Design of Behavior-Based Reactive System for Autonomous Stuff-Collecting Mobile Robot

Eddy Wijanto[1]

[1]Department of Electrical Engineering,
Faculty of Engineering and Computer Science,
Universitas Kristen Krida Wacana, Jakarta
[1]eddy.wiyanto@ukrida.ac.id

Abstract
Recently, autonomous robots play an important role in many aspects. Autonomous robots help improve efficiency and productivity, further reduce errors, and risk rates. Employee safety in high-risk work environments is also becoming one of the benefits of autonomous robots. When designing an autonomous robot, two paradigms can be implemented, i.e., planning and reactive paradigm. A distinct feature of the reactive paradigm is that all activities are carried out through behavior. The benefits of the reactive paradigm include can be applied in limited inexpensive hardware resources, low complexity, goal convergence, easy adaptation to changing situations, and a completely unfamiliar environment with unpredictable mobile obstacles. In this paper, the behavior-based reactive system of an autonomous mobile robot is designed to collect stuff objects. Negotiators have been proposed to combine several behaviors into a system that can be implemented in a variety of situations and cases. The results show that a negotiator can be implemented to realize a fully autonomous mobile robot based on the behavior of the reactive system.

Keywords: behavior-based, reactive, autonomous, negotiator, mobile robot

1. Introduction

Autonomous robots play an important role in various aspects today. Autonomous robots can help to increase efficiency and productivity, further reduce error, re-work, and risk rates. Safety aspect for employees in high-risk work environments also becomes one of the benefits from autonomous robot [1]. In designing autonomous robot, there are two paradigms can be implemented, i.e., planning and reactive paradigm [2]. The reactive paradigm's defining characteristic is that all activities are carried out through behaviors [3]. Behaviors are the results of a direct mapping of sensory inputs to a set of motor actions that are subsequently employed to complete a task. The plan component was removed from the reactive paradigm [4]. The benefits of the reactive paradigm include the ability to use it on robots with limited and low-cost hardware resources, low complexity, goal convergence, ease of adaptation to changing circumstances, and being able to operate a robot safely through wholly unfamiliar settings with unpredictable moving obstacles [2].

In recent years, behavior-based robots have made their presence known in a variety of domains. Behavior-based robotics applications have continued to expand in sectors such as demining, search and rescue, office automation, health care, and are increasingly replacing humans in dangerous and monotonous work [5]. A number of topics of

Received: Revised : Accepted

behavior-based robotics and the reactive paradigm, including its history, principles, applications, and current research was examined in [6].

Three genetically-evolved reactive obstacle-avoidance behaviors for mobile robots were presented in [7], by using genetic algorithms to find the one allowing a mobile robot to best reactively avoid obstacles while moving towards its destination. Three approaches, i.e., a standard method based on potential fields, finite state machines (FSM), and hidden Markov models (HMM)-based probabilistic finite state machines (PFSM) was analyzed [7]. Furthermore, as an intelligent system, a mobile robot must sense its surroundings, perceive its working environment, plan a trajectory, and respond appropriately based on the information. Robotic control architectures specify how these capabilities should be combined in order to build and develop autonomous navigation [8].

Robot autonomy is one of the most important requirements in any field of robotics application. Engineers face numerous obstacles while designing and developing autonomous robots, and full autonomy for robots is still a work in progress [9]. Autonomous robots are complicated systems that necessitate the interaction and cooperation of a large number of disparate software components. Robots are growing more human-like, and as a result, they are becoming important systems that must meet safety requirements, including logical, temporal, and real-time limits [10].

Many intralogistics businesses, such as manufacturing, warehousing, cross-docks, terminals, and hospitals, are now adopting autonomous mobile robots. Autonomous operations in dynamic conditions are possible [11]. Unlike an automated guided vehicle (AGV) system, which relies on a central unit to make scheduling, routing, and dispatching choices for all AGVs, autonomous mobile robots may interact and negotiate with other resources such as machines and systems independently, decentralizing the decision-making process [12].

In this paper, a behavior-based reactive system is designed for autonomous mobile robot in order to collect the stuff object. Negotiators is proposed to realize the system by connecting some behaviors and can be implemented for different situation and case, for example different number and position of stuff object, different number and position of obstacles, and different home position. The implementation of negotiators brings a potential to design a fully autonomous mobile robot.

The remainder of this paper is organized as follows. Section 2 describes the system design, include the robot and components used, along with the behavior implemented in the system. Section 3 presents the system architecture with the negotiator that connect the behaviors. Section 4 shows the strategies and results, for different situations and cases. Finally, Section 5 conclude the work along with the proposed future work.

2. System Design

In order to realize an autonomous stuff-collecting mobile robot system design, which is based on behavior-based reactive robot, a mission was conducted. The mobile robot can collect random objects from a search area and bring it to the home area, which is marked by the hive. The search area is limited by circumference, which is marked by a black line. Inside the search area there is an obstacle that must be avoided. The stuff object and obstacle are placed randomly in the search area. Searching process are limited by working time, which is set for five minutes. After working for five minutes, the mobile robot must return to the home area to rest for two minutes. Furthermore, when the mobile robot battery reaches a certain point i.e., below 7,000 mV, the mobile robot must also return to the home area immediately. Figure 1 depicts the design of the area while the layout of the area is showed in Figure 2. The area consists of nine stuffs to be collected.

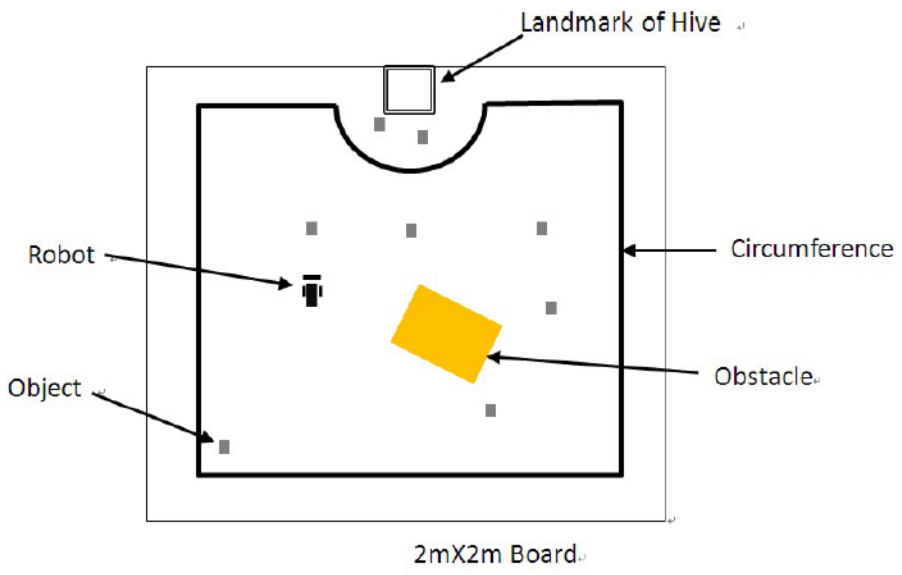

Figure 1. Design of the area

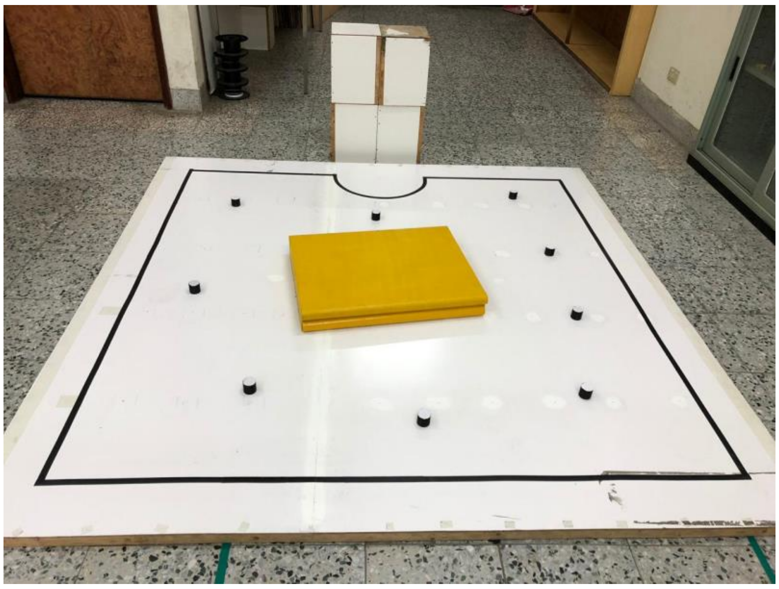
Figure 2. Layout of the area

The robot used in this system is Mindstorms Ev3 where Figure 3 illustrates the design of the robot. The robot is equipped with ultrasonic sensor, light sensor, touch sensor, gripper, and bumper. The component used in this mobile robot are as follows:

- A light sensor for sensing black lines, so that the mobile robot stays in the circumference
- A light sensor near the gripper to detect and find stuff object
- Ultrasonic sensors are used to detect hive, which mark the home area
- Touch sensor is used to detect collisions with obstacle

- A battery level sensor is also used to detect the level of insufficiency of the mobile robot battery
- Timer is used to measure the work time and rest time of the mobile robot.

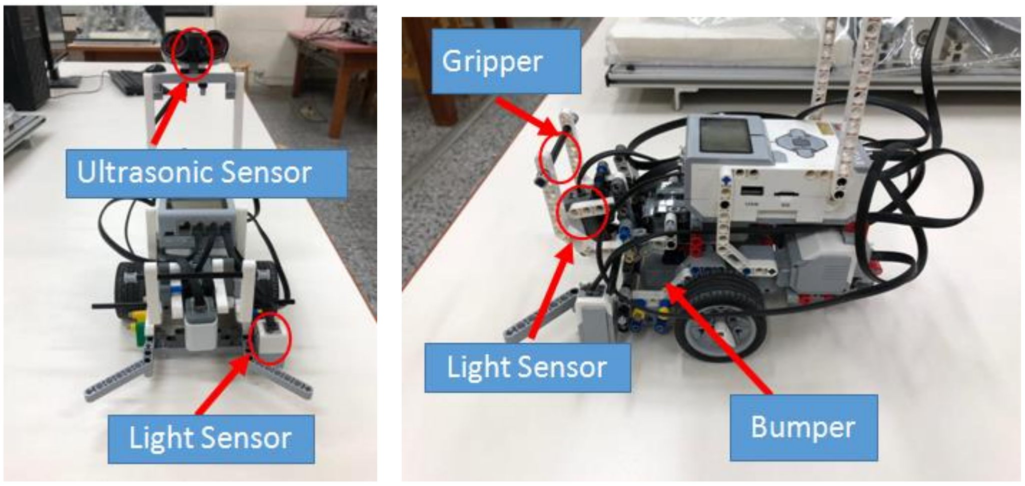

Figure 3. Design of the robot

To realize the behavior-based reactive system, LabVIEW program was implemented in the system. In order to carry out its functions, the mobile robot has several behaviors divided into three levels, listed in Table 1.

Table 1. Mobile robot behaviors

Level	Behavior
Basic	Forward
	Backoff
	Turn Left
	Turn Right
	Grasp
	Release
General	Wandering
	Edge Following
	Rest
	Avoid Edges
	Avoid Obstacle
	Scan for Hive
	Battery Low
Advanced	Search
	Collect Object
	Return to Hive

Forward and Backoff behavior are the mobile robot's behavior to move forward or backward while Turn Left and Turn Right behavior are the behavior of the robot for turning left or right. Grasp is the mobile robot's behavior for retrieving stuff objects after finding it. To detect stuff objects, a light sensor is used where the value of the light sensor reading is greater than 50 indicating that the stuff object is found. Further, the power motor C will be activated to move the gripper. The LabVIEW program for grasp behavior is presented in Figure 4.

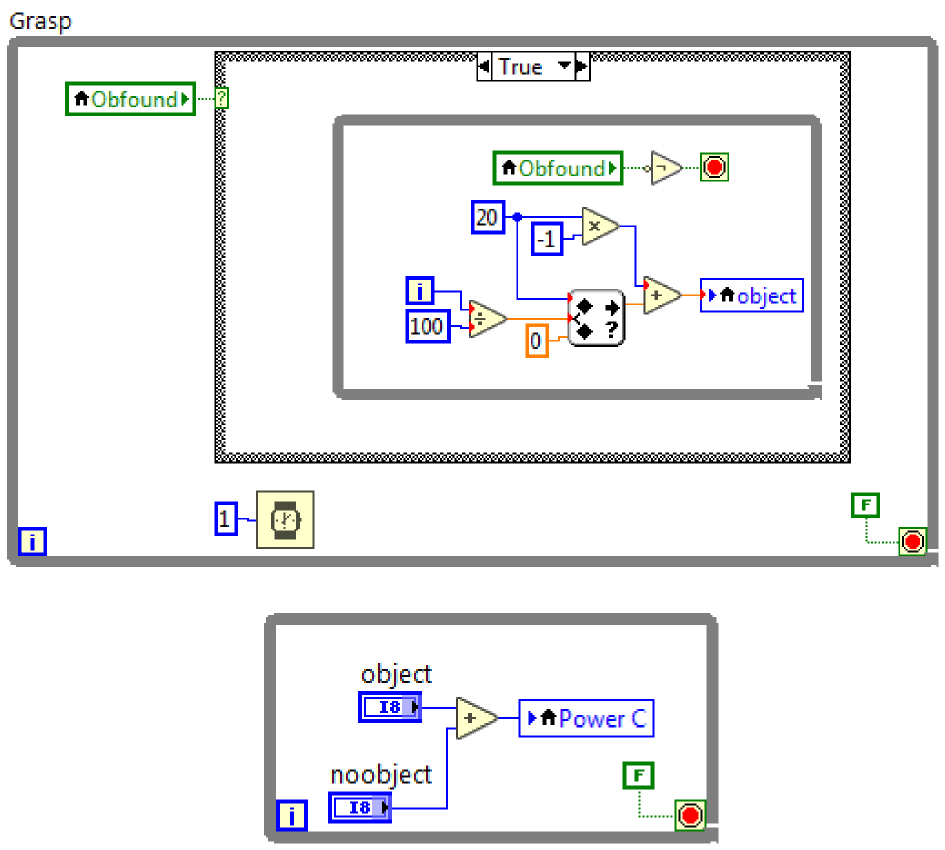

Figure 4. LabVIEW program for grasp behavior

Another basic behavior is Release, which is the behavior of the mobile robot to lift the gripper when the mobile robot has brought the founded stuff object into the home area or when the stuff object is lost from the gripper. When the mobile robot detects the hive as a marker of the home area, then motor C will move the gripper counterclockwise. Other condition, when the value of the light sensor near the gripper is less than 50 which indicates that there are no objects detected, motor C will move gripper counterclockwise, i.e., open. Figure 5 shows the LabVIEW program for Release behavior.

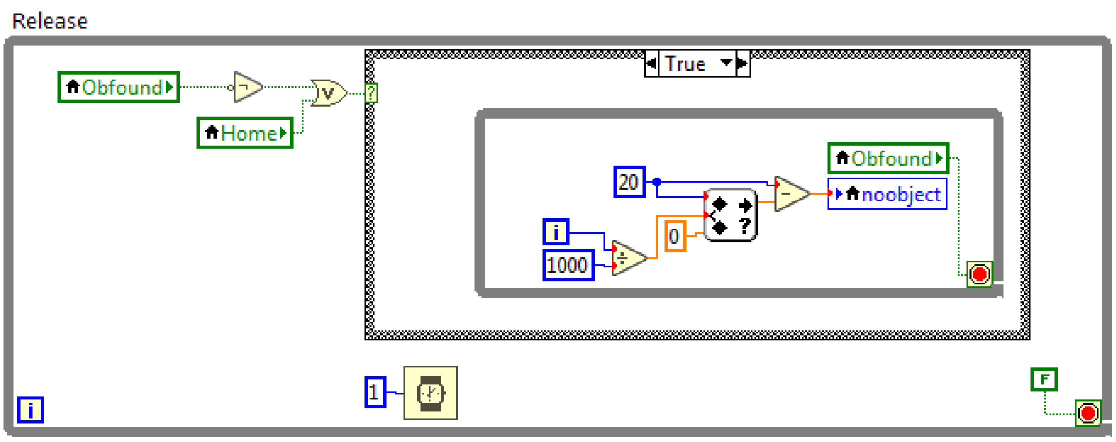

Figure 5. LabVIEW program for release behavior

General behaviors are built with basic behaviors, consisting of Wandering, Edge Following, Rest, Avoid Edges, Avoid Obstacle, Scan for Hive, and Battery Low behaviors. Wandering is the behavior of the mobile robot to look for stuff object. This behavior uses a random number to generate random power from the left and right motor, so that the mobile robot will run a random motion. The wandering behavior is executed when the light sensor does not detect any stuff object and when work time is not over. Figure 6 shows the LabVIEW program for Wandering behavior.

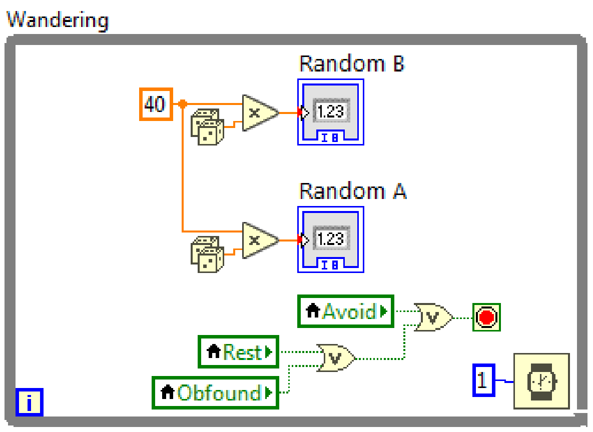

Figure 6. LabVIEW program for wandering behavior

Next general behavior is Edge Following which is the mobile robot's behavior to follow the black line using a light sensor, where the motor speed is 30. The behavior is used to return to the home area, either after finding a stuff object or during a break where the mobile robot will move to follow the black line until it detects the hive as a marker for the home area. The gain used in this behavior is 0.7. LabVIEW program for Edge Following behavior is depicted in Figure 7.

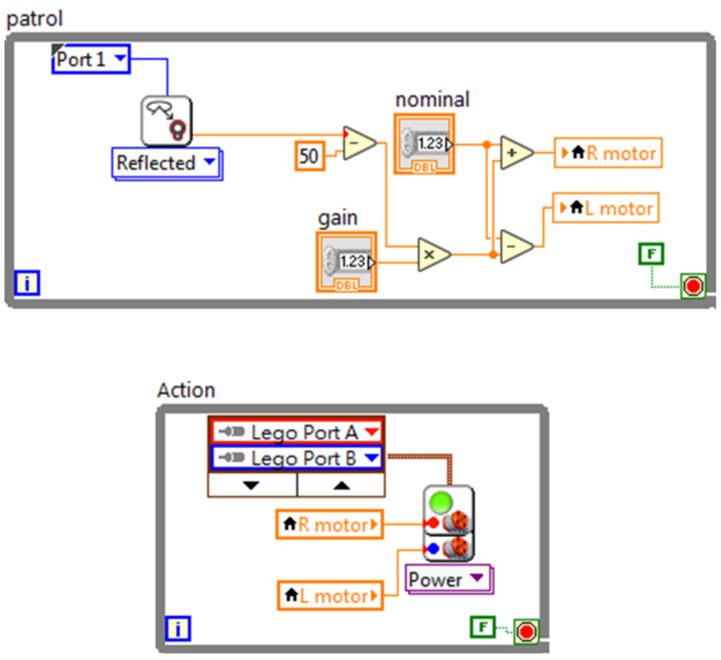

Figure 7. LabVIEW program for edge following behavior

Another general behavior is Rest and Battery Low. In these two behaviors, the mobile robot ends the search process and return to the home area for rest. The specified rest time is two minutes and the specified working time is five minutes. The battery is declared low if the value has reached below 7,000 mV. Figure 8 presents the LabVIEW program for Rest and Battery Low behavior.

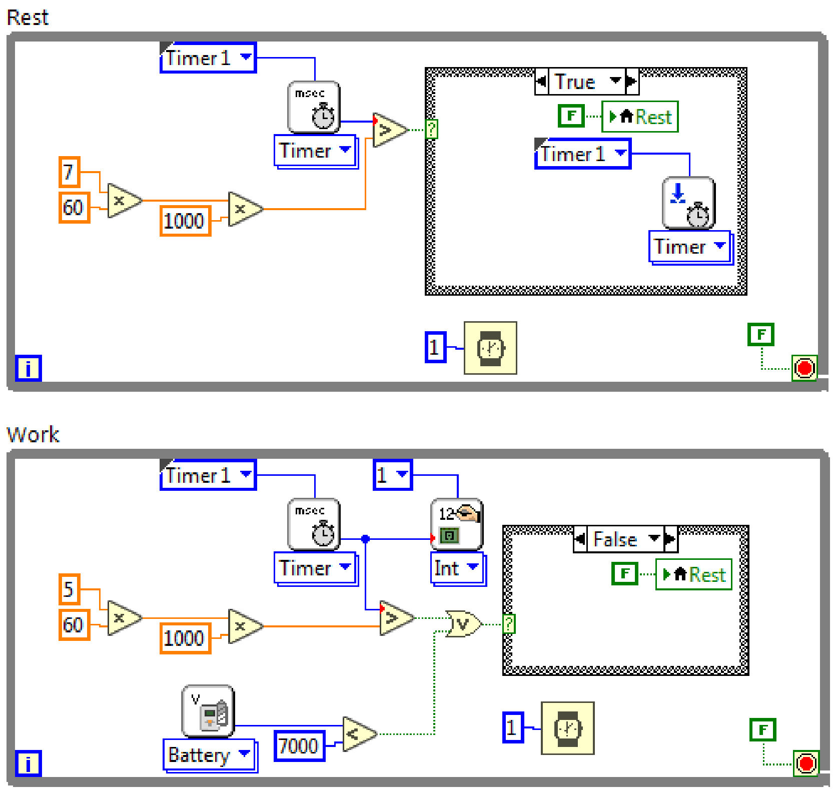

Figure 8. LabVIEW program for rest and battery low behavior

Avoid Edge and Avoid Obstacle are the behavior of the mobile robot in order to ensure that the mobile robot stays inside the search area and ensures that the mobile robot can overcome the collision condition with the obstacle. To detect the edge, a light sensor is used, where when the value of the light sensor reading is less than 30, the mobile robot will avoid behavior. To detect an obstacle, touch sensor is used, where touch sensor value equals 1 if there is an obstacle. The mechanism of avoid behavior is when the mobile robot touches the edge or the obstacle, then the mobile robot will backoff for three seconds followed by random motion to the left or right for one second. Figure 9 illustrates the LabVIEW program for Avoid Edge and Avoid Obstacle behavior.

Scan for Hive is the mobile robot's behavior to find the home area. Mobile robot will return to the home area when stuff object has found and retrieved it; when work time has been completed; or when the battery is low. In order to detect the hives, ultrasonic sensor is implemented. Mobile robot will make a straight motion until the light sensor detects the black line, then the mobile robot will do the Edge Following behavior until the ultrasonic sensor reading value is smaller than 50 cm, which marks the hive. LabVIEW program for Scan for Hive behavior is depicted in Figure 10.

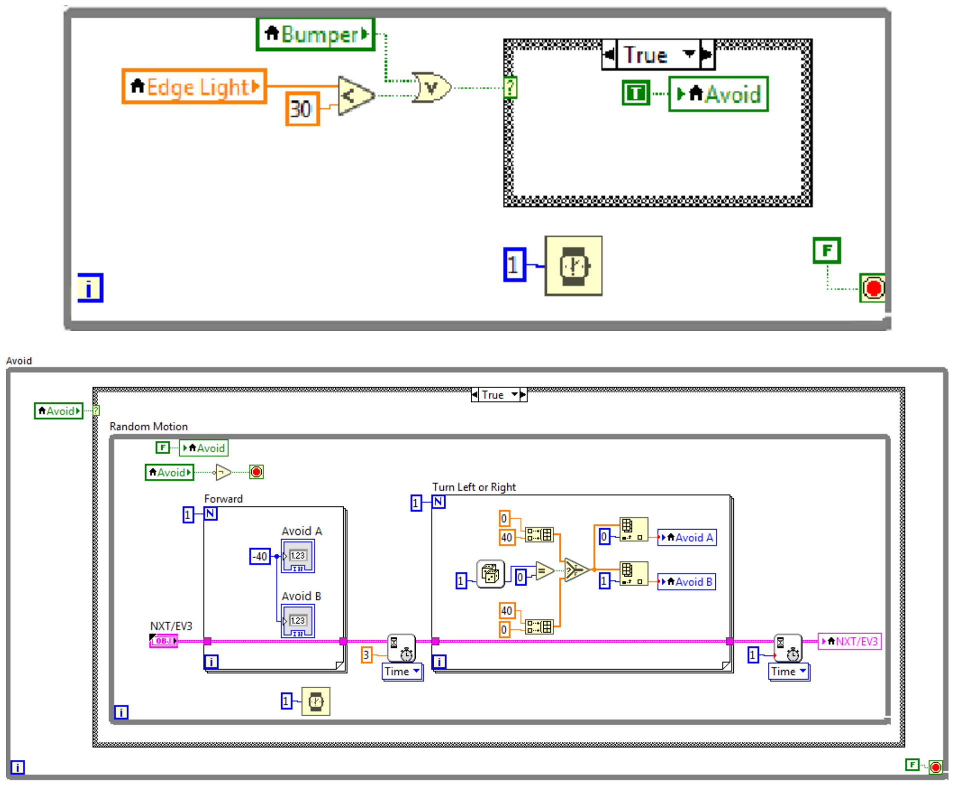

Figure 9. LabVIEW program for avoid edge and avoid obstacle behavior

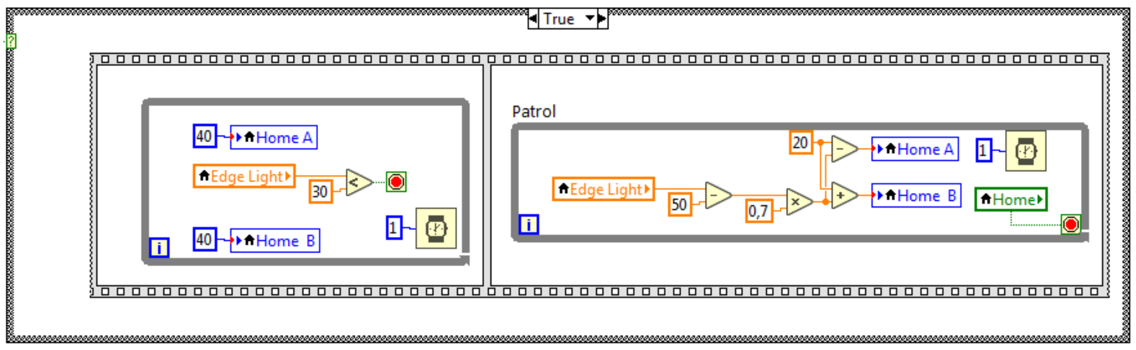

Figure 10. LabVIEW program for scan for hive behavior

Search, Collect Object, and Return to Hive are advanced level behaviors that applied by using basic and general behaviors. Search behavior is implemented with Wandering, Avoid Edges, and Avoid Obstacle behaviors while Collect Object behavior is applied with Wandering, Edge Following, and Avoid Obstacle behaviors. Further, Return to Hive behavior is implemented with Edge Following, Rest, Avoid Obstacle, Scan for Hive, and Battery Low behaviors.

3. System Architecture

After designing the behaviors needed, in order to realize the system, eight negotiators are used to connect the behavior of the mobile robot system. The negotiators are proposed in this paper in order to realize a behaviors-based reactive system where the situation and environment are random and unplanned. The negotiators used in the mission are design based on the condition as follows:

- Mobile robots run the Wandering behavior to look for stuff objects
- When a stuff object is detected, the mobile robot runs the Grasp behavior where the Grasp behavior suppresses the Wandering behavior so that the mobile robot will stop searching for stuff objects and run the Edge Following behavior and Scan for Hive behavior.
- When the hive is detected, the mobile robot will run the Release behavior, further it will run the Wandering behavior to look for the next stuff object.
- If in the search for stuff objects, the mobile robot hit the black line or hit the obstacle, then the mobile robot will run a behavior for Avoid Edge or Avoid Obstacle, suppressing the Wandering behavior.
- When the working time has reached five minutes, the mobile robot will run the Edge Following behavior and Scan for Hive behavior suppressing the Wandering behavior to return to the home area and rest for two minutes.
- When the battery is low, the mobile robot will run the Edge Following behavior and Scan for Hive behavior suppressing the Wandering behavior to return to the home area.

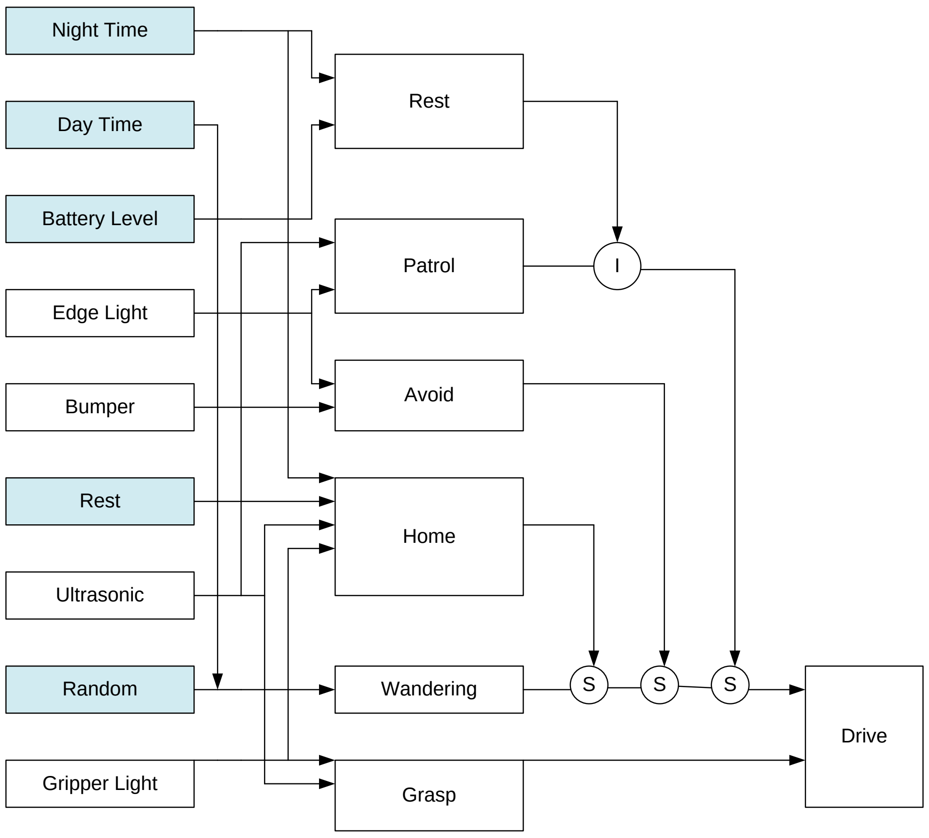

Figure 11. System architecture

Figure 11 illustrated the system architecture when the condition will initiate or suppress the behaviors. In Figure 11, *I* refer to initiate and *S* refer to suppress. Night time represents the working time is over while day time represents the working time of mobile robot. Patrol represents the searching process when there are no edge or obstacle detected.

First negotiator is activated when no object has been found, no edge detected, no obstacle detected, work time is not over, and low battery has not been detected, then the mobile robot will do a Wandering behavior. Figure 12 presents the LabVIEW program for negotiator 1.

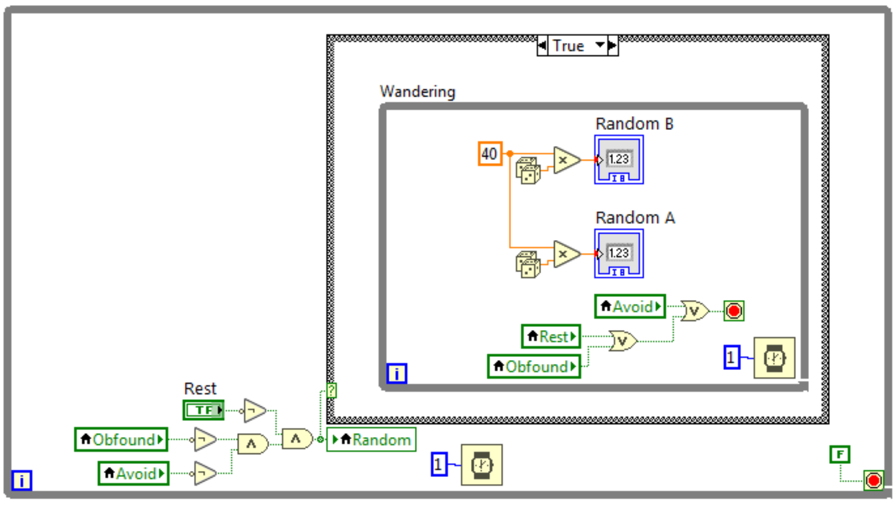

Figure 12. LabVIEW program for negotiator 1

Negotiator 2 will work when object is found, edge detected, obstacle detected, end of work time, or low battery detected, then mobile robot will stop wandering behavior. Figure 13 depicts the LabVIEW program for negotiator 2.

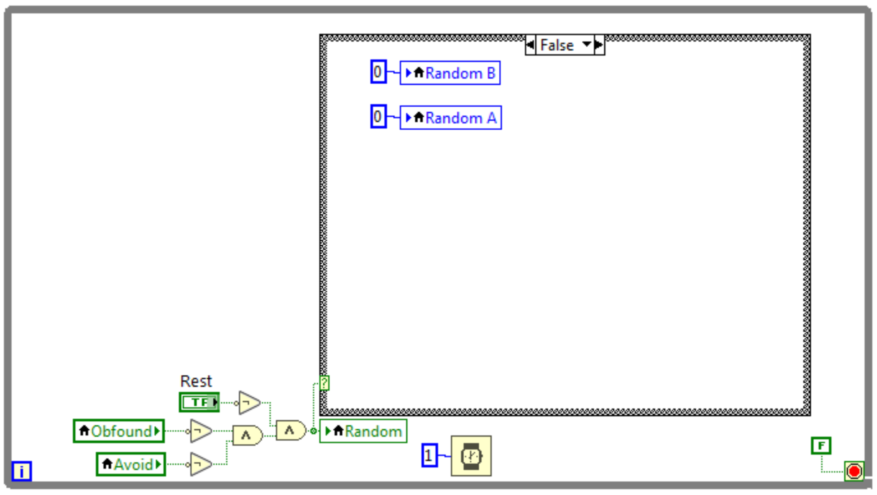

Figure 13. LabVIEW program for negotiator 2

Negotiator 3 is used when the object is found, work time is over, or a low battery is detected, then the mobile robot will run the Edge Following behavior and Scan for Hive to return to the home area. Figure 14 shows the LabVIEW program for negotiator 3.

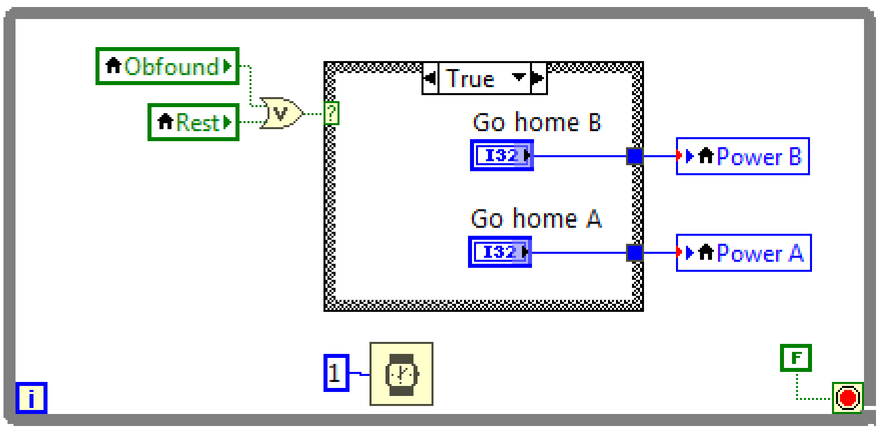

Figure 14. Negotiator 3

Negotiator 4 will be activated when the object has not found, work time is not over, or the low battery is not detected, the mobile robot will run the Wandering behavior. Figure 15 illustrates the LabVIEW program for negotiator 4.

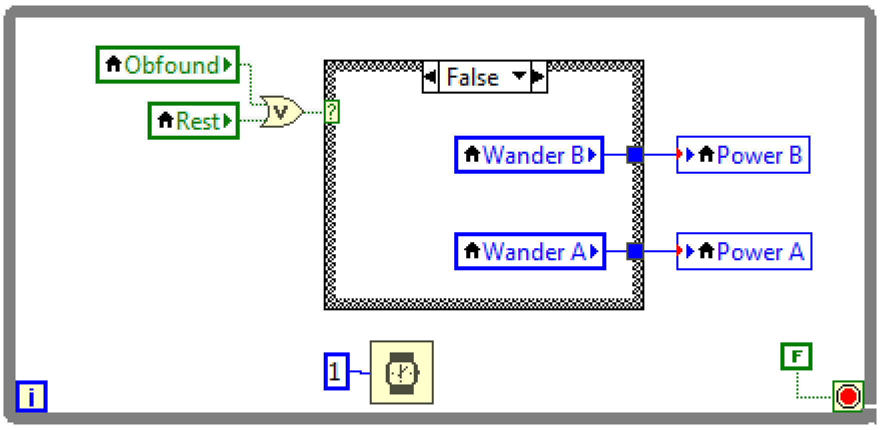

Figure 15. Negotiator 4

Negotiator 5 is implemented when the object is found, work time is not over, or a low battery is detected, then the mobile robot will run the Edge Following behavior and Scan for Hive behavior to return to the home area. Figure 16 demonstrates the LabVIEW program for negotiator 5.

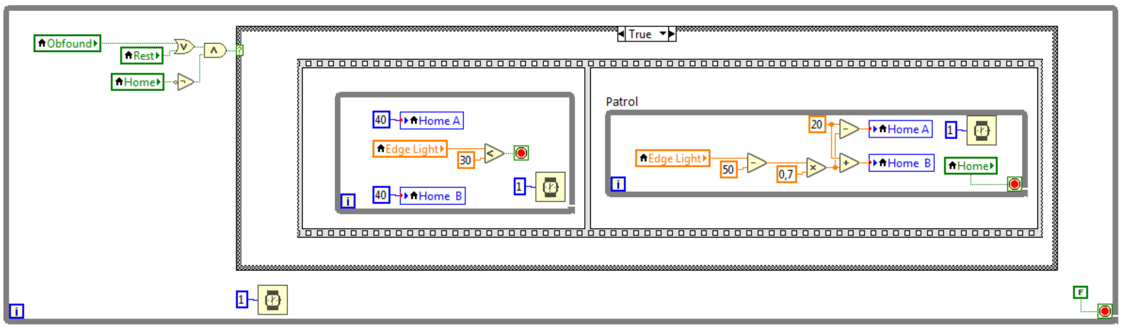

Figure 16. Negotiator 5

Negotiator 6 will run when the object has not found, work time is not over, or the low battery is not detected, the mobile robot will not run the Return to Hive behavior. Figure 17 presents the LabVIEW program for negotiator 6.

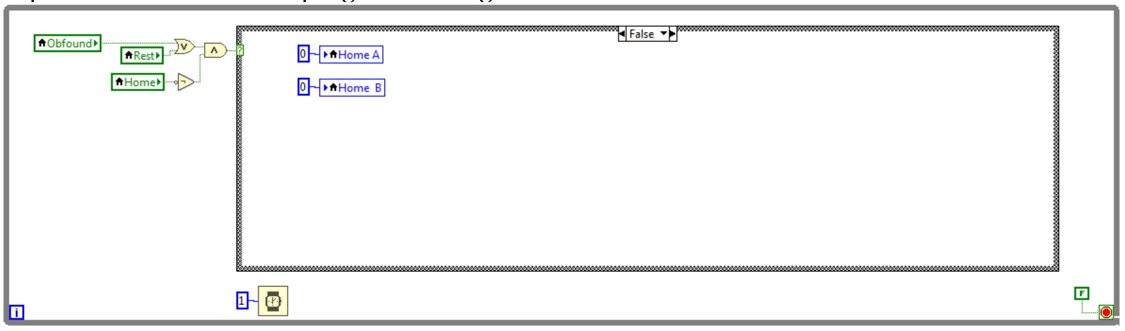

Figure 17. Negotiator 6

Negotiator 7 is conducted when the ultrasonic sensor detects a hive and the working time is not over, then the mobile robot will run a Backoff and Turn behavior to go out from the home area for finding the next stuff object or when touch sensor detects a collision with the obstacle the mobile robot will run the Backoff and Turn behavior. Figure 18 depicts the LabVIEW program for negotiator 7.

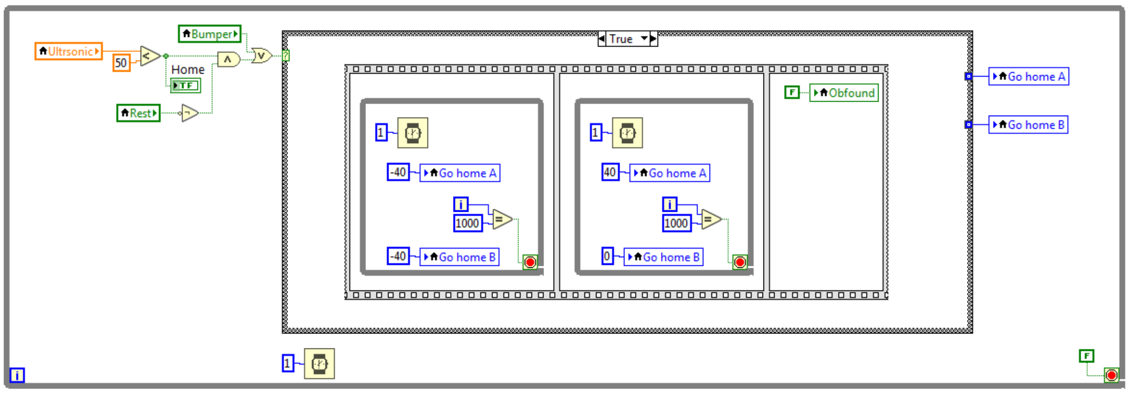

Figure 18. Negotiator 7

Negotiator 8 will function when the ultrasonic sensor does not detect a hive, the working time is over, and touch sensor does not detect a collision with the obstacle then the mobile robot will run the Rest behavior. Figure 19 illustrates the LabVIEW program for negotiator 8.

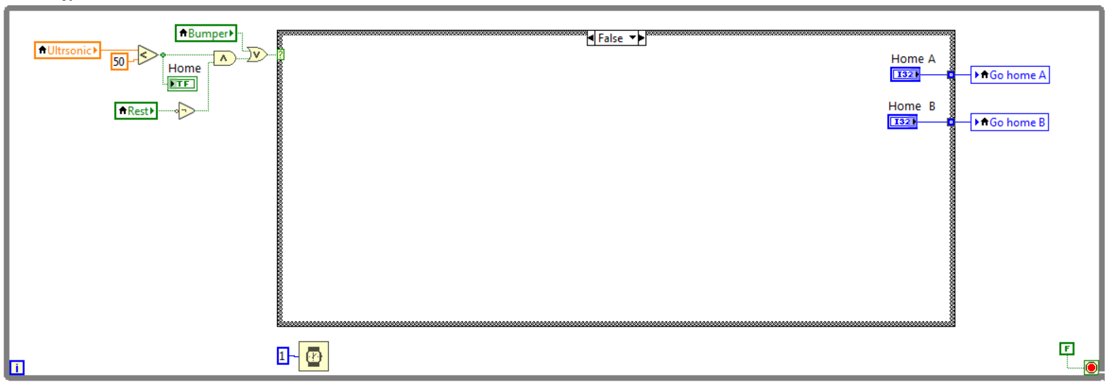

Figure 19. Negotiator 8

4. Strategies and Results Analysis for Different Situations and Cases

The strategies used in several situations and cases to ensure the success of the mission are as follows:

- Robot Design Strategy

  Robot design, including sensor placement, is one of the determining factors for mission success. For the detection of stuff objects, light sensors are installed in such a way so that the position is not too high from the point that the stuff object is detected. It is necessary to consider the position of the sensor installation to the size or the height of the objects. This strategy used to ensure the value of the light sensor reading when there is no stuff object and when there is a stuff object has a considerable difference. In addition, the gripper is also designed in such a way to ensure that the stuff object that has been taken will not lost from the mobile robot's gripper. The position of the ultrasonic sensor is also designed in such a way with a position slightly upward, so that it has a wide enough detection range to make it easier to detect hives.
- Wandering Strategy

  Searching for stuff objects by robots is done with random movements. Initially the random movement strategy used includes random forward movement, random left movement, and random right movement by using a random number in LabVIEW. If the random movement generated is too random where the robot moves so randomly, it will very difficult for the robot to find the stuff object. The wandering strategy can be changed, still uses random motion, but only the power from the left motor and the right motor is randomized. This strategy produces a more stable robot movement so that it is easier to find the stuff objects. Mobile robots find all nine stuff objects in four work cycles.
- Strategy Towards the Home Area

  The chosen strategy to return to the home area, both after the mobile robot finds the stuff object and when the work time has ended is in a straight motion until it finds a black line and then execute the Edge Following behavior until the hive is detected. This strategy saves more time than doing random movements to detect hives.
- Strategy for Getting Out from Home Area

  After the mobile robot finds the stuff object and takes it to the home area, if the work time is not over, the robot will return to search for the next stuff object. Initially, the strategy used by mobile robot after putting food in the home area was to do an Edge Following behavior for a certain amount of time before returning to search for the next stuff object. However, this strategy requires a relatively long time. For that, the strategy can be changed, after the mobile robot put the stuff object in the home area, the mobile robot will make a backoff movement then turn left or right and continue with wandering random motion. This strategy saves the working time.

There are several conditions that cause the mobile robot fail to carry out its mission, namely:

- In the condition where the stuff objects are quite a lot in the home area, when the mobile robot detects the hive and release the stuff object in the gripper, the touch sensor is pressed by the existing stuff object so that the mobile robot runs the Avoid Obstacle behavior. Because Avoid Obstacle behavior suppressing other behaviors so that the mobile robot will be stuck and unable to leave the home area. When trapped in the home area, it can happen that the mobile robot detects and retrieves stuff objects that are already in the home area.

- The mechanism after the collision with the obstacle is a backward movement for three seconds followed by left or right movement. The mechanism for avoid edges is also done with a backward motion followed by left or right movement. At the position where the obstacle distance with the black line is close enough, it can happen that the mobile robot does a backward mechanism when the touch sensor hits the obstacle and when the mobile robot backoff, the mobile robot hit the edge so the it will backoff again, causing the mobile robot out from the circumference. For that, the time given for the backward movement needs to be arranged in such a way that it is not too short and not too long. Figure 20 shows some failed condition.

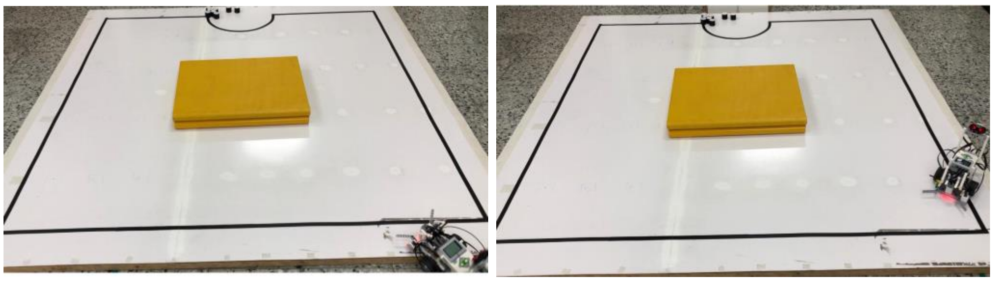
Figure 20. Failed condition

Table 2 shows the times needed for collecting each stuff object with the average time of 225 seconds.

Table 2. Stuff object collection time

Stuff Object	Time
1	198 seconds
2	182 seconds
3	244 seconds
4	208 seconds
5	245 seconds
6	291 seconds
7	247 seconds
8	157 seconds
9	250 seconds
Average	225 seconds

5. Conclusion

Behavior-based reactive robot systems do not require a detailed plan like in the hierarchical system, thereby reducing the complexity of programming. Behavior-based reactive robot systems have a fast reaction toward the changing environments. Negotiators proposed in this work has potential to realize fully autonomous mobile robot. Further, robot design and sensor placement are one of the determining factors for success in the mission of this project. Behavior and negotiator need to be designed appropriately for mission success so that the robot can give the right reaction to the environment. Hybrid system can be implemented to increase the performance of mobile robot in future work.

References

[1] J. R. S. Ibáñez, C. Perez-del-Pulgar, A. Garcia, "Path planning for autonomous mobile robots: A review," *Sensors*, vol. 21, h. 7898, 2021. Doi: 10.3390/s21237898.

[2] M. Robin. (2000). Introduction to AI Robotics. MIT Press.

[3] X. Wang, J. Zhang, "RPL: A robot programming language based on reactive agent," 2017. Doi: 10.2991/eame-17.2017.59.

[4] T. A. Tobaruela, A. Rodríguez, "Reactive navigation in extremely dense and highly intricate environments,". *PLOS ONE*, vol. 12, e0189008, 2017. Doi: 10.1371/journal.pone.0189008.

[5] I. Hassani, I. Maalej, C. Rekik, "Robot path planning with avoiding obstacles in known environment using free segments and turning points algorithm," *Mathematical Problems in Engineering*, h. 1-13, 2018. Doi: 10.1155/2018/2163278.

[6] N. Lazzeri, D. Mazzei, L. Cominelli, A. Cisternino, D.D. Rossi, "Designing the mind of a social robot," *Applied Sciences*, vol. 8, h. 302, 2018. Doi: 10.3390/app8020302.

[7] J. Savage, S. Muñoz, L. Contreras, M. Matamoros, M. Negrete, C. Rivera, G. Steinbabuer, O. Fuentes, H. Okada, "Generating reactive robots' behaviors using genetic algorithms," h. 698-707, 2021. Doi: 10.5220/0010229306980707.

[8] M. Alatise, G. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, h. 1-1, 2020. Doi: 10.1109/ACCESS.2020.2975643.

[9] T. Asokan, "Autonomy for robots: design and developmental challenges (Keynote Address)," *Procedia Technology*, vol. 23, h. 4-6, 2016. Doi: 10.1016/j.protcy.2016.03.066.

[10] P. Panigrahi, S. Bisoy, "Localization strategies for autonomous mobile robots: a review," *Journal of King Saud University-Computer and Information Sciences*, 2021. Doi: 10.1016/j.jksuci.2021.02.015.

[11] J.-H. Cho, Y.-T. Kim, "Design of autonomous logistics transportation robot system with fork-type lifter," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, h. 177-186, 2017. Doi: 10.5391/IJFIS.2017.17.3.177.

[12] F. Giuseppe, D. Koster, R. Sgarbossa, F. Strandhagen, and J. Ola, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journal of Operational Research*, vol. 294, 2021. Doi: 10.1016/j.ejor.2021.01.019.