Bomb disarmer robot
Team description paper. LARC 2008. IEEE Open Category.

Hugo Cruz, *Undergraduate, UNAM* and Marco Negrete, *Undergraduate, UNAM*

***Abstract*— The IEEE Open category challenge is to assembly a bomb disarmer robot. For this competition we present the current proposal, which gives an overall idea of the algorithm we programmed and focuses on some topics we consider other teams could be interested in reading about.**

The first part of this paper describes briefly our algorithm and the material we used to build our robot. The second part is about some troubles and the proposed solutions that might interest the reader

I. Introduction

Building an autonomous bomb disarmer robot is a challenging problem. As we know there are two bombs, the first bomb of them is allocated above a ramp; the second one is allocated randomly in the competition field. Once the robot has found a bomb it must pull out the red wire and then the green one. However the bomb must remain in its base otherwise it explodes. Also it explodes if the robot pulls the wrong wire.

II. Material and algorithm

A. Material

To build our robot structure we used Sintra® plastic, brackets and screws. We use a gripper to pull out the wires; this was made with a servomotor that is attached to a couple of gears of the same size. Each gear has a nail on it, whenever the servo moves, the nails take the wire inside.

We used rubber tires with their respective motors and we also used omni directional tires to provide movement. For the vision system we used the CMUCAM3 vision sensor that allows us to track colors. We designed a printed circuit board which uses the PIC16F877A. This board provides input ports for sensors and output ports for motors and servos. It communicates with the camera via serial. We are using rechargeable batteries.

Manuscript submitted October 14, 2008. This work was supported by the Engineering Faculty of the UNAM, spanish acronym of the National Autonomous University of Mexico.

Hugo Cruz is an electronic engineering undergraduate of the UNAM, working temporarily in the Biorobotics lab as an assistant. (e-mail: cruzbarco@gmail.com).

Marco Negrete is an electronic engineering undergraduate of the UNAM, working in the Biorobotics doing his thesis. (e-mail: mnegretev@gmail.com).

B. Algorithm

Our former idea was to find the yellow bomb first, seek for the face that has the wires, disarm it and then go for the blue bomb climbing the ramp that faces the red wire and unplug it. After that, we planned to descend the ramp and climb the other side of the ramp to unplug the green wire. Doing it, the task would be accomplished. However this task approaching has some problems. The first problem is that the robot would spent many time doing the task this way, also we noticed that climbing any ramp that has 45° of inclination is such a complicated job. Also to make the robot to go from one bomb face to another is a waste of time.

We considered many solutions to these problems and after analyzing the rules finally we made a completely new strategy. The robot maximum dimension may not be exceeded before it is turned on, so this allows us to consider the use of a crane and an second tool, these components are inside our robot and after it is turned on these components would be used, but let us explain how.

Once the robot has got into the field it will go and stand aside the ramp. The robot launches its second tool above the ramp, which is 20 cm tall, so the launcher device fits very well inside the robot. Once the robot has put this second tool our algorithm splits. At the same time the robot will find and deactivate the yellow bomb, while its second tool is working on the blue bomb. We also mentioned the use of a crane, which rotates the bomb. Once we have found the yellow bomb the crane will help us to seek the face with the wires and it will be faster than sending the robot from one face to another analyzing the bomb.

III. Focused problems

A. Servomotors

As we know a servo needs a DC voltage, ground and a control signal. For the servos we used this control signal is usually a pulse with a 20 ms period, and it depends on the duty cycle the position of the servo. Usually the duty cycle is regulated with a PWM (Pulse Width Modulation) module that many microcontrollers have by default.

We used the PIC16F877A to control the servos, which has two PWM modules. In our robot these modules are used to control the spin of the tires because we wanted to have a softer control of the speed of the robot. This technique is very useful because it allows us to rotate or move faster over the competition field.

However using these modules leave no room for servos, and we need to control at least four servos in the robot.

To solve this problem we used the next proposal. We used a Timer module to cause an overflow each 20ms, this allows us to use an interruption that leaves the main process and execute the servo routine. This can be accomplished easily in the case of one servo, just imagine: we already have the period of the signal, we just need to turn on an output immediately after the interruption has begun, we must maintain it on for 0.5ms to 2.5ms according to the desired position of the servo. Once it has finished we finish the interruption and execute the main routine again. This allows us to control our robot and one servo at the same time.

How to accomplish this proposal for two or more servos? Servo signals must be sent at the same time, but we have neither enough PWM modules nor enough Timer modules. For this, we turn on different output signals at different times, like a phase control. As soon the pulse of one servo has been sent, we send another pulse until all the signals have been sent, at this moment the interruption ends and we go on with the main routine. Look at Figure 1.

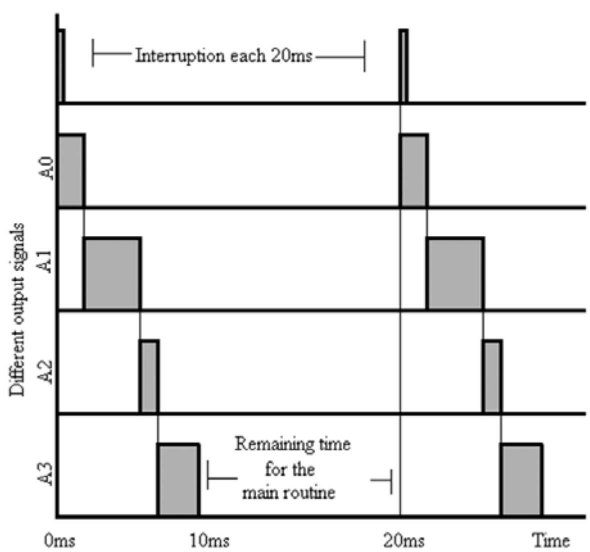

Fig. 1. Servo signals. This is our proposal for controlling servos with just one Timer module.

This technique is useful for small robot applications; however it has an important disadvantage: it consumes much main routine time depending on the number of servos that are connected and the signal that is being sent to them. For example, just imagine eight servos and all of them moved to their limit, which could be a 2.5ms signal on each one that is 20ms; this would leave no time for main routine. It is also important to consider an extra DC supply for the servos.

B. *Reaching the Bomb*

The robot must seek the bomb and diverts its direction once it finds it. But reaching the bomb is difficult because we want to reach it in a perpendicular way; it is convenient to do it this manner because it allows us to make an easier algorithm that helps us to find the wired face of the blocks if the robot reaches the bomb at the wrong side. But the robot would hardly ever reach it this way.

For solving this problem we used the next method. We used two limit sensors that are located in front of our robot. Each one of them has a width almost equivalent to the half of the robot, so all the front is capable to sense any touch.

Using these sensors we wrote the next routine. Whenever the front left sensor has been pressed the robot stops and softly turns right. Whenever the front right sensor has been pressed the robot stops and softly turns left.

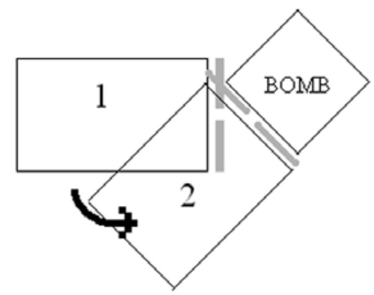

Fig. 2. Technique for reaching the bomb perpendicular to it. As soon as one sensor is pressed the robot will move in order to make the other sensor be pressed too.

As it can been seen in the previous figure, this technique will allows us to reach the bomb in a perpendicular way. However there are two problems, the first one is that occasionally the robot reaches the bomb at 45° degrees and either sensors or none of them could be pressed. Both can be solved through programming or the modifying shape of the sensor. The second problem is that this method implies the necessity of touching the bomb; this causes a bomb displacement that can be undesired.

Once we are on position we are ready to seek the wired face of the bomb. We are using a crane to accomplish this task. The crane will rotate the bomb periodically 90° degrees. This technique saves time compared with the technique that implies sending the robot from one face to another.

C. *Blue Bomb Tool*

We consider more complicated the deactivation of this bomb because it has two wired sides and it is above the elevation which has no border walls. It also present vision problems due to there is no border, so the robot can see something blue outside the field and it can try to go there and fall.

For these reasons and the ones considered in the first page we approached this bomb in a different way. As we have said we are using a second tool, that in this section we will denominate the Blue Bomb Tool (BBT).

The robot is not capable of climbing the ramp; that is the reason of launching the BBT. Once the robot has put the BBT above the elevation it will leave and it will deactivate the yellow bomb. The BBT will seek the blue bomb. As we have said it might fall, so we are using infrared sensors, so anytime it is about to fall it will divert itself.

The BBT reaches the bomb by the side of the red wire, after unplug it; the BBT will go to the left for searching the green wire face. The BBT is small enough that it can be inside the robot and it can move freely above the elevation. Once it has found the green wire it will unplug it and it will stop. If the robot stops or accomplish the task the fact that the BBT is stopped will allow us to take the decision of finalize the round or reset and try again.

D. Color issues

Color tracking is very troublesome. The first problem is color calibration, we mean; finding the right limit values for the color space it is being used. The second problem is illumination, some color spaces, such as RGB, change dramatically its values depending on illumination. The third problem is color and object discrimination, for example some green colors are very alike to black color or some object in the outside can have the same colors of the bomb.

To be honest, we have not solved any of the problems at all. So we are going to mention the solutions that we are currently trying which have been programmed or done in the robot.

For the first problem we programmed a code that gives us the mean of the color that the camera is seeing. We used the data and a tolerance for setting the minimum and maximum limits for the tracking color space. This is useful for the color of the bombs because each can cover all the camera vision, but represent a big problem when the wire color is being calibrated.

The second problem is an issue that we are trying to solve by changing the color space. We are trying to migrate our code from RGB space to YCrCb space due to this apace is less vulnerable to illumination changes. However we have not obtained a good result yet.

We are trying to solve the last problem through programming and extra hardware. Color discrimination is a serious problem when the robot tries to disconnect the green wire; sometimes the camera confuses green with black. For solving this we are thinking about using an extra illumination, like a lamp. The technique has not been proved but it might give good results. Object discrimination must be solved with programming, once the robot has found something that might be a bomb it will go towards to it. If it is really a bomb it is a good thing, however it can be a yellow dress of a little girl watching the competition field. We are going to use the sensors for solving this because the camera can give erroneous info about the bomb. For example it the robot tries to reach something yellow outside, the front sensors will give us a signal whenever the robot reaches the field limits. The program must have an “if case” combining the color tracking and the sensor signals in order to determine that it is a wall and not a bomb what has been reached.