Anomaly Detection and Correction in Line Tracing Binary Sensors and Study of Sensor Behavior from Sensor Parameters

[1]Akshay J Ghadage, [2]Aniket K Shirsat, [3]Sushilchandra R

Vishwakarma Institute of Technology, Pune

Email: akshayghadage@gmail.com, aniket.shirsat@live.in, sushil_teja@yahoo.com

Abstract— Navigation is an important aspect in many applications of mobile autonomous robotics. There exist many types of navigation tools. The most simple and widely used navigation technique is basic binary output line detection. A simple line-tracer usually takes input from binary sensors stream and directly analyze the input and actuate the motors accordingly. In this paper, we have developed an algorithm that detects anomalies in binary data from sensor arrays used in line-tracer. Anomalies in the input may be caused by change in light conditions, tuning error, defective sensors or defective surface (dark patches on surface). This algorithm corrects the anomalies, if detected. Also the sensors performance can be determined by study of sensor parameters, proposed in this paper, which help in the detection of sensors' malfunction.

Index Terms— Binary sensor, Line Tracer, line tracing algorithm, sensor parameter.

I. INTRODUCTION

Line tracers form the important part of undergraduate projects (electronics: microcontroller course, etc) and college and national level competitions. Along with hardware such as sensors and motors, the navigation algorithm also plays an important role in the complete system. Usually the input from the sensor is matched with the predefined set of conditions, and then accordingly the controller takes the decision and actuates the motors. In the outdoor environments, some sensors may get misfired; this leads to anomalies in matching predefined conditions. In this paper we have devised an algorithm that checks the input for abnormalities and also corrects the anomalies.

II. THE SYSTEM

The system consists of sensor modules, which may contain 3, 5, 7 or 8 sensors according to resolution required, that provide binary output to the controlling unit, generally a microcontroller. The sensor modules are prone to misfire, meaning the actual output expected is the complementary of the available output. For better resolution we have considered module of 8 sensors. We store data from 8 consecutive sensors in the 8-bit memory location of the microcontroller. The sensors are tuned such that they provide high level (binary logic 1) output when line is detected, otherwise low level (binary logic 0) output.

III. ANALYZING THE INPUT

The analyzing algorithm has been realized in C programming, but can easily be ported to other languages. The algorithm takes the input from an 8-bit integer and calculates the parameters total, max and big. The algorithm processes bits in sequential order from one end of stream of sensor modules to the other end. The total parameter calculates the total number of fired sensors in the array. The big parameter calculates the number of fired sensors in consequent order. The max parameter stores the position of highest fired sensor in the biggest continuous subset of fired sensors.

Algorithm:

1. Start.
2. Initialize total, max, big.
3. Traverse through each bit starting from first position.
4. If sensor is fired increment total.
5. If a consecutive sensor is fired, and bigger than old total consecutive sensors fired, update max and big.
6. Repeat till last bit is processed.

If big is equal to total, there is no anomaly in the input. If big is less than total there is an abnormality and has to be filtered out.

IV. FILTERING ALGORITHM

The filtering algorithm is only applied if there is a detected anomaly in the input data. The filtered output can be generated by using the parameters. The parameters provide the information of position of longest continuous bit stream and its length. Using this information we can generate a replica of input with filtered out anomalies.

Algorithm:

1. Start.
2. Initialize output.
3. Loop big number of times.
4. Set bit at position of max in output.
5. Set bit at position of max-1 for big number of times.
6. Output is filtered input.

V. SENSOR PARAMETERS

A. SensorRate:

$$\textit{SensorRate} = \frac{\textit{Number of MisFires}}{\textit{Total Fires}}$$

Its value lies between 0 and 1. A value close to 0 indicated the sensor is highly reliable and a value close to unity is an indication of malfunctioning sensor. A close to unity value indicates the sensor is biased, that is, its value remains high most of the time.

However a sensor can be termed as improper by SensorRate if its adjacent sensor is predominantly biased to a value of zero. A SensorRate of zero indicates that the sensor and its adjacent sensors are working properly.

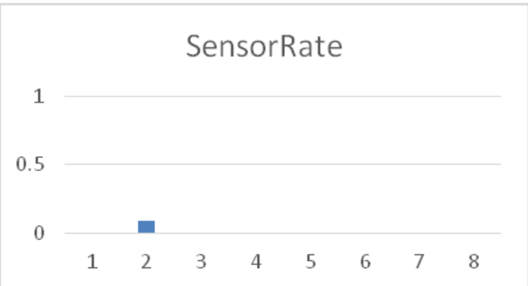
B. TotalProperFires:

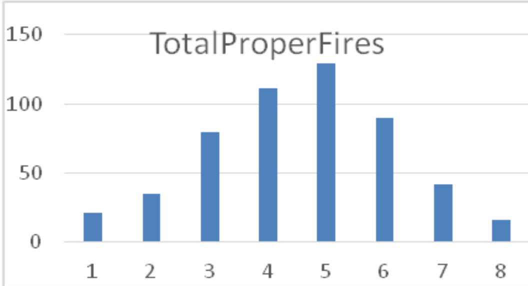
It is amount of times the sensor is fired properly during the entire run. Its value can range from null to the overflow value. Precaution has to be taken to avoid overflow.

Since the sensors in the middle are fired more number of times than the sensors at the end. Hence we usually see a graph that maximizes at the middle and tapering at both the ends.

VI. PARAMETERS CASE STUDY

Case A:

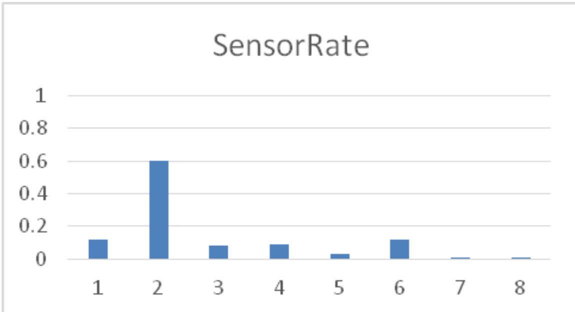


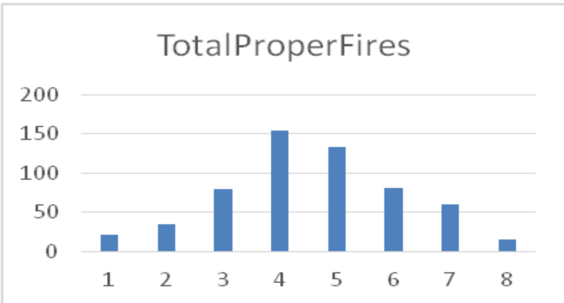


The SensorRate for second sensor is 0.09, this indicates that the sensor provided wrong value at atleast one instance amongst many readings. However for all other sensors the SensorRate is null indicating that there were no wrong fires detected. If for all the sensors, the SensorRate would have been null that would definitely have meant that during the entire course of the run, not even a single anomalous input was received from the sensors.

The TotalProperFires indicate that for the maximum number of chances the line used to be in the middle. This also means that the control mechanism of our system is working fine. The relative small amount of TotalProperFires at the corner sensors indicate that line tracer was very few times diverted from its path.

Case B:





The SensorRate for the second sensor is very high at 0.6 while others are very low. It would mean that sensor is failing numerous times. However, there has been an above average fall in TotalProperFires from fourth sensor to third sensor. The asymmetric nature of the TotalProperFires plot indicates that the third sensor has been fired less than expected. The reason behind that might be that the third sensor is not working properly, that is, it's not properly indicating the line in the sensors range. Also the high value of SensorRate of second sensor would also suggest that the neighboring sensor is not indicating properly. So here the high value of SensorRate at second place is caused by the defective sensor at third place.

VII. CONCLUSION

In this research we have successfully detected anomalies in stream of line tracing sensors. We also have corrected it using our algorithm. We also proposed the Sensor Parameters which help diagnose the sensor activity. These sensor parameters also help us predict the efficiency of our control system.

VIII. REFERENCES

[1] Hiroshi Takimoto and Yukinobu Hoshino. "Development of Fuzzy controller for the steering angle and the motor power control application to the line trace car." FUZZ-IEEE 2009 By, Korea, August 20-24, 2009

◈◈◈