**AUTONOMOUS INDOOR FOOD DELIVERY AGENT**

Team Members: Nishqa Sharma, Ignacio Ampuero Gonzalez, Belen Castellote Lopez

TA: Weihang Liang

**1. INTRODUCTION**

**1.1 Objective**

**(a) Problem Statement**

The ECE Building at UIUC is home to one of the strongest ECE Departments in the United States. The students, faculty and staff at UIUC are extremely busy, working on research, teaching, or studying. However, everyone needs food, getting which can often be a time-consuming source of disturbance or a break of concentration. Placing orders on apps such as DoorDash and Grubhub can be expensive, especially for the students, since these apps charge about $4-$8 [1] per order on an average. Moreover, tracking or waiting for an order from an external restaurant can also be a source of disturbance. Hence, our goal is to build an autonomous food delivery agent, which can take customer orders, accept payment, and deliver food within the ECE Building, allowing for substantially cheaper, quick and hassle-free delivery from The Daily Byte Café, located in the ECE Building.

**(b) Solution**

Our solution is to build an autonomous food delivery agent with a metal body, with separate compartments for hot and cold food, which can accept orders from customers using a drop down of the daily byte menu (with updated item availability) using a virtual, remote user interface application, accept payments to pass on to the Daily Byte, pick up the food to the Daily Byte, and deliver the food to a room number in the ECE Building specified by the user whilst placing the order, which will be delivered for a very minimal delivery fee.

**1.2 Background**

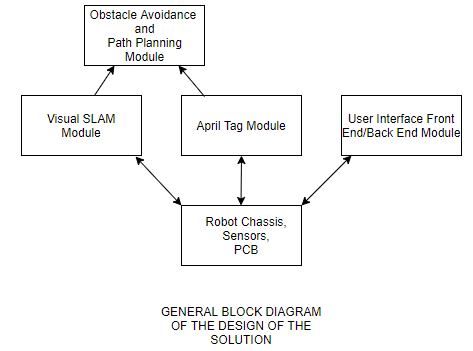
There are currently several delivery robots that are being deployed indoors, by companies such as Bear Robotics [2], Savioke [3], and Vesta [4]. We are going for similar functionality; and we aim to extend the capabilities of these robots in the extent of the ECE Building by providing insulated compartments to ensure the desired temperature, quality and freshness of the delivered food items (which only the Savioke robots have at the moment; the others only have trays which leave food open to the air), by drastically cutting down the costs that would be incurred in leasing such robots for the ECE Building, with Savioke robots (which are very similar to our robot) costing $2,000 per month for leasing [5], whereas ours will cost around $200 each, as a one time fee (and a small monthly cost for electricity to recharge the batteries and operate the back end system, as well as associated minimal labor costs), and by integrating a complete front end and back end user interface with the body of the agent, which provides for a smooth and comprehensive user shopping experience, along with nearly free, speedy food delivery (which is again only being done by Savioke at the moment, with the user interface only present on the body of the robot; orders cannot be placed remotely virtually (the user would have to make a phone call/interact with a staff member), as per the robot descriptions on the Savioke website.)

**1.3 High Level Requirements**

Our agent will be judged by the following criteria:

* It must be able to update availability correctly most of the time, based on the updates performed by the Daily Byte Staff, as well as process payments and take orders correctly most of the time. Errors will be tolerated in case of returned or enqueued items, or slow updates due to slow internet speeds, etc. We are yet to develop a numerical metric for this, since it will depend on the design of the actual user interface system, that we are currently in the process of discussing with the Daily Byte Café.
* It must be able to navigate back and forth between the “home base” (the robot’s resting place in the ECE Building Lobby, the Daily Byte Café, and the rooms of the ECE Building, and must avoid atleast 70% of the obstacles correctly. As for the other 30%, the agent must NOT collide into the obstacles, but rather stop in it’s tracks and take more time to evaluate what to do in that situation (turn left or right, back away, wait for the obstacle to pass, etc.) In order to assess this, we will look at the live demo, as well as plot data for several test runs of the robot.
* It must stop within 2 feet of the door of the room specified by the user whilst placing the order.

**2. DESIGN**

**2.1 Block Diagram and Module Explanation**

The “Robot Chassis, Sensors, PCB” module has been explained in the next section; but in brief, this module pertains to all the physical components of the agent, namely the robot body.

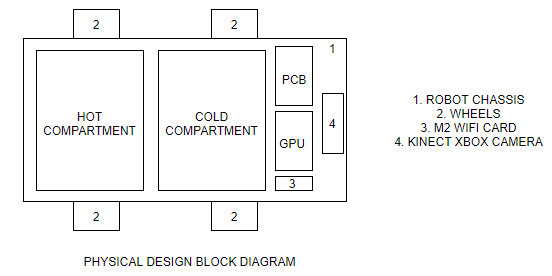
The robot will make use of a Kinect Xbox video camera to implement Visual SLAM and April Tag recognition, which will be used to localize the robot and map the surroundings. To implement these, we will be making use of the following ROS libraries:

* “Slam tool box” for performing Visual SLAM
* “Apriltags\_ros” to perform bundle calibration and video stream tag detection

Once our localization and mapping is done, we will use our maps to implement the Obstacle Avoidance and Path Planning Module via the following libraries:

* "global\_planner" for global path planning
* "teb\_local\_planner" for optimizing the robot's trajectory with respect to trajectory execution time, separation from obstacles and compliance with kinodynamic constraints at runtime
* “locomotor”, which will “provide a mechanism to for controlling what happens when the global and local planners succeed and fail."

Our user interface (front end and back end) will be written in Python/C++. The back end will be updated by the Daily Byte staff, as items are sold over the course of the day, and the front end will be used by the user to place the order as specified in section 1.1 (b). The body of the agent, or the “Robot Chassis, Sensors, PCB” Module will be used to actually navigate the ECE Building and transport the food. In combination, these modules will be able to meet all the functionality needs detailed in section 1.1 (b). All the individual modules in the above block diagram can be developed separately and put back together; however, the “Obstacle Avoidance and Path Planning Module” will need data from the “Visual SLAM Module” and “April Tag Module” during operation.

**2.2 Physical Design and Explanation**

We plan to have a robot body of a 1’ x 1’ base, and 0.5’ in height, made of Aluminium, and lined with double reflective insulation material (used commonly by almost all restaurants in the USA to maintain food temperature). Also, Aluminium is a bad conductor of heat, and will help preserve the food temperature as a result. Each compartment will have a separate hinged lid. There will be 4 wheels, which will be used to drive the vehicle using differential drive. The PCB will consist of the electronic circuit, the power systems (Li 12-V batteries), and the motors (370 brushless motors, which will be connected to the wheels, but controlled by the electronic circuit on the PCB). ‘4’ in the diagram above represents the Kinect Xbox Video Camera, which will generate a video stream data, which will be transmitted to the laptop via the Jetson Nano Single Board Computer, indicated by “GPU” in the above figure, using ‘3’, the M2 wifi card.

**2.3 Functional Overview and Interface**

**(a) The April Tag Module**

This module will use the “Apriltag\_ros” library to calibrate April Tags, which we will stick at strategic locations around the ECE Building, and also to recognize the April Tags from video data stream sent from the Kinect Camera via the GPU. This will help localize the robot with a very high accuracy. This will help make path planning decisions, which will further help navigate the robot in real time.

**(b) The Visual Slam Module**

This module will also use the data sent from the Kinect Xbox camera to form a map of the ECE Building. This will be done using the “Slam tool box” ROS library, which will create a 3D map of the ECE Building, which will then be used to plan the path and avoid obstacles in the planned path, as detailed in 2.3 (d), which will further help the robot navigate in real time.

**(c) The user interface Module**

This primary software module that the robot chassis will connect to will have two components, namely the front end and the back end. The backed of this module will be updated by the Daily Byte staff to record item availability, as well as to collect payment and orders. The front end of this interface will be used by the user to place the order, the room number and make the payment via PayPal/credit card. Once the order has been placed, this module will activate the camera sensor on the chassis, which will start collecting the data and transmitting the video data stream to the two aforementioned software modules, namely “April Tag Module” and “Visual SLAM Module”.

**(d) The Path Planning and Obstacle Avoidance Module**

This module will use the libraries listed in section 2.1 to plan two paths, global, and local—and also to specify what to do if either of the paths don’t work out. Once the paths have been planned, this module will send signals back to the chassis of the robot via the wifi, which will then use the PCB to turn the wheels of the robot as required to navigate it.

**(e) Robot Chassis, Sensors, PCB Module**

This module contains the physical body of the robot. The PCB component of the board, detailed in section 2.2 of this document will have an on and off switch. When the agent is turned on, it will be connected via wifi to the aforementioned software modules, which will be written with the assistance of the ROS libraries. Once the software modules plan the path, this module will make use of the electronic circuit to guide the chassis to the relevant locations and positions around the ECE Building.

We will be making use of the Kinect Xbox camera as our main sensor, which will collect the video data to be used by the software modules, and also two temperature sensors, to ensure correct placement of the hot and cold food.

**2.4 Block Requirements**

**(a) Robot Chassis, Sensors, PCB Module**

This module will be required to collect data using the camera and transmit it at an approximate rate of 1.4 GHz using the wifi card and Jetson Nano Single Board Computer. It is also required to receive signals from the obstacles and path planning module and navigate the vehicle by avoiding 70% of the obstacles in motion, and using the temporary stop option to gather more time to avoid the remaining 30% of the obstacles. There will be tolerance for colliding with fast moving or sudden obstacles, and a measure for the exact requirements will be decided along the design process in the next 2-3 weeks, by the time the design document is turned in, since this amount will be based on upcoming design decisions, such as how precise we will be able to make our code, how fast our robot body will be able to turn, etc.

**(b) User Interface Module**

It must be able to update availability correctly 70% of the time, based on the updates performed by the Daily Byte Staff, as well as process payments and take orders correctly 70% of the time.

**(c) April Tag Module**

This module must track the April Tag locations which are programmed into the system using the library, and use those to correctly localize the robot with a linear error of 2 cm from their actual location, and an orientation error of about 5 degrees [6].

**(d) The Visual SLAM Module**

This module must create a 3-D map of the ECE Building. The accuracy will be determined in the design document phase of the project, after we start working with the “Visual SLAM toolbox” library.

**(e) The Obstacle Avoidance and Path Planning Library**

In addition to the obstacle avoidance described in section 2.4 (a) of this document, this module must plan a path such that the agent stops within 2 feet of the door of the room specified by the user whilst placing the order.

**2.5 Risk Analysis**

The weakest link in this design is the Obstacle Avoidance and Path Planning Module. This is because it is not self dependent in operation; it requires data from the visual SLAM Module and the April Tags Module. Moreover, it needs to process the constant data that it will get from the GPU. In addition to that, it also needs to plan the path both locally and globally, specify what to do in case one fails, which can be a tough decision to make, and also avoid dynamic obstacles on the fly. Hence, it is central to the project, heavy in computation and dependent on other modules for functioning. Hence, it is the biggest risk to the success of the project.

**3. ETHICS AND SAFETY**

Since our agent is going to be carrying items from the Daily Byte, it will most likely carry liquid beverages. We will have to make sure that those do not come in contact, with the electrical components, which could cause short circuits, which would not be significant enough to hurt anyone, but would cause the robot to malfunction, or potentially stop working completely until the circuit were repaired. To prevent this, we will add an extra layer of casing around the electronic components present on the chassis, compliant with the IP67 regulatory standard.

We will also have to make sure that the navigation is accurate, so that no people or property present in the ECE Building are hurt by the body of the navigating robot. This is in alignment with section 1.2 of the ACM Code of Ethics [7]. We intend to do this by adding temporary stops and emergency stops, so that if the robot is unable to assess where to turn in a situation, it will come to a halt rather than collide with a person or property.

Although we will take full care to only use the video stream data for the purposes of navigation, it is possible that this data could be intercepted and used maliciously over the wifi for gathering private data from the ECE Building. This would go against sections 1.1, 1.2 and 1.6 of the ACM Code of Ethics [7], because it would violate the privacy and safety of individuals in the surroundings of the product. To avoid this, we will encrypt the transmitted video data so that it cannot be used maliciously.

**REFERENCES**

[1] Helling., Brett, et al. “DoorDash - Everything You Need To Know About This Delivery Service.” Ridester, 2 Feb. 2020, [www.ridester.com/doordash-on-demand-food-delivery-service/](http://www.ridester.com/doordash-on-demand-food-delivery-service/).

[2] “Bear Robotics, Inc.: Future of Restaurants.” Bear Robotics, [www.bearrobotics.ai/products/](http://www.bearrobotics.ai/products/).

[3] “Savioke.” Savioke, [www.savioke.com/](http://www.savioke.com/).

[4] vestacookingrobot001, and vestacookingrobot001. “China's 1st Cooking Robot: Robot Waiter: Dishwasher Supplier - VestaCookingRobot.” Vesta Cooking Robot Co., Limited, 13 Jan. 2020, [www.vestacookingrobot.com/](http://www.vestacookingrobot.com/).

[5] mpetrova92. “These Butler Bots Are Building Resumes at Companies like FedEx and Residence Inn.” CNBC, CNBC, 25 Aug. 2017, [www.cnbc.com/2017/08/24/fedex-and-hotels-and-hire-autonomous-delivery-robots.html](http://www.cnbc.com/2017/08/24/fedex-and-hotels-and-hire-autonomous-delivery-robots.html).

[6] https://www.mdpi.com/1424-8220/19/24/5480/pdf

[7] “The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society.” Code of Ethics, www.acm.org/code-of-ethics.