regards to what would be experienced in a real world scenario. A good example is occlusion which purposes to block a certain section of the image to enable the model still recognise the object even when not fully visible. Some of the pre-processing steps that were applied on to the annotated data include: auto-orient, resizing the images to a standard size of 224X224, augmentation such as horizontal flip, rotation of both +39 degrees and -39 degrees was applied. All these was applied to the model to make it perform better when encountering real world objects.

*3) Move Base API.:* Move base is an application programming interface that enables autonomous navigation capabilities in a robot. It contains inbuilt algorithms that enable the robot to perform path planning given a navigation goal. During these experiment, move base / simple goal topic was used to publish destinations i.e. where the turtle bot needed to move to. In a nutshell, in this experiment, anything that required navigation was performed using move base ROS API. It is important to note that a publisher subscriber model was used in all ROS operariions.

*4) Graphical User Interface.:* The graphical user interface was an interface built to enable customers place an order that is directed to the respective entity such as a chef. Since move base api in ros was being used, the specified locations were captured and hard coded into the application such that when a button on the GUI is keyed in, it directs the robot to the specified location. The GUI was built using rqt ros package. It contained elements such as buttons and text view.

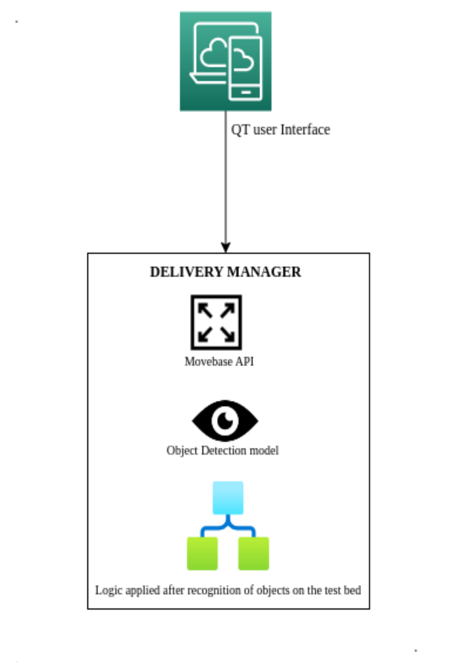


Fig. 3. System overview: Delivery manager.

The diagram below clearly depicts the system overview and sheds some light on the inner workings. The overview is focused on the ”heart” of the application which is the delivery manager.

## *B. Setup.*

This experiment simulated a restaurant setting where there are customers and the chefs making the food. The whole restaurant experience that was being simulated in this ex-

Fig. 4. The actual test bed

periment was one with no human intervention. Basically, a human would walk into a restaurant, place and order on the robot's graphical user interface and normally this order would be directed to a specific chef depending on the type of food ordered. In this particular setup, there were 3 designated points where a customer could order from. As for the chefs, there were two locations marked for the two chefs. There also traffic lights deployed on the test bed that were put on purpose to signal when the food is ready. This was meant to give the robot a heads up that it could go ahead to deliver the food to a customer. Once the robot was done delivering the food to the customer, a success message would be published and the robot would return home if there was no other in the queue that needed to fulfilled. Below is a photo of how the test bed looked like.

### *C. Implementation.*

To implement this, the first step was to perform mapping. This step was imperative to enable the robot understand its environment. It was in this process that the coordinates of the important points, such chef A,on the map were captured. Gmapping and hector mapping algorithms were used due to their high precision

Once the map was ready, next was to configure move base. This was done by loading the map in rviz, a visualization software, and using move base / simple goal to publish a navigation goal. An IR sensor, which was used to detect loading and unloading, was embedded on to the turtle bot by connecting it on to the opencr board.

The QT interface was designed, and during this step, for each button e.g chef A, the relevant coordinates, where robot should move when the button is tapped, were embedded.

Another key piece of the solution, which was regarded to as the heart of the application, was a delivery manager python script that was meant to handle all the logic regarding deliveries. For example queuing of deliveries on first come first serve basis and getting rid of the fulfilled orders from the queue. Embedding coordinates captured from the mapping process and publishing of goal states all were done on this management script. Publishing of goals states was made possible by use of move base simple/goal topic. It's important to that a voice

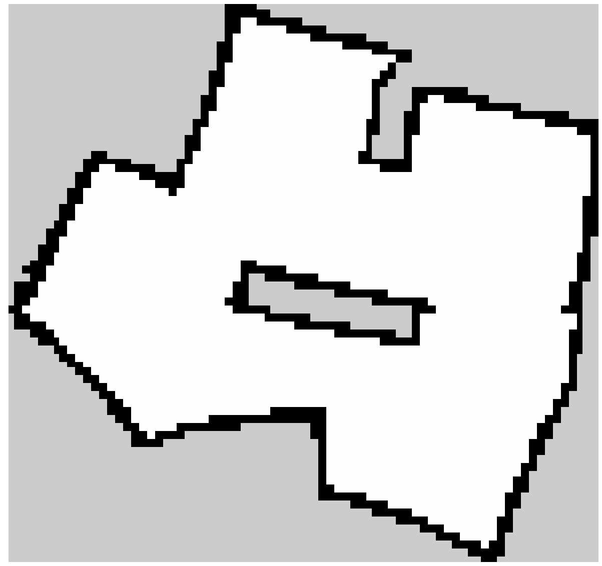

Fig. 5. A map of the test bed implemented captured using hector algorithm.

feature was implemented which was invoked every single time, the robot got to a goal state.

The last piece of the puzzle was the object detection that was done using YOLO where the data was annotated using roboflow software. Once the yolo model was trained and tested locally, it was deployed on to the turtle bot. To accomplish this, the weights of the model resulting from the training process were obtained and added on to custom detect python script which is responsible for outputting the labels of detected objects and the corresponding confidence level of that prediction. The resulting predicted labels were then published to a topic that would later be listened to. Logic on what to do after an object has been detected, for example, a stop sign, was implemented on the delivery manager script by listening to topic publishing the labels and having an If statement actualize the logic.

## IV. RESULTS

The robot moved to the locations after receiving instructions from the qt interface however, move base navigation struggled a little bit with path planning as it kept lingering around one spot trying to find the optimal path.

## V. IMPLEMENTATION CHALLENGES

- Getting the IR sensor to work with open CR was a challenge. Sometime it would detect object loaded and sometimes it would not however, after experimenting with a few IR sensors, a more stable one was found.
- Deploying the yolo model is quite problematic as it was not directly compatible with turtle bot. A few scripts had to be modified to get it to work.
- Move base was sometimes unreliable as it would take too long for the robot to figure out a path something that caused the robot to linger around one spot for long.

- It was challenging to implement auto race package on the turtle bot led to issues such as, missing libraries and environment set up issues.

## VI. Learning

- We have a better understanding of how robotics work.
- We have more experience than ever in hardware.
- We gained troubleshooting skills.

## VII. Bonus Points

- Expected Grade: A. We implemented an extra voice feature in our solution.
- Suggestions for SDR Spring 2024: Document all expected hardware issues and their correspending solutions.

## VIII. Conclusion

Automation being a key concern in robotics during this age and era, the writers of this report purposed to implement an autonomous delivery robot that is meant to operate in a restaurant setting. To implement autonomy, a couple of strategies were implemented, namely: computer vision and move base API. Computer vision, object detection using YOLO to be precise, was implemented to enable the robot perceive objects around it as humans do. This would help in building logic for what needs done when certain objects are detected. The paper also reports on the model that was used to make the project a success, which is a publisher subscriber model. This was the go to model due to its simplicity and ease of use. In a nut shell, the paper went over an approach that was used to build an autonomous delivery robot as well as clearly bringing out the issue that came along with that implementation.

In future the team looks forward to implementing flying delivery robots that are fully autonomous, can travel over long distances, and resistant to harsh weather changes.


## Acknowledgment

Thanks to the Robotics professor, Dr. J.Kim from Advanced Tele Robotics Lab for the guidance and support through the robotics class.


## References


[1] Figliozzi, Miguel  Jennings, Dylan. (2020). Autonomous delivery robots and their potential impacts on urban freight energy consumption and emissions. Transportation Research Procedia. 46. 21-28.

[2] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[3] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[4] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[5] K. Elissa, “Title of paper if known,” unpublished.

[6] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[7] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[8] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.