Submitted to Journal of Research in Engineering and Technology, Kasetsart University, Thailand

# Design of a Behavior-Based Air-Duct Cleaning Robot

**Withit Chatlatanagulchai**[1]
**Kanjana Chawbankor**[2]
**Kunnayut Eiamsa-ard**[3]
**Phichai Kritmaitree**[4]
Department of Mechanical Engineering, Faculty of Engineering,
Kasetsart University, Bangkok 10900, Thailand
([1]fengwtc@ku.ac.th, [2]kanjana_chawbankor@hotmail.com,
[3]fengkye@ku.ac.th, [4]fengpck@ku.ac.th)

**Abstract**

Indoor air quality affects human's quality of life. In building that has central air-conditioning system, contaminants accumulate especially at supply and return air ducts. Traditional cleaning of the air ducts is inconvenient and expensive, making the cleaning frequency less often than it should. This paper presents the design of a novel air-duct cleaning robot. The robot is aimed to navigate by itself in the air-duct system and to come back to base when necessary. It is designed to have brushes and vacuum. Using behavior-based robot programming, operations are broken down into tasks, and tasks are broken down into behaviors. Fixed arbitration is used to decide which behavior is to be performed. This mobile robot should enable the cleaning to be more often and more convenient. It is aimed that the users should be able to operate this robot by themselves. A prototype is built to test some navigation tasks.

**Keywords:** Mobile Robotics, HVAC Cleaning, Air Duct Cleaning, Behavior-Based Robotics

## 1 Introduction

Scientists have found indoor air environment to be even more hazardous to our health than outdoor air. Indoor air quality can cause problems from cough, eye irritation, and headache to more serious illness such as asthma and allergy. World Health Organization reported that as many as 30 percent of buildings worldwide generated excessive complaints related to indoor air quality [1]. In the US, random sampling of office workers showed

that 24 percent perceived air quality problems in their work environments, and 20 percent believed their work performance was hampered [2].

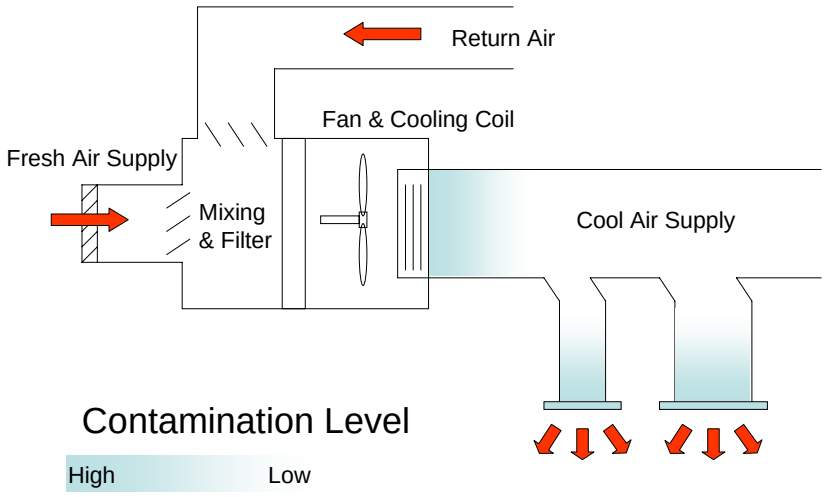



Fig. 1. Overall air-conditioning system.

Indoor air contaminants can originate within the building or be drawn in from outdoors. Sources of contaminants can be from pollinating trees, industrial emissions, vehicle exhaust, cleaning chemicals, and pesticides. The indoor air pollution problem is worsen especially in buildings that use central air-conditioning system. Fig. 1 shows a typical air conditioning system, whose fresh air supply is drawn from outside, and a portion of air is re-circulated. Contamination level is high especially at inlet and outlet of cool air supply. Typical contamination results from dust and debris accumulation, excessive humidity, condensation, and water infiltration.

Fig. 2. Contamination in supply and return ductworks.

Fig. 2 shows some examples of contamination found in cool air supply and return air ductworks, ranging from floor wax, carpet deodorizers, air fresheners, cigarette smoke, and microbes such as mold and fungi. Contaminants can also be those fallen out from components of the air-conditioning system itself such as from return ductwork plenum, mixing plenum and filtration unit, blower and coil section, sound attenuator and supply ductwork, diffusers and mixing box.

Typical air-duct cleaning procedures include using zone bags to block airways, having air compressor to create high-pressure air inside the duct, using robot to scrub and brush duct walls, and having filtration unit at the other end to filter out dust and contaminants. Fig. 3 shows the typical air-duct cleaning procedures. More often, whip head, python brush, and human are used instead of the robot to scrub and clean walls. The filtration unit can sometimes be replaced by a speck bag.

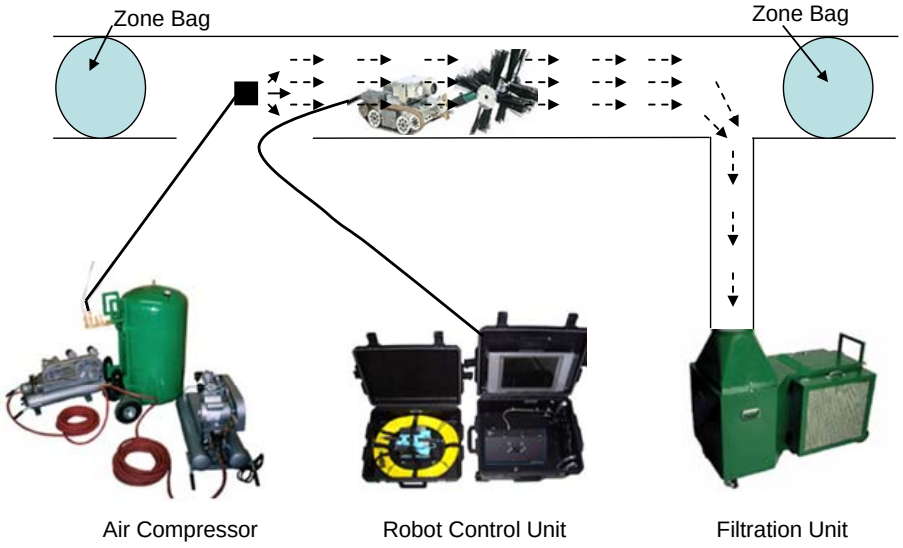



Fig. 3. Typical air-duct cleaning procedure [3].

The problem with the typical cleaning procedure lies in the inconvenience of having to block the airways. The air duct can be cleaned one section at a time, which requires strict planning and usually not all sections can be cleaned due to lack of available openings. The air compressor, robot control unit, and filtration unit are difficult to maneuver. More often, the cleaning crews must go in at night or weekend, when there is nobody in the building. The cleaning cost is usually high, and most buildings cannot afford

to clean the air duct as often as it should. It is irony when we think of how often we clean our floor comparing with how often we can clean our air duct.

In this paper, we design an intelligent mobile robot that can navigate, scrub, and vacuum the air-duct system. Two treads support the cylinder-shape robot. Since inside of the air duct is not flat, sometimes there are protruded nuts or uneven floor, we need treads to provide good traction. Cylinder shape helps escaping from obstacles. There are four brushes: a top brush connected with a telescope to scrub ceiling and walls, a small brush underneath to clean corners, and two roller brushes to pick up big particles. A motor creates suction for small particles on the floor as well as air-suspended particles. Fig. 4 shows a sketch of the air-duct cleaning robot.

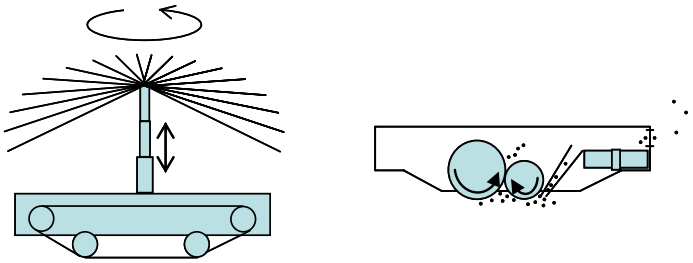


Fig. 4. Sketch of the air-duct cleaning robot.

The robot can achieve its required tasks by breaking each task into behaviors that the robot can perform one at a time. This robot programming technique is called behavior-based programming, made famous by Rodney Brooks and colleagues at MIT [4]. In the past fifteen years, there are surprisingly not many good textbooks on the behavior-based robotics. Some of the notables are [5] - [7].

The paper is organized as follows. Section 2 illustrates a day in a life of the cleaning robot. It contains many illustrations of how our robot behaves when left in a test track. This section should give the readers clear ideas of how the robot actually operates. Section 3 follows the materials in Section 2 by listing necessary tasks of the robot to perform the whole operation. Section 4 is where we list robot behaviors. The robot can only perform one behavior at a time, and combinations of different behaviors enable the robot to perform its tasks. Section 5 is when we get down to robot physical. This section contains various sensors and actuators used with the robot, as well as some design concepts specific to robot physical. Section 6 presents details of robot arbitration. Since a robot can perform only one behavior at a time. Arbitration prioritizes behaviors and chooses to perform a behavior according to its

priority level. After all the design concepts are devised, we actually built a prototype robot mainly to test some navigation algorithms. We discuss the prototype in Section 7. Conclusions are given in Section 8.

## 2 A Day in a Life of a Cleaning Robot

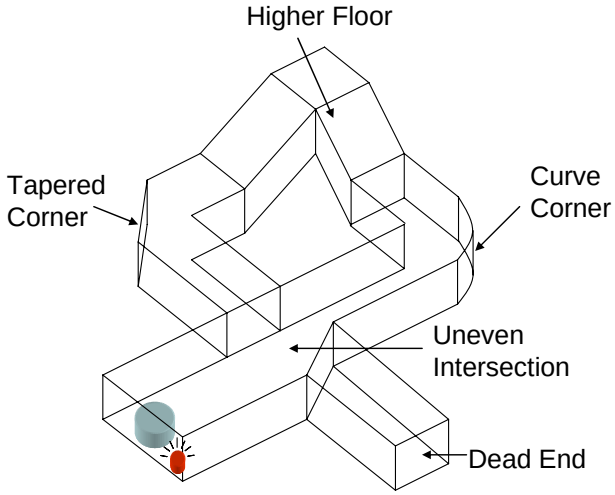



Fig. 5. Model of air duct used as robot test track.

Consider a test track in Fig. 5. We designed this test track to contain most of the challenges the robot will face in real world. The test track has an intersection that the robot must choose its heading. One route is intended to be narrower than the others to test the ability to navigate through uneven routes. The test track also has both round and tapered corners to test the ability to turn. A portion of the test track is lifted higher, so the robot must be able to climb steep slopes. The dead end is used to test the ability to reverse.

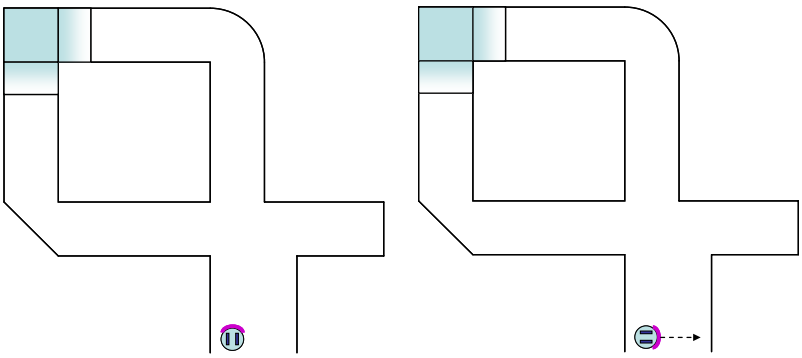


Fig. 6. How the robot operates at entrance.

(Fig. 6: Left) Suppose an operator starts the robot at the entrance of the test track. There are two modes to choose from: tidy or random. The operator chooses the tidy mode. (Fig. 6: Right) Vacuum is turned on at the same time as the top brush starts to rotate and the telescope extends until the brush touches and scrubs the ceiling, as indicated by a telescopic switch. When the robot touches the wall, the bumper switch activates the telescope to move the top brush down to scrub the wall.

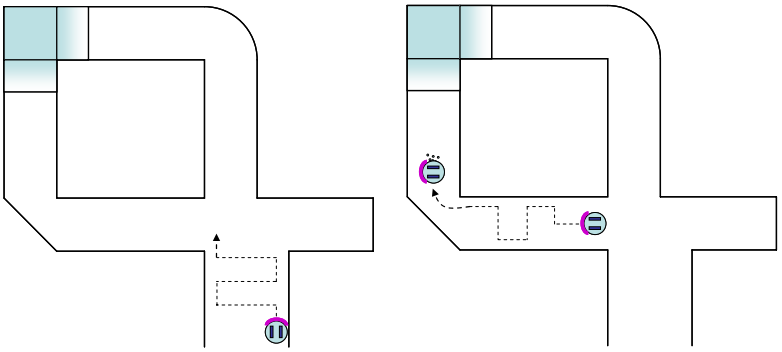


Fig. 7. Robot's behaviors at intersections and at dirtier spots.

(Fig. 7: Left) Being in the tidy mode, the robot zigzags to cover the whole area until it reaches the intersection. The information from IR range sensors around the robot tells the robot that it has just arrived at an intersection. If there are more than one possible routes, the robot will choose in the clockwise direction, that is, it will go left first then straight then right. (Fig. 7: Right) The robot chooses to go left until it reaches the tapered corner, then it decides to go right because there are no left and straight routes. At that time, dust sensor detects high level of dust. The robot switches from zigzag to circle the area to concentrate on the dirtier spot. After the area is clean, the robot assumes the zigzag movement.

(Fig. 8: Left) The robot arrives at the slope area. Signal from tilt sensor tells the robot whether the slope is too steep for the robot to climb. If too steep, the robot will reverse. (Fig. 8: Right) At that instance, the battery level reaches the low level as indicated by battery sensor. The robot will enter the homing mode, stop all activities, and head back to base by following signal from IR beacon. When the robot is home, it will recharge the battery and will go back to the same spot. Homing can take place in three incidents. First, when operator calls, second, when dust is full, third, when all areas are clean, and finally, when the battery reaches low level.

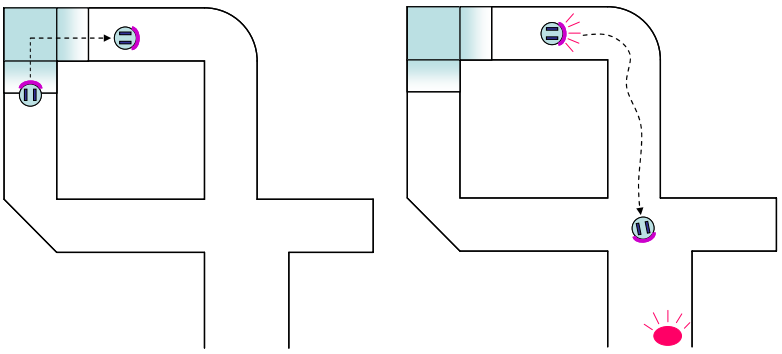


Fig. 8. When the robot is at slopes and homing behavior.

(Fig. 9: Left) After temporarily homing, the robot must go back to where it left off. There are two possible options. First, the robot follows the same route without zigzagging and will assume the tidy mode (zigzagging and vacuuming) when the dust sensor senses high level of dust. Second, we may have another robot as pilot. The pilot robot consumes less battery than the vacuum robot and acts as IR beacon for the vacuum robot to go back to the same spot to assume the cleaning work. (Fig. 9: Right) When the robot arrives at a dead end, it will reverse and cruise without repeating the cleaning. This behavior can be done by using information from dust sensor or compass. To prevent the robot from falling down an opening, IR proximity sensor located beneath the robot pointing down will signal a stop when the sensor cannot sense the floor.

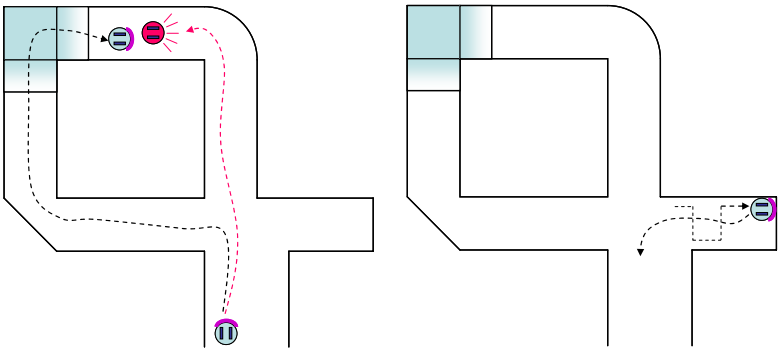


Fig. 9. How the robot goes back to where it left off and its behavior at dead end.

(Fig. 10: Left) For the random mode, the robot goes straight until it reaches an intersection, then it will reverse and start cleaning by moving randomly until that section is clean. How long the robot cleans each section

depends on the length of that section (the robot can measure this while running straight) or on the dust level sensed by the dust sensor. While running randomly, if the robot comes to another intersection, it will reverse and back away from the intersection until that section is clean. (Fig. 10: Right) When the cleaning time for that section runs out, the robot uses compass to go back to the first intersection it came across, then it will move to different section according to the clockwise direction.

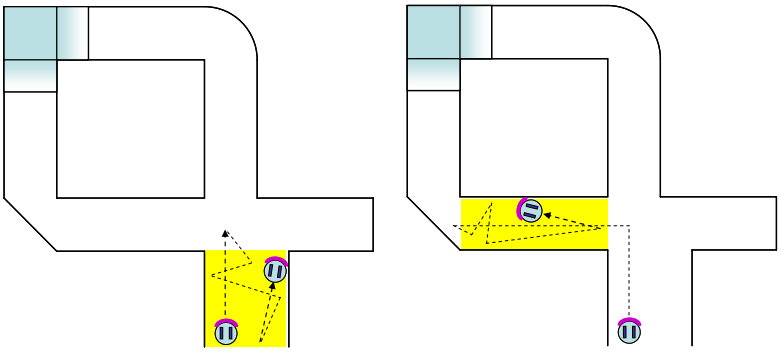


Fig. 10. When the robot is in random mode.

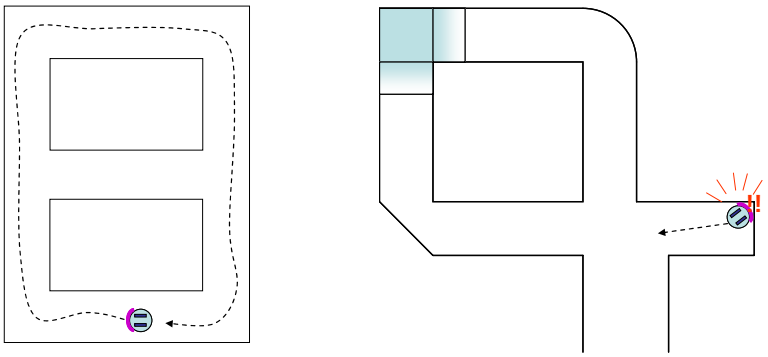


Fig. 11. A special case and when the robot gets stuck.

(Fig. 11: Left) Commanding the robot to go in clockwise direction has a flaw. For example, suppose the air duct has number-eight shape, if the robot always chooses to go in clockwise direction, it will miss cleaning the middle section. Therefore, we need to monitor dust level using the dust sensor. If the dust level is low for a long period, switch to counter-clockwise direction instead. (Fig. 11: Right) When the robot gets stuck, for example, the robot might tangle with an obstacle in the air duct, the wheels' shaft encoders indicate the stoppage of the robot. In case of wheel slip, the encoder might indicate that the wheel spins but the robot does not move, we can then rely on

the IR range sensors around the robot. If their readings do not change for a long period, then the robot is stuck. When the robot is stuck, it will go into escape mode. The wheels will spin faster with random direction back and forth until the robot is untangled. After trying for a while, if it is still not loose, the robot will use its buzzer to call for help. The sound leads the operator to come to where the robot is.

## 3 Robot Tasks

From Section 2, the readers should have some ideas of what are needed for the robot to operate effectively. This section lists robot tasks. A task is what a robot must perform in order to achieve its goal. In designing a behavior-based robot, one should list all the necessary tasks before designing its behaviors. All tasks are divided into navigation tasks, tasks the robot must perform to navigate effectively in the air-duct system, and operation tasks, tasks concerning scrubbing and vacuuming operations.

The navigation tasks are as follows:

- To cover all areas
- Not repeat the same area
- To circle around dirtier spots
- To avoid falling off openings and holes
- To avoid moving into unmovable zones such as different-level floors or too-steep slopes
- To avoid hitting obstacles
- To come back to base when an operator calls
- To come back to base when all areas are clean
- To come back to base when the battery almost runs out
- To come back to base when dust is full
- To go back to where it left off to continue its work

The operation tasks are as follows:

- To vacuum and store big and small particles and air-suspended particles
- To scrub floor, walls, and ceiling
- To clean at corners of walls
- To come back to charger and self charge
- To drop the dust at base automatically
- To be able to attach UV light to kill germs
- To be able to attach nozzle for spraying disinfectant
- To be able to connect to a video camera
- To communicate with operator via wireless communications

## 4 Robot Behaviors

Most robots can only execute one behavior at a time. A behavior consists of a control component and a trigger component. Control component transforms sensory information into actuator commands. Trigger component determines when it is appropriate for the control component to act (or a behavior to be executed.) A task can be performed by executing groups of related behaviors. We should design a behavior to be as simple as possible because overloading a behavior can lead to programming error.

Behaviors of the robot are as follows:

- Zigzag: cleaning by zigzagging or tidy mode
- Random: cleaning by random movement or random mode
- Intersection: behavior when the robot arrives at an intersection
- Dirty: circling behavior when the robot is in dirtier area
- Tilt: a behavior when robot moves over slopes
- Cruise Low Battery: running without scrubbing or vacuuming when battery level is low
- Cruise On Call: running without scrubbing or vacuuming when being called by an operator
- Cruise Dust Full: running without scrubbing or vacuuming when dust is full
- Cruise Finish: running without scrubbing or vacuuming when all areas are clean
- Cruise Left Out: running without scrubbing or vacuuming to where the robot left out to continue its work
- Cruise Dead End: running without scrubbing or vacuuming when finds dead end
- Cruise Random: running without scrubbing or vacuuming before assuming Random behavior
- Charge: behavior during battery charging
- Escape: when the robot is stuck and tries to untangle

More details of how each behavior is triggered and controlled will be given in Section 6 where we discuss arbitration.

## 5 Physical Interfaces

This section discusses sensors and actuators. Robots perceive their surroundings through sensors and perform actions through actuators. We divide sensors into four groups according to their functions as follows:

1) Collision and stall detection

- Bumper switches

- Shaft encoders
- Tilt sensor
- Wheel switches

2) Obstacle avoidance

- IR range sensors
- IR proximity sensors
- Sonar range sensors
- Dust sensor

3) Homing

- IR coded beacon
- Compass
- Park sensor

4) Scrub and vacuum operations

- Telescopic switch
- Battery level sensor
- Dust weight sensor
- Mode switch

Actuators are right-wheel motor, left-wheel motor, top-brush motor, roller brush motor, corner brush motor, vacuum motor, telescopic motor, and buzzer.

One of the most important groups of sensors is the IR range sensors around the robot. Fig. 12 shows top view of the robot. Twelve IR range sensors are mounted around the robot body; each reports distance to nearest obstacle. The range information can be used in navigation algorithm using a simple gradient descent or potential method [8]. Closer obstacles will push the robot away whereas further obstacles will pull.

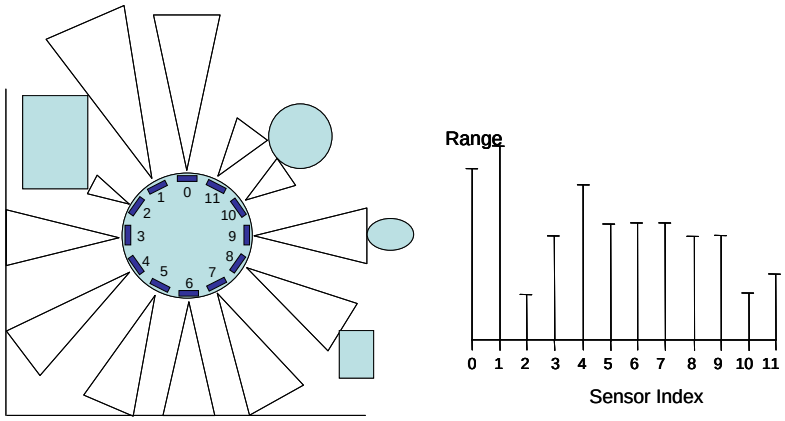



Fig. 12. IR range sensors for navigation system.

## 6 Arbitration

Arbitration uses some logics to select which behavior the processor should execute if two or more behaviors happen to trigger at the same time. There are various types of arbitration. The most-common one is called fixed arbitration. In fixed arbitration, each behavior is assigned a fixed priority. If a

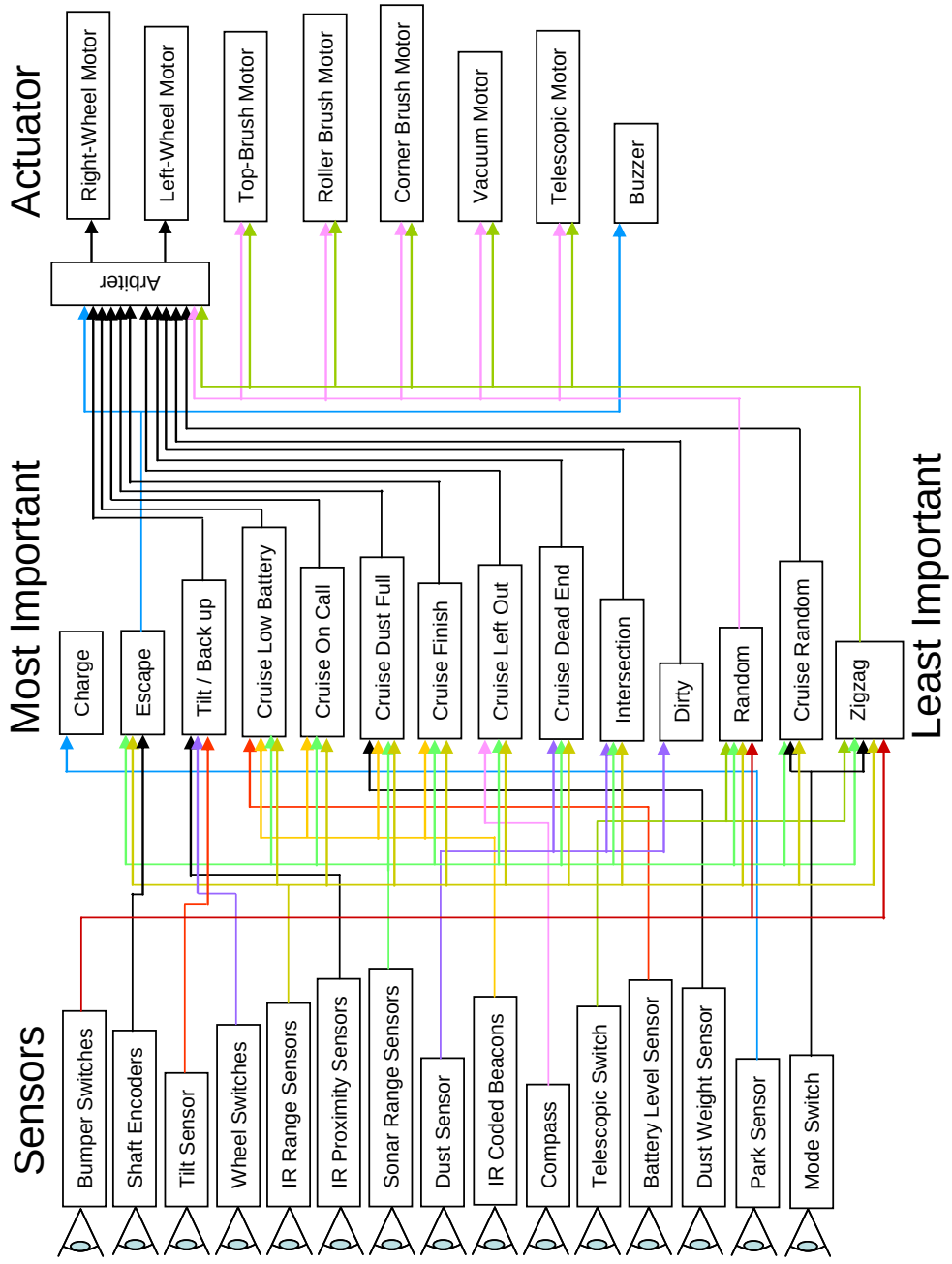



Fig. 13. Diagram of fixed arbitration linking sensors with actuators.

higher-priority behavior and a lower-priority behavior are triggered at the same time, the higher-priority behavior wins and will be executed by the processor. The second type of arbitration is called variable-priority arbitration where priority of each behavior can be changed according to a certain logic. This method is usually difficult to debug since it is difficult to link robot's behavior to what the robot actually does due to variable priority. The third type of arbitration is called subsumption arbitration [4] where some behaviors are allowed to inhibit other behaviors. However, subsumption arbitration tends to complicate the programming process. There are other arbitrations such as motor schema and least-commitment arbitration whose details can be found in [6] and references therein.

Our robot uses the fixed-priority arbitration as shown in Fig. 13. The middle column lists all of the robot's behaviors from the most important to the least important behavior. The left column contains all sensors, and the right column lists all actuators. Due to limited space, it is not appropriate to describe all behaviors as well as their links with sensors and actuators. We will then explain only the zigzag behavior and the readers should understand the rest by careful inspection of the diagram in Fig. 13.

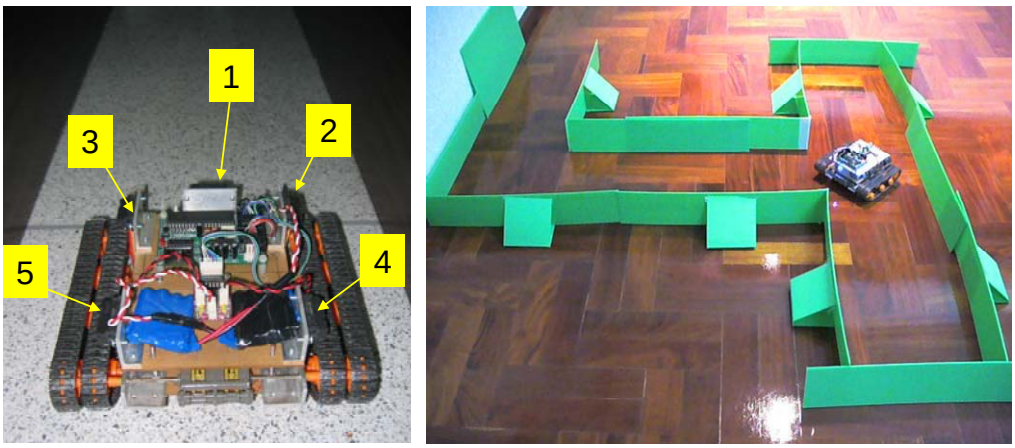
The zigzag behavior is triggered by the mode switch, that is, when the operator chooses the tidy mode over the random mode. The behavior is then controlled by the bumper switch, the IR range sensors, the sonar range sensors, and the telescopic switch. The bumper switch tells the robot when to move the telescope to scrub walls. The IR and the sonar range sensors orientate the robot. The telescopic switch tells the robot when the telescope hits the ceiling. The zigzag has the lowest priority. For example, suppose escape behavior is triggered simultaneously, the robot will perform the escape. The zigzag behavior calls for the execution of all the actuators except the buzzer.

## 7 A Prototype

Fig. 14(a) shows a prototype built at the Department of Mechanical Engineering, Kasetsart University. To measure the effectiveness of the navigation algorithm, we built a maze model to test the zigzag, random, escape, and cruise behaviors. The robot was able to perform these behaviors flawlessly.

The prototype uses a microcontroller (Microchip, 16F877) as processor. There are five infrared range sensors (Sharp, GP2D120) installed around the robot body to measure ranges from 4 cm to 30 cm, with accuracy of $\pm 1$ cm at 30 cm. There are also a compass and a wireless RS232 modem

(ET-RF24G V1.0) onboard the robot to sense robot's direction and to communicate with operator via 2.4 GHz radio wave.




(a) (b)

Fig. 14. A prototype to test navigation tasks.

It is not possible to present all algorithms in details. However, we choose to discuss some of the main algorithms the robot use in order to navigate.

During the cruise behavior, the robot tries to run in the middle of the duct. There are five sensors numbered as Fig. 14(a). In the following pseudocode, we write "Length (#)" to mean the range obtained from sensor number (#). The pseudocode is written as follows:

```
Behavior Cruise
        if Length 2 < Length 3
        go Left
        else if Length 2 > Length 3
        go right
        else
        go forward
End
```

When the robot is at a dead end, the cruise dead end behavior is triggered with the following pseudocode.

```
Behavior Cruise Dead End
        if Length 1 < 10 and Length 2,3,4,5 < 25
        u turn
End
```

When the robot arrives at an intersection, the robot must choose a way to go clockwisely. The following pseudocode explains how robot decides.

```
Behavior Intersection
        if Length 3 > 30
        go left
        else if Length 1 > 30
        go forward
        else
        go right
End
```

## 8 Conclusions

A novel air-duct-cleaning mobile robot is designed using the principle of behavior-based programming. The robot is aimed to replace or assist the traditional cleaning methods. Users should be able to operate this cleaning robot themselves, which should increase frequency of cleaning of the air-duct system. A prototype is built to test some navigation-related behaviors.

The air-duct system is a complicated physical system that includes slopes, bends, obstacles, intersections, turns, and unparallel floors. Navigating through the air-duct system is like navigating though a complicated maze. The nature of the air duct also prevents the assistance from outside such as from human or from another absolute positioning system like GPS. The robot must rely on its local sensors such as infrared range sensors and wheel encoders to navigate using some devised logics.

It should be interesting if anyone can build a full-scale robot that includes all mechanical parts proposed in this paper in order to test not only navigation capability but also operations such as scrubbing and vacuuming.

## References

[1] U.S. Environmental Protection Agency, Office of Air and Radiation, *Indoor Air Facts No. 4: Sick Building Syndrome,* 1991.

[2] K. Kreiss, “The sick building syndrome: where is the epidemiologic basis?,” *American Journal of Public Health,* No. 80, pp. 1172-1173, 1990.

[3] www.biovacsystem.com.

[4] R.A. Brooks, “A robust layered control system for a mobile robot,” *IEEE Journal of Robotics and Automation,* No. 2, pp. 14-23, 1986.

[5] R. Arkin, *Behavior-Based Robotics,* Cambridge, MA, MIT Press, 1998.

[6] J.L. Jones, *Robot Programming: A Practical Guide to Behavior-Based Robotics*, Mc Graw Hill, 2004.

[7] J.L. Jones, A.M. Flynn, and B.A. Seiger, *Mobile Robots: Inspiration to Implementation*, A K Peters, 1999.

[8] J.C. Latombe, *Robot Motion Planning*, Kluwer, 1991.