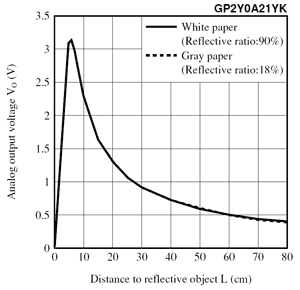
CE2007 Lab4 Assignment Sheet (to be submitted to NTULearn before next lab)

Name: Lab Group: Date:

1. Section 6.1. What is the issue when an obstacle is place to close to the IR sensor? What can you do to prevent such ambiguity?

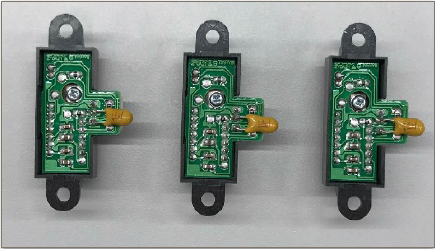
The characteristic of the IR sensors (between voltage reading (which also translated to ADC reading) and distance) is non-monotonic. That means, the sensor readings decrease with increasing distance after a threshold distance, BUT also increase with increasing distance before this threshold, as shown in the below characteristic curve:



Thus, if the object is too close to the sensor (below the threshold distance), the collected reading might be the same as the reading of the sensor at a larger distance (larger than threshold distance) too. This duality in sensor reading will be catastrophic in dead recognition application if object is mistaken to be far away while it just in front of the sensor.

There are a few ways to get around this. One way is sensor placements on the robot/ application such that the distance from the sensor to the edge of the robot is equal or larger than the threshold distance (i.e. which is 8-9 cm based on the characteristic above). The sensors readings will only then behave like the right side of the curve peak (where distance increases, sensor reading decreases). Therefore, the ambiguous reading due to blind spot will not happen.

Another way is to select the minimum distance to obtain sensor calibration equation to be slightly to the right of the turning point (for example 10 cm in this case). Any reading higher than that shall be deemed as invalid and returned a fixed value. This method also works only if the sensor placement is slightly inside (not on the edge of robot/ application being installed).

1. Section 6.1. What is the purpose of the 10uF decoupling capacitor?  
   

The decoupling capacitor is connected to the power supply of the sensors, helping the voltage reading to stabilise and reduce noises from the sensor readings. Capacitors are fundamentals element of filtering circuit.

1. Section 6.2. Which port pins is ADC Ch12, 16 and 17 input mapped to? What is the PSELx settings required to configure the pins to ADC function?

ADC Channel 17 -> Port 9, pin 0 (P9.0)

ADC Channel 12 -> Port 4, pin 1 (P4.1)

ADC Channel 16 -> Port 9, pin 1 (P9.1)

PSEL settings for Port 9 ( both SEL0 and SEL1 for pin 0 and 1 of port 9 = 1)

P9->SEL1 |= 0x03; // 8) analog mode on P9.0/A17 and P9.1/A16

P9->SEL0 |= 0x03;

PSEL settings for Port 4 (Both SEL0 and SEL1 for pin 1 of port 4 to be = 1)

P4->SEL1 |= 0x02; // 8) analog mode on P4.1/A12

P4->SEL0 |= 0x02;

1. Section 6.3. With respect to the ADC on MSP432, what are the two stages involved in every Analog to Digital Conversion of a Analog signal?

MSP432 uses 14-bit SAR analog to digital converter. There are 2 stages in the process of converting an analog signal to a digital signal: Sample and Hold stage and actual SAR ADC conversion.

Sample and Hold stage is to sample the input voltage, which is then passed to the SAR ADC for digitisation in the next stage.

1. Section 6.4. What does the function LPF\_Calc() does? What are the initial values of the buffer associated with LPF\_Calc()? Why do we need this function?

LPF\_Calc() implements a running average algorithm to compute the 14 bit ADC value after digitalized. Whenever a new data is read by sensor, it is added into the sum and then the oldest value will be removed from this sum. Then an average value is computed for the ADC reading of the sensor. This basically create a window of size = s where s is specified in main() of Lab4\_ADCmain.c. This window of fixed size will scan through the raw readings of the sensor and output a filtered, average reading of the sensor.

Initial values of the buffers associated to LPF\_Calc is a single 14-bit value obtained by reading the sensors once after a warm up “BUSY” period as described in ADC\_In17\_12\_16 function.

It is important to have some sort of filter for the sensor (such as running average in this case) to prevent inaccuracy from sudden overshooting noise (high frequency) from reading noisy sensors. This allows a more accurate value to be obtained for sensor’s ADC values, aiding us in the final linearization to estimate distance process. This is a software filter and serve sort of similar function to the hardware filter (by RC in qns 2, different in what frequency we are filtering).

1. Section 6.4. Describe the algorithm you used to estimate the actual distance based on the IR Sensor value.

Algorithm has 3 steps:

Step 1: Collect data of sensor to prepare for linearization.

* Place object (preferably black for greater accuracy) at different distance away from robot/sensor.
* Collect data (ADC 14-bit) of the sensor by placing breakpoint in main loop after 2000 samples in the while loop and close monitoring of the nl, nc, nr values at this moments (every 2000 samples).
* Obtain the ADC value of the sensors at different distance and input into Excel spreadsheet.

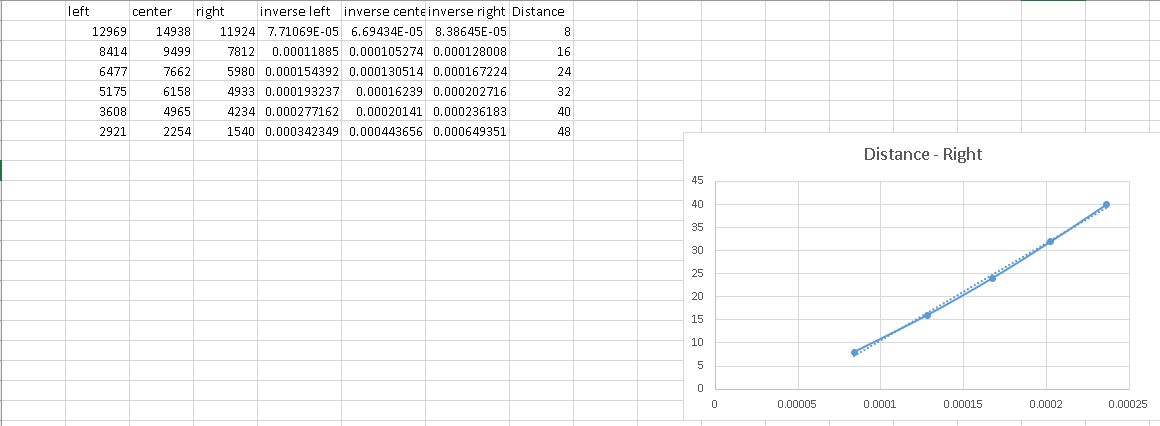
Step 2: Linearization by curve fitting in Excel or equivalent software

* Plot a graph of distance vs 1/a where a is the sensor 14 bit ADC reading.
* Graph obtained should be somewhat linear (unless before the blind spot threshold distance)
* Use Trendline function of excel to fit the graph into a linear equation. Take note of this equation.

Step 3: Implement character equation of sensor vs distance in the IRDistance.c function.

* Copy the equation obtained for distance vs nl,nc and nr respectively into the conversion functions LeftConvert, CenterConvert and RightConvert in IRDistance.c
* Provide necessary logic of coding to ensure the distance is return correctly provided an input.
* Testing and reiterate the 2 steps above if needed.

Snippet of Excel spreadsheet obtained:



1. Section 7.2. Which timer capture input (Timer and Channel number) does P8.2 and P10.4 correspond to?

Both are corresponding to Timer A3 input.

P8.2 corresponds to channel 2 of Timer A3 (TA3CCI2A)

P10.4. corresponds to channel 0 of Timer A3 (TA3CCI0A)

1. Section 7.2. Which edge (falling, rising, both) is the timer input capture configured to trigger on? What happens when a capture event occurs?

The timer input is configured to capture on rising edge of the square wave pulse from the Tachometer.

When a capture event occurs, the timer value at that moment (rising edge) will be saved to corresponding CCR register channels (0 and 2 of Timer A3 in this case for the two tachometers). This is done in the timer interrupt handlers TA3\_0\_IRQHandler and TA3\_N\_IRQHandler respectively based on lab 4 code, occurs when interrupts from Timer A3, channel CCI0A and CCI2A happens.

In these ISR, the two user tasks PeriodMeasure0 and PeriodMeasure2 (corresponding to the CCI0A and CCI2A channels connecting to tachometer output) are executed. These tasks calculate the period between two rising edges (or between 2 “ticks”) of the tachometers. From there, we can calculate the speed that the motors are rotating at and make feedback adjustments for odometry.

1. Section 7.2. Why is the calculated value of pulse duration, derived from the timer capture values, not a constant value but seemed to keep changing?

Tachometer has a light emitter and a light collector. The later upon receiving light from the former (due to light passing through little slits of the tachometer wheels which is rotating when motors are rotated) will generate a voltage value depending on how much light it receives. This in turn is converted and digitalized to either 1 or 0 to generate the square wave with rising edges to be captured in our application.

As it is a conversion from light (analog physical quantity) to voltage and hence digital value in term of square wave, there might be fluctuation in readings due to continuous nature of the light being sampled at collector, making the square pulse not to be at constant intervals between one another. Example, the light may just be ½ pass-through and half not through the slit and the light may be 2/3 pass-through and 1/3 not, these scenarios might cause different voltage readings, which may cause different value when quantized to 0 and 1. This leads to changing pulse duration as result of the time capture.