CE2007 Lab2 Assignment Sheet (to be submitted to NTULearn before next lab)

Name: Do Anh Tu Lab Group: SEP1 Date: 27/9/2019

1. Section 6.2. Give a short 2-3 lines description on concept behind the reflectance reading process. Why does the black surface result in slower voltage decay?

Each IR Sensor consists of an LED emitting Infrared light and a phototransistor to receive light being reflected, which in turn converts this to electrical energy in term of conducted current and voltage. More light receives at transistor means more current conducted.

Black surface has a slower decay in voltage as black materials absorbs more infrared than white surfaces. This made the reflected amount of light to the photo transistor to be less, hence current conducted at the transistor will also be smaller. As more current Is conducted, the effective resistance at the phototransistor for black materials is lower than white materials. Hence, the time constant Tau = RC for black materials is lower, leading to slower discharge of capacitor (every tau, discharge by 1/e). Thus, the voltage decay is slower for black materials

1. Section 6.3. Where are the sources of the offset error between estimated and actual distance?

Assuming midpoint of the black line being traced by robot is used for measuring exact distance from the robot centre line.

Given there are limited number of sensors, there are limited value of reading from the sensors to contribute to the final weighted estimation, making it less accurate. In other words, if the midpoint of the black line is lying more to the left or to the right of a sensor, the sensor reading will still be the discrete step as declared in the array. This maybe improved if we add more sampling point (sensors in between the existing sensor, making the weighted sum more accurate)

For example, even if the line (midpoint of line) is 30mm away from centre of robot, because the outermost sensor (weight 33.2) is still detecting the traced line. The distance estimated will be 33.2mm.

Similarly, if the midpoint line of the black line is 5mm away from robot centre, the sensor at 4.7mm away from the centre will detect the black line (black line is more to the left of the sensor). It will returns the discrete value 4.7 and contribute it to the weighted sum => estimated is 4.7mm.

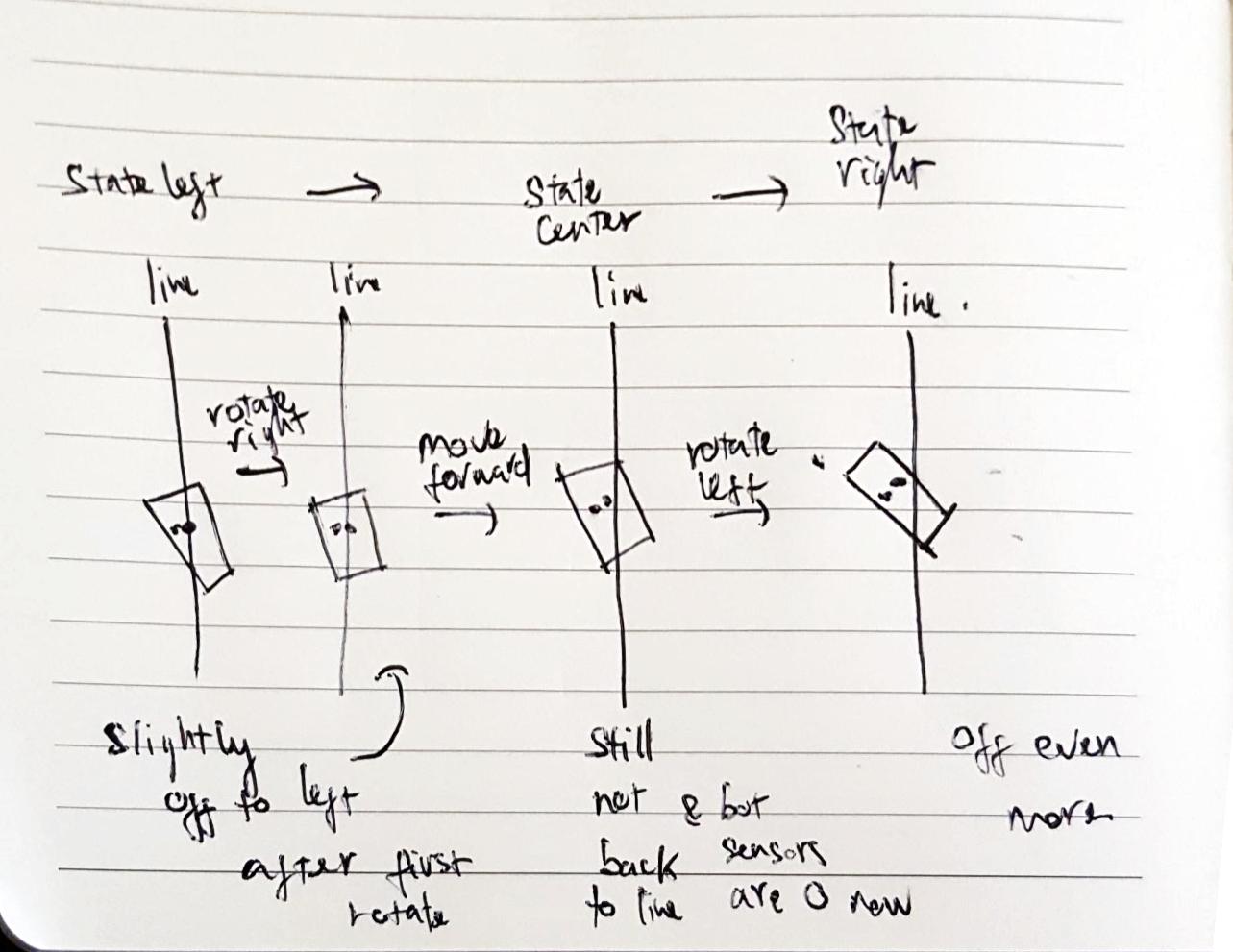
Offset error depends on the line width and also the sensor range. This is reduced if we add more sampling sensors in between existing one, making the discrete range smaller. This allow more accurate value to be contributed to the final estimation.

1. Section 7.2. Figure 7. Why is the robot being toggled between LEFT and CENTER state repeatedly when it is detected that the robot is off to the left of the line (input: ‘01’)? Won’t that cause the robot to move in a “\*RIGHT\* TURN, MOVE STRAIGHT” fashion? Since in the LEFT state, motor is supposed to turn \*right\* and in the CENTER state, motor is supposed to move straight.

There is a bug in the current design of FSM. As when input is ‘01’ (slightly off to left of line), the robot will be in Left state. If the input is still ‘01’ after sometimes, the state will transit to Center state and then transit to Left again, back and forth. This is due to the input of robot is ‘01’ when it is off to the left and input to transit from Center to Left and Left to Center are both ‘01’, thus such oscillation in state will happen as long as input is ‘01’.

In ideal case, the robot would do a right turn (in Left) and then move straight (in Center), return itself to the line it need be follow. However, if the robot is very off to the left and the right turn in Left is not able to rotate it back on the straight line so that it can move straight, returning to line following when it is transited to Center state, then the robot will move even more out of the line (to the left direction). Once this happen, the input is ‘00’ and since it is in Center state, the FSM dictates that the robot will move into the Right state. In this state, the motor moves so that the robot will make a left turn and so robot will be more off to the original line.

Refer to picture below for clearer



In short, this bug makes the robot not very deterministic in case the right turn is not hard enough.

1. Section 7.3. Sketch the FSM diagram of your design that overcome the issue mentioned in this section. Label the FSM according to that shown in Figure 7. Take a photo/copy of your sketch and paste it here.

A close up of a map

Description automatically generated

1. Section 7.3. Write down the procedure to initialise P2.4 to be an input pin with internal pull-up resistor

P2->SEL0 &= ~0x10;

P2->SEL1 &= ~0x10; // 1) configure P2.4 as GPIO

P2->DIR &= 0x10; // 2) make P2.4 as input

P2->REN |= 0x10; // 3) Enable Pullup/Pulldown for 2.4

P2->OUT |= 0x10; // 4) Pull up (=1) for p2.4

1. Section 7.3. Other than the List within List method used in the original Lab2\_FSMMain.c, which other construct is commonly used to implement a FSM?

Switch statements can be used to implement FSM easily (for small and simple FSM). However, this method is hard to debug when the number of states increases.

Rough implementation

Switch (state), where in “case ‘stateX’:” the logic of stateX is implemented. stateX is name of any state for FSM.

1. Section 7.3. What is the purpose of toggling LED within the main routine or ISR?

The toggling LED in the main routine or ISR is for debugging purpose. This method is called heartbeat LED debugging and is common in embedded software developments. This is used to us know that the microcontroller is actually doing something. If our heartbeat LED is blinking, our main program or ISR are being executed uninhibitedly. Otherwise, there is something wrong with our software such that the program is no longer looped and iterated and LED light will not toggle any more (either all off or all on).

1. Section 7.4. The center two sensors are used as the main line follower sensor, using the Reflectance\_Center() function. Describe one enhancement to this method so that it could cater for different line width.

Settings for width, we can select no. of sensor used

It is possible to add in the line’s width as an argument to this method. Based on the width of the line and given the known covering distance of each sensor/pair of sensors, we can select the number of sensor to substantially cover the line, making sure it is followed by the suitable number of sensors.