MANCHESTER
1824
The University of Manchester

School of Electrical and Electronic Engineering

Embedded Systems Project

DESIGN REPORT #2

Title: Sensor Selection and Navigation Strategy

Group Number: 18

Group members name:	ID Number	I confirm that this is the group's own work.
Charlie Shelbourne	9725297	☒
Marcell Tóth	9747325	☒
Thomas Hollis	9563426	☒
Jianli Gao	10079470	☒
Jason Brown	9582307	☒

Tutor: Dr Laith Danoon

Date: 01/11/2016

Table of contents

1. Introduction

This report aims to produce a recommendation for the navigation strategy and for the sensor array design of a line following buggy.

A high-level view of the buggy's architecture is first introduced before the line following algorithm options, applied to this specific situation, are presented. The optimal chosen strategy is then explained in detail, contrasting any alternatives considered and addressing all the design requirements. The navigation strategy is also carefully tailored to the sensor array design choices to maximise performance potential.

The sensor selection process is then explained along with the tests undertaken to ascertain optimal sensor characteristics. Specifications considered include but are not limited to sensitivity, response time, wavelength, power consumption, price, and additional features (such as light blocking casings, package leading …etc.). This choice is backed up by clear layout and schematic diagrams with corresponding simulations for assurance and clarity.

This report, in conjunction with the motor characterisation report, will provide the necessary simulation and experiment-based data to allow for a full system proposal of the line-following buggy to be developed.

2. Navigation strategy

2.1 High-level architecture

Before the line following algorithm and sensor array choices are presented, the buggy's hardware architecture must first be ascertained. This has a large impact on how sensors and their positions deal with track constraints. The chassis base plate will support four main sections: controller & status display board, motor driver board, gear & actuator section and sensor array section. These are arranged in such a way that the status display is at the top (to be visible to the user), the gears and actuators are near the wheels and the sensor array is near the bottom and sides to detect side walls, end walls and the white line. The line sensors are placed on a hinged platform to allow them to have the tightest fit with the floor to avoid losing the line even during a ramp situation and during extreme turns. They are placed at the same level as the wheels for optimal line following precision and to prevent lag between sensor feedback and steering. Other line sensor array geometries were considered such as having two rows of sensors (front row of 4 and back row of 2) for backup but this was deemed impractical and inaccurate from a performance perspective. The entire section is placed near the wheels for front wheel drive and steering, similar to cars, since rear wheel steering is inherently unstable [1]. These high-level design choices are shown in figure 1.

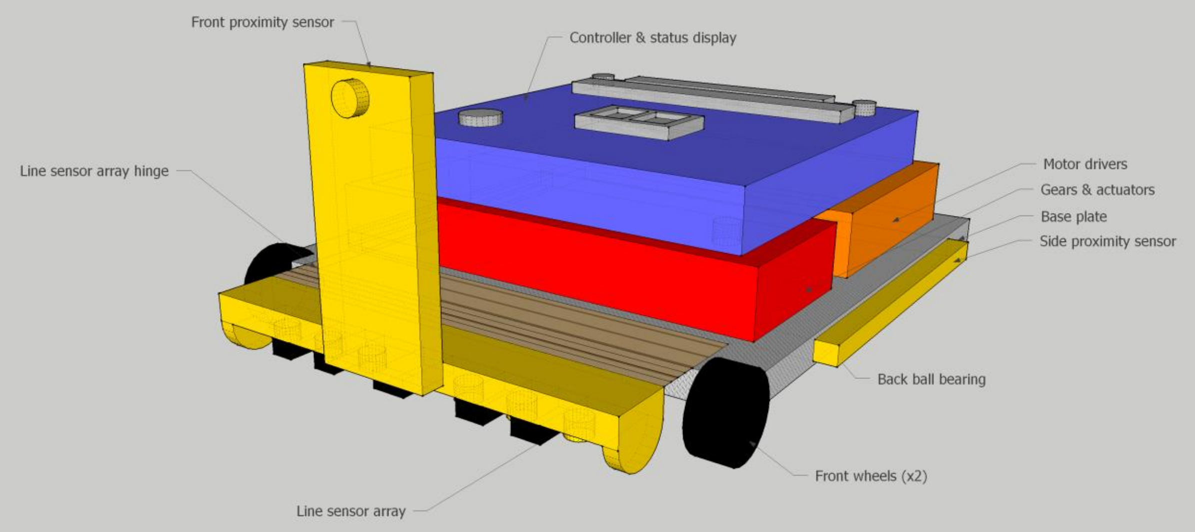


Figure 1 – High-level architecture

2.2 Review of navigation strategies

The designed buggy will be required to navigate itself in automation around the track provided, via the use of line detecting sensors, following a white line in the centre of the track. The sensors will be used to feedback data to the PIC for its steering algorithm. The track will have several obstacles including ramps, curved bends, small breaks and even fast line oscillations and the buggy must be able to overcome these. A range of control algorithms were considered to select the one most adapted to our sensor array. This choice will ensure that our hardware complements the algorithm optimally as well as deal with track conditions.

The four line following algorithms considered are: bang bang (BB), proportional control (PC), proportional integral and derivative (PID) and model predictive control (MPC). Simulations of buggy behaviour are presented in figures 2, 3 and 4 below.

BB is a rudimentary on/off control algorithm, which would cause our buggy to follow a saw tooth path about the white line as shown below. This would undoubtedly yield a poor but reliable performance. Total distance travelled would be considerably larger and speed diminished. This backup algorithm will suit the two CNY sensors (explained in section 2.3).

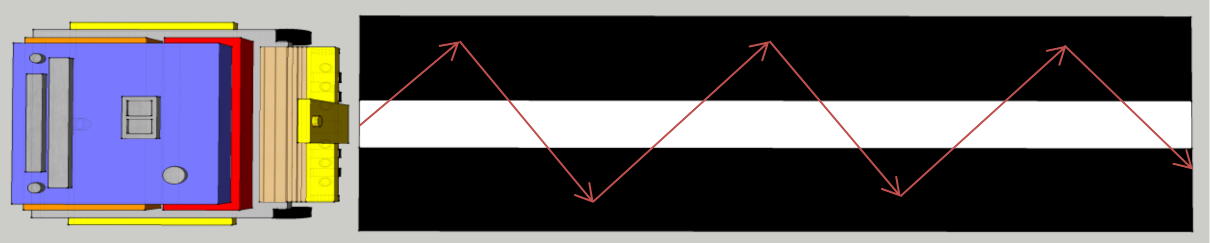


Figure 2 – BB buggy behaviour

PC on the other hand would allow for a more appropriate curve fit of the buggy to the line instead of BB’s inefficient saw tooth fit. This would mean less acceleration is required but such a wide fit could cause more line loss if a bend arises during a path wave in the same direction. This does not allow for optimal performance of any sensor used (CNY or TCRT).

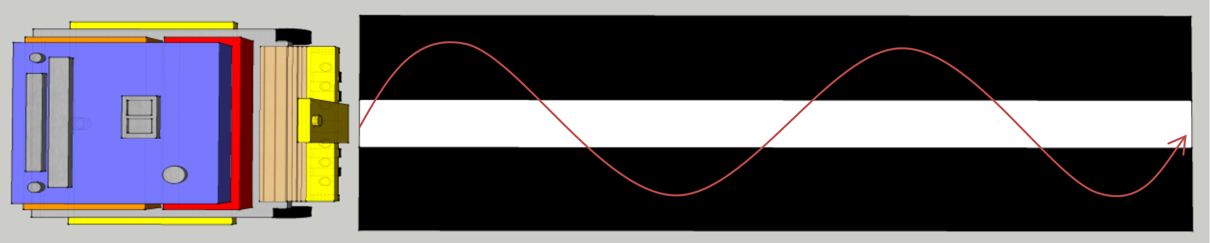


Figure 3 – PC buggy behaviour

PID would provide an even tighter curve fit with the buggy covering the smallest distance and being able to go at the highest speeds with less frequent rectification required. The white line would also be harder to lose due to the tight fit of the algorithm. This algorithm will be used for most of the navigation time by the entire sensor array of six sensors (see section 2.3).

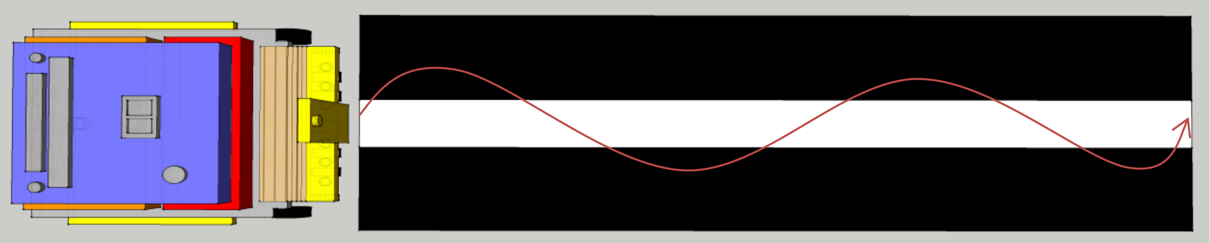


Figure 4 – PID buggy behaviour

The only way in which MPC could be integrated with the sensors would be for the buggy to remember its path’s sensor data, and drive the return path based on the way there. However, this was deemed inefficient and unreliable in case of errors in the first half of the navigation.

2.3 Details of chosen strategy

With the sensor selection chosen in section 3.1, a combination of BB and PID control was chosen as it allows for the most performant control algorithm (PID) to be used for most of the navigation, with the full six sensor array (4x TCRT, 2x CNY). This can be reduced to only two sensors (CNY) as they can provide slower BB backup navigation in case of obstacle-caused issues such as ramps, curves, fast oscillations or even track breaks. Interrupts are also implemented in software as the main solution to coping with walls, for an effective detection of the half way point, controlled stop and safe overall travel.

The general control of the buggy will be obtained by regulating a difference in power provided to the motors via pulse width modulation (PWM). Using the right-hand side motor as a reference of the left-hand side motor, power can be adjusted to be equal to, greater than or less than the reference power. In this way, the motors will provide differential steering dictated by the PIC's main line code, interrupts and its sensor feedback loop.

As an overview, our BB/PID fusion control algorithm and its 2/6 sensor array deals with the criteria required [3] as follows:

1. Follows the line with optimal efficiency – TCRT/CNY sensors provide a more sensitive line detection than others tested. Also, PID algorithm is used most of the time as it allows for tightest curve fit compared to any of its alternatives. It also allows for a continuous closed-loop feedback signal enabling the PIC to constantly monitor errors and produce self-corrections. This is only changed to BB for rare specific cases (explained below).
2. Avoids line loss and provides a backup in case of light interference – Our BB/PID dual redundant system allows the buggy to change momentarily to BB control. It does so by using only its two shielded CNY sensors (in switch mode) and excluding the other four unshielded TCRT sensors in the case of ambient light interference. This provides a failsafe system with backups to deal with unexpected TCRT readings for any obstacle. The error is detected programmatically as BB control is activated if TCRT's readings are not expected values with respect to CNY's. This will however require a percentage scaling to be implemented in software as both sensors have slightly different peak voltage values. In addition, a purely software-based solution is also generated if the line is completely lost as the buggy will be instructed to scan from left to right to find the line again if such a situation arises. To detect these errors, the sensors must be initially connected in active mode (analogue) during PID control then in switch mode (digital) during BB control. This allows a full analogue value to be read, processed and interpreted during PID for error detection purposes to switch to the backup BB switch control.
3. Deals with breaks in the line of up to 2mm – This is implemented programmatically with exceptions coded in to ignore the case where the central TCRT sensors in the middle are activated and deactivated rapidly without passing through the side CNY sensors.
4. Deals with bends of $45°$ per $50\mathrm{mm}$ – The PID control suggested causes the least line loss as it has the tightest curve fit. In addition, sensors were designed to be arranged in a line of six instead of in multiple rows for widest coverage in case of extreme bends. The distance between each sensor is set to be slightly less than half the width of the white line. This allows the buggy to cope with any fluctuations within the line as at any moment a minimum of two sensors will be detecting the line. Black light absorbing separators are also added preventing interference between sensor pairs.
5. Can stop 30cm before the half-way end wall, rotate and return – This criterion is addressed by an interrupt being generated on PB0 by the front ultrasound proximity sensor (MA40S4R) once it detects the end wall within a preprogramed distance (see 3.2 for PIC interfacing). It differentiates between end and side walls by its elevated position

as shown in figure 1. The range of the MA40S4R ultrasound front proximity sensor ($0.2 - 4\text{m}$) is well within the 30cm required so is deemed suitable for end wall detection.

6. Stays within a track width of 297mm ± 3mm without touching side walls – The extra side proximity sensors are designed to provide reliable main function breaks and steer back towards the line when a side wall is detected. The HC-SR04 side proximity sensor range of $0.02 - 4\text{m}$ is fitting for preventing collisions with side walls.
7. Deals with fast line oscillations in the track – This is implemented programmatically with an exception coded to acknowledge the case where sensors are activated on alternating sides within a fast time period. The PIC is set to temporarily provide constant power to both wheels in such a scenario relying on side proximity sensors just in case. In addition, the sensors are oriented with the LEDs front-most and the phototransistors rear-most decreasing any lag from detection to the motor's steering.
8. Deals with ramp climbs and falls of up to 15° - The high-level architecture's semi-circular hinge legs allow all the sensors to stay at a constant distance to the line even during ramps, without creating significant drag. In addition, the use of very low operational height sensors such as the CNY allows minimal line loss in elevated ramp situations. To allow the control system to cope with the navigation of a ramp, constant speed must also be maintained. Therefore, an encoder disc and photo-interrupter will be used in conjunction with the motor shaft so that the speed of the buggy is monitored. This will allow a control system to ensure constant speed in ramps by applying extra power to the motor and reverse breaking when required.
9. Stops within 20cm of the end of the start line – This requirement is met programmatically if and only if the following condition is met: buggy on its return journey AND middle sensors turn off without going past the other side sensors AND the buggy can no longer find a white line by scanning using the backup BB control. If all of those are met, the buggy is instructed to halt as it will have reached the end.
10. Processes sensor data appropriately – A hardware buffer is implemented in conjunction with a software buffer to prevent overflow in case of different read/write rates. The speed of sampling is set by the PIC's interface to the buffer preventing overflow errors. Although common-emitter circuits are more commonly used, a common-collector configuration is chosen to allow for simpler non-inverting logic. The data read from the sensors will allow for less complex programming without having to invert analogue waveforms read from the ADC. In addition, since the sensors chosen operate within a $0 - 5\text{V}$ range, no additional phototransistor voltage gain is required and ADC scaling can be kept simple.

3. Line sensor characterisation

3.1 Line sensor selection

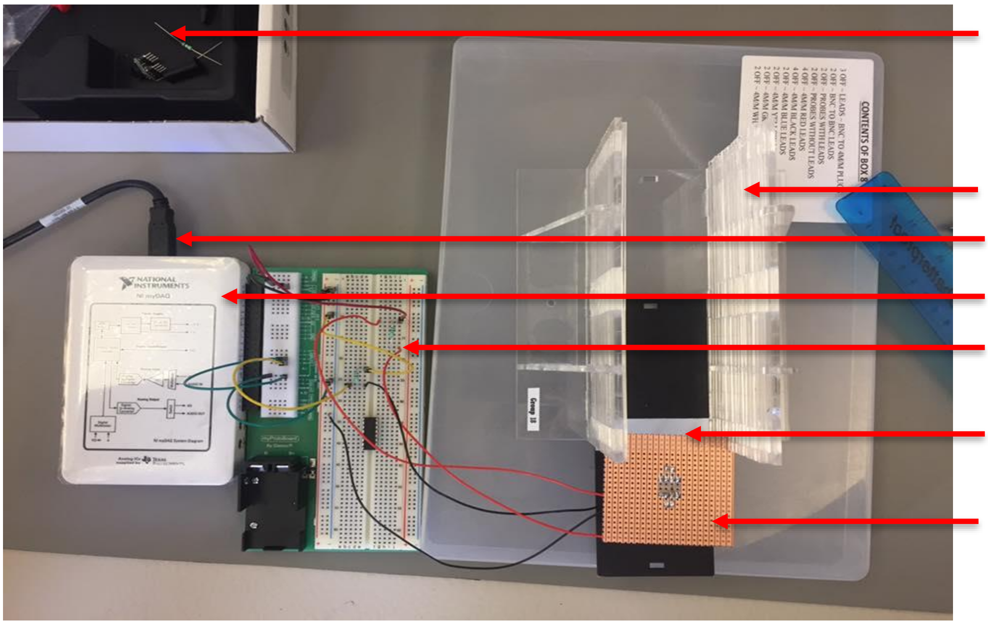
As indicated in the high-level architecture (figure 1), two main groups of sensors are required: front and side wall proximity sensors (x3) and line-following sensors (x6).

The next step is to select the sensors required to provide the algorithm with the best possible data. A sensor comparison spreadsheet was created comparing the technical specifications (sensitivity, wavelength, response time, power consumption, price…) from various datasheets to create a shortlist of sensors to test for use in the buggy's sensor array.

From datasheet comparison (via excel document) and further online research, the following shortlist of line sensors was produced: TCRT5000, CNY70, BPW17N LDR with OPE5683 LED, SFH203P LDR with OPE5685 LED.

These sensors could not be differentiated from research alone and so a thorough testing procedure was carried out to determine the optimal sensors to be used.

The shortlisted line-following sensors were tested in 3 stages [3]. Once connected to their appropriate test circuit (figure 5) the sensors were first tested with full light and no light to verify operation and approximate response times. They were then tested by varying height over the black and white sections to determine optimal operational heights. Finally, they were tested across an x-displacement over the white line at three slightly different heights to show how sensitive the sensors are and determine optimal positioning. Sensitivity was the most important characteristic to test for as the next most important characteristics such as response time and wavelength were all at suitable levels with no particular sensor providing a significant advantage over the others.



Other components

Protoboard holder

Connection to PC

NI MyDAQ^{TM} device

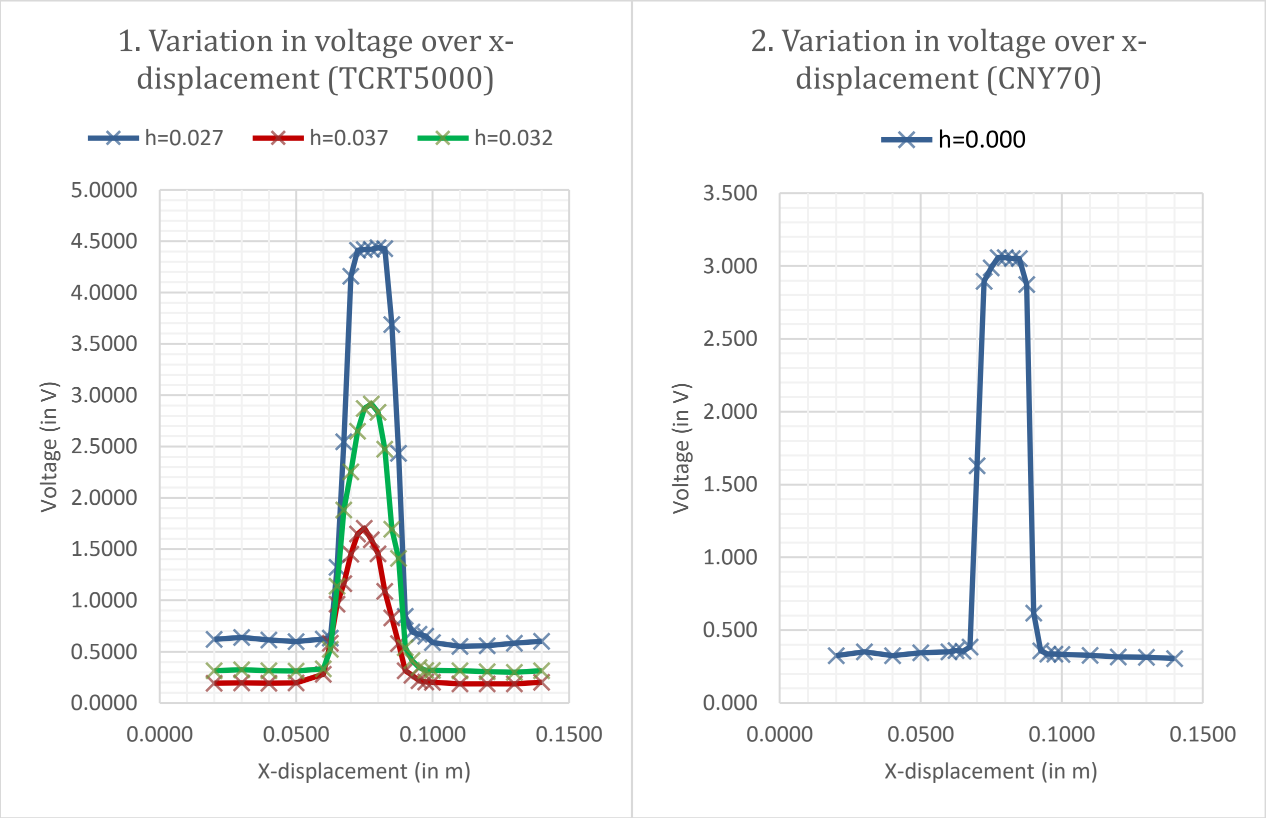
Breadboard with test circuit

White line track on black background

Protoboard with sensor under test

Figure 5 – Experimental setup for line sensor characterisation

The four chosen sensors produced the four following sensitivity curves for three heights (h):



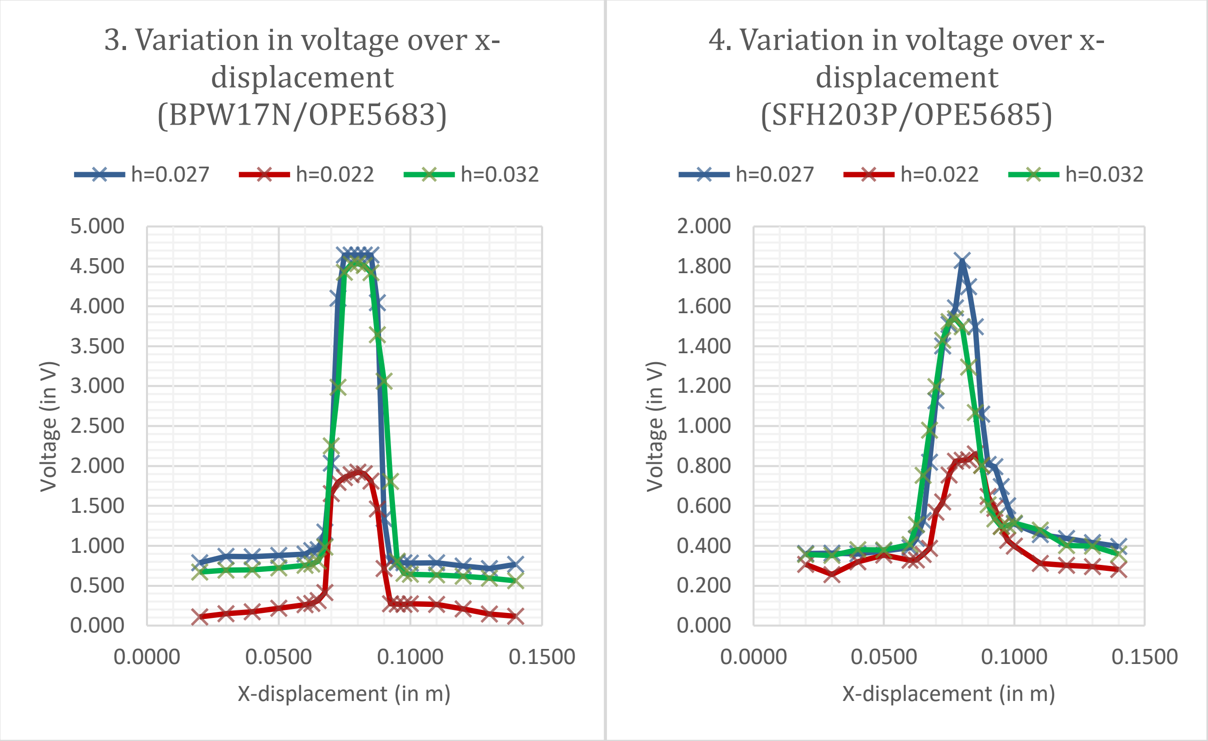


Figure 6 – Sensitivity curves for line sensors (change in voltage over x-displacement)

Figure 6 shows that the TCRT outperforms the two other sensor-LED pairs as well as the CNY as it has the largest difference in voltage from maximum voltage to minimum voltage $(3.9 \pm 0.14\mathrm{V})$ while others had differences of up to $3.5 \pm 0.2\mathrm{V}$. However, when looking at the rate of increase of voltage when crossing over the white line, the CNY performs the best as it has the steepest gradient $(4.4 \pm 0.2\mathrm{V/mm})$ while others had gradients of up to $3.6 \pm 0.2\mathrm{V/mm}$. This suggests the TRCT provides a more reliable change in voltage, while the CNY provides a more precise indication of the line's position. It is also important to note that the CNY has an operational height close to $h = 0$ (instead of the TCRT's recommended $h = 27\mathrm{mm}$). This is because the CNY comes with an anti-interference protective casing that shields the surface being sensed from external light preventing known issues [3]. These interference issues can still be a problem as even if the wavelength of light emitted from the TCRT is not in the visible range, light from the sun also contains energy within the same wavelengths. Due to these characteristics, it is recommended that the line sensor array strip be composed of four TCRTs with two CNYs, one on either side of the TCRTs. This is to have a high-precision, high-reliability, dual-redundant system resistant to interference errors as required for the algorithm chosen and presented in section 2.2. This will however require sensor output data to be scaled for each sensor as they do not have the same maximum voltage.

The main sources of random error in this experiment were due to manufacturer's uncertainty of the sensors, manufacturer's uncertainty of the myDAQ$^{\mathrm{TM}}$, number of divisions of the measuring device and environment (heat, lighting…). The main sources of systematic error could have arisen from the Veroboard holder or table not being perfectly horizontal but these are assumed to be negligible.

Proximity sensors were researched, shortlisted and characterised in a similar way but this is beyond the scope of this report.

The following sensors are therefore selected for use in the buggy:
- Side & front proximity sensor: 2x HC-SR04 [4] 1x MA40S4R/L [5]
- Line-following sensor: 4x TCRT5000 [6] + 2x CNY70 [7]

3.2 Sensor array design

The two sensor groups are arranged on four separate boards as shown in figure 1: one board with the line-following sensors, one with the front proximity sensor, two for each side proximity sensor. These boards are connected to the PIC as shown in figure 7.

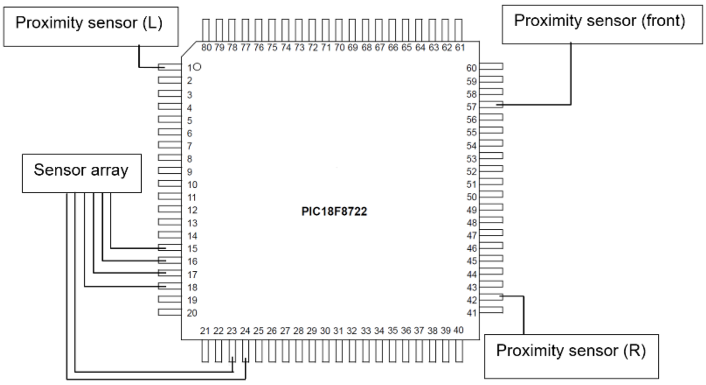


Figure 7 – Sensor array layout

The front proximity sensor is connected to PB0 so that it can be configured to issue an interrupt while the other proximity sensors are connected to spare IO on RJ7 (R) and RH2 (L). The line sensor array is allocated to the analogue-capable IO on port F (RF0-RF6). This will allow the sensor array to be connected to the ADC. The inner connections of the line sensor are shown in figure 8 with netlabels for power and PIC interface connections for easy routing. Note that the use of the PIC's on-board ADC and relevant pin configurations should be set up to convert the sensors' analogue output into digital readings. A buffer may also have to be written in C to cope with hypothetical differences in sample rates highlighted in section 2.3.

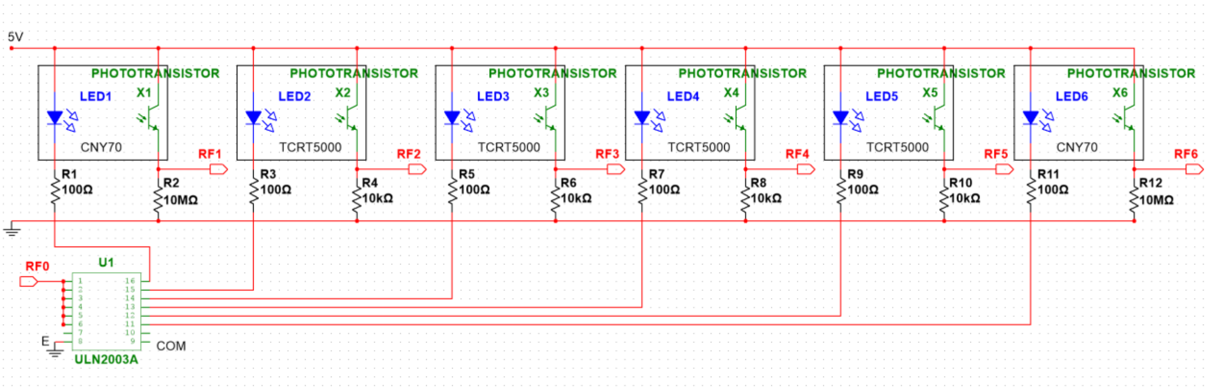
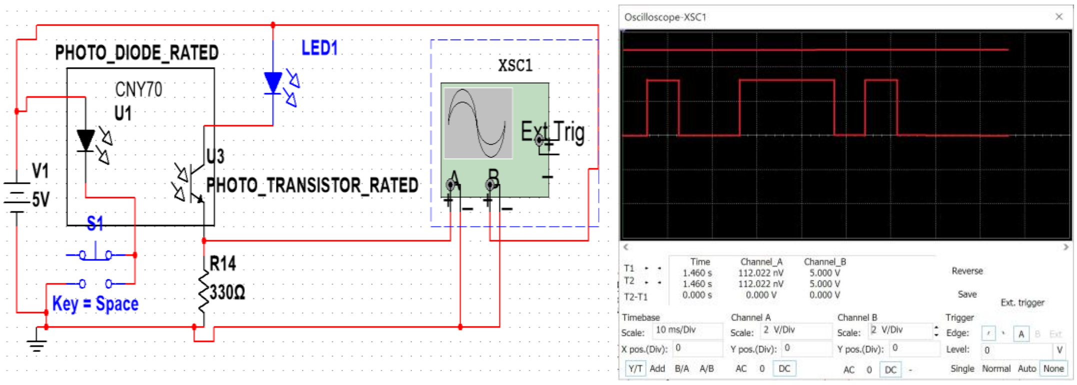


Figure 8 – Line sensor array schematic

Here we can see the 6 sensors all connected in a line to the PIC's 5V voltage rail and ground rail with the 4 TCRTs in the centre and the 2 CNYs on the edges. The LED parts of the sensors are connected to their respective resistors for current regulation while the phototransistors are connected to theirs. The RF0 pin of the PIC is used to enable or disable the ULN2003A buffer connected to the sensors in order to only turn them on when a reading is required, saving power. Lower power consumption may allow us to go for a smaller battery decreasing weight and increasing overall performance. The line sensors are in common collector arrangement as recommended in section 2.3. Therefore, the sensor array's interface wires connected to the PIC (RF1-RF6) are connected between the phototransistor and the resistor to be able to determine the voltage drop across the phototransistor which will serve as the PIC's main source of control data along with proximity sensor data. This proximity sensor data is again beyond the scope of this report.

3.3 Sensor array simulations

The sensor array above is simulated to gain an understanding of the input data we can expect in the microcontroller board and to ascertain the power flow around the system. It also serves as a proof of concept to support our working design. The simulation is setup using a standard photodiode and phototransistor-pair in NI MultiSIM^{TM} as shown in the schematic in figure 9. The switch is opened and closed to simulate the sensor going over the reflective white line or non-reflective black line.



	Time	Channel_A	Channel_B
T1	1.460 s	112.022 nV	5.000 V
T2	1.460 s	112.022 nV	5.000 V
T2-T1	0.000 s	0.000 V	0.000 V

Timebase Scale: 10 ms/Div X pos.(Div): 0 Y/T

Channel A Scale: 2 V/Div Y pos.(Div): 0 DC

Channel B Scale: 2 V/Div Y pos.(Div): 0 DC

Trigger Edge: A Level: 0 V None

Figure 9 – Circuit simulation schematic with output voltage over time with open/closed S1

The voltage-time graph produced shows ideal behaviour with instantaneous reaction from the phototransistor (reaching $V_{max} \sim 3.5\text{V}$). It is important to note this performance will be slightly delayed by the non-ideal components as seen previously in figure 6. This simulation also assumes a white line will provide 100% reflection (like the direct light source in the simulation) while a black track will absorb 100% of the light giving no reflection. Of course, this is not the case for the different surfaces such as an imperfect track and despite many more assumptions made in this simulation, it still supports the proposed designs on a theoretical level.

4. Conclusions

This report recommends using a combination of BB and PID control algorithms for a failsafe dual-redundant line following algorithm. The following choice of sensors was based on datasheet comparison, experimental analysis and integration with the navigation strategy: side & front proximity (2x HC-SR04, 1x MA40S4R/L), line-following (4x TCRT5000, 2x CNY70). The navigation strategy was thoroughly analysed and shown to have the potential to deal with all the issues requested through a variety of software and hardware methods. An optimised sensor array layout of six inline sensors has also been selected for minimised line loss and thoroughly verified by simulation.

Choice of line sensors	4x TCRT5000, 2x CNY70 (RF0-RF6)
Choice of proximity sensors	2x HC-SR04, 1x MA40S4R/L (RH2, RJ7, PB0)
Arrangement of sensors	6x inline array (4x TCRT in middle, 2xCNY on side)
Design of sensor interface circuit	Connection of sensors to PIC & power via ULN buffer
Navigation strategy	Combination of PID and BB (dual-redundant, failsafe)

The main assumptions made are that the weight of the line sensors is too small to be of relevant impact to the buggy's operation. It was also assumed that the power consumed by individual sensors is not a priority as they are all roughly similar compared to the high-power consumption of the DC motors. From simple worst case scenario calculations, our buggy should be able to last the length of the proposed track even with the most power-demanding of line sensors. However, it is important that these sensors, as an array, aren't running constantly so a buffer was implemented to only turn the sensors on when a reading is required.

The main sources of error were due to setup positioning and machining as well as manufacturing uncertainties and environment but this did not significantly impede on results and the suggested design remains robust to an acceptable degree of certainty.

5. References

[1] Whitehead, J.C. (1990) ‘Rear wheel steering dynamics compared to front steering’, *Journal of Dynamic Systems, Measurement, and Control*, 112(1), pp. 88–93. doi: 10.1115/1.2894144.

[2] Podd, F, Apsley, J, *Embedded Systems Project: Project Handbook*, School of Electrical and Electronic Engineering, The University of Manchester, Sept 2016.

[3] Podd, F, Apsley, J, *Embedded Systems Project: Technical and Management Manual*, School of Electrical and Electronic Engineering, The University of Manchester, Sept 2016.

[4] Elec Freaks (2016) *Ultrasonic Ranging Module HC-SR04 Datasheet.* Available at: cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf (Accessed: 19 Nov. 2016)

[5] RS (2016) *Pro Ultrasonic Sensor Barrel PCB MA40S4R Datasheet.* Available at: uk.rs-online.com/web/p/ultrasonic-proximity-sensors/2370783/ (Accessed: 19 Nov. 2016)

[6] RS (2016) *TCRT5000 Datasheet*. Available at: uk.rs-online.com/web/p/reflective-optical-sensors/8187524/ (Accessed: 18 Nov. 2016).

[7] RS (2016) *CNY70 Datasheet*. Available at: uk.rs-online.com/web/p/reflective-optical-sensors/7085348/ (Accessed: 18 Nov. 2016).