University of Southern Queensland

Faculty of Health, Engineering & Sciences

Time-optimal Controller Design for An Auto TCD Probe

A dissertation submitted by

Kieran Higo [redacted]

In fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Electrical Engineering

October 2021

Abstract

Transcranial Doppler (TCD) ultrasound (US) is a form of non-invasive medical imaging primarily used for measuring cerebral blood flow velocity (CBF-V). Velocity of the flow of blood in the intracranial arteries during dilation and constriction. By placing the TCD probe at the thin bone windows of temporal bone, the juncture of the frontal, parietal, temporal, and sphenoid (Naqvi et al. 2013), low frequency US waves in the order of 2 MHz or less, are used to produce a high-resolution imaging of CBF-V and vessel plasticity. Whilst relatively inexpensive the performance of manual probes is highly dependent on the operator, with auto probes scanning the entire brain. Both methods require considerable time resulting in discomfort for the patient and consuming resources of the user.

Proposed designs implement several algorithms to automatically scan and generate imagery. The initial algorithm is a global search of the Circle of Willis within the human brain, identifying targeted blood vessel sections (Huang et al. n.d.). Once the global search is complete, a local search is initiated to increase the accuracy of search outputs by examining the true or false signals from the global search, acquiring a continuous and stable signal spectrum.

A controller that could shorten the time required to perform ultrasound of the brain will reduce discomfort for the user and increase the availability of the resource, increasing the number of patients that can be serviced further improving the efficiency of the device (Huang et al. n.d.). This dissertation aims to simulate an auto TCD probe and reduce the time taken for these algorithms to perform searches and produce imagery of the CBF-V. The algorithms are the Decision Tree, Naive Bayes, Equal Interval Search, An Unknown Algorithm and the Golden Section Search.

University of Southern Queensland

Faculty of Health, Engineering and Sciences

ENG4111/ENG4112 Research Project

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Certificate Of Dissertation

I herby certify the that the following work is completely my own, except where otherwise indicated, acknowledged, and referenced.

I further certify that this work has not been submitted or used elsewhere, unless specifically stated.

Kieran Higo

[redacted]

Acknowledgements

I would like to acknowledge the support of my project supervisor Dr Bo Song for his guidance and assistance through this project, as well as Professor Belal Yousif for his guidance through the course.

Also, to my family who make the ultimate sacrifice of time.

Kieran Higo

Contents

Introduction

The following chapter outlines the justification for the project, in where it is needed as well as resource requirements. The overall objectives and how the outcomes will be evaluated.

1.1 Project Justification

Automation has an increasing presence in every industry including healthcare. Increases in efficiency, reduction in costs and less human generated errors make the option of automating processes appealing and perhaps necessary to be competitive in the industry. Studies from Robert Schier, MD (Guilford-Blake 2018), predict that all medical imaging will be performed by machines within the next 20 years. Once the implementation of automation is complete, the focus will be to increase the efficiency and accuracy of these devices.

Errors occurring in ultrasound can range from five to nine percent of all scans, leading to re-scanning or at worst misdiagnosis of medical conditions. With millions of scans performed per year this presents a significant number of human induced errors. This can be attributed to the skill of the operator; this varies between operator's dependent on training and experience which does not guarantee uniform care for patients. Whilst other factors as age, sex and past injuries can affect the temporal bone thickness which limits the accuracy of the TCD due to a limited trans-temporal acoustic view. These factors are unavoidable when assessing the viability of TCD US results (Dr Andrea 2016).

Elements of the TCD procedure that can be controlled are equipment design and operator application and training. Replacing the human interface could reduce the induced error into TCD results. There would be consistency with scanning procedure, with multiple units afforded high quality accurate scans with reduction in misdiagnosis and multiple scans. The areas that can reduce the time taken for scans, lies predominantly within the signal processing, particularly within directing the local search function, which is the crux of this dissertation.

Typically, a TCD US procedure can take up to several hours. Manual scanning will require the patient to remain motionless on their back while the radiologist holds the probe against the patient's temple. The difficulties associated with this are for the patient to remain still for extended periods whilst having the probe applied to the temple and exposed to noise generated by the device. Issues created for the operator are repetitive strain injury if not managed correctly coupled with issues locating and identifying the required cerebral arteries. Auto probes place the same discomfort on the user as manual scans.

Reducing the time taken will consume less of the patient's time and exposure to discomfort coupled with providing more accurate results.

1.2 Project Objectives

The overall objective of this dissertation is to present background information about the auto TCD probe, how it acquires data and uses this to produce a medical image of the cerebral artery blood flow. There are several components that are required to complete the entire process, which are explained briefly in the following chapter. The dissertation will be focused on the step of data acquisition and imaging process. This is how the maximum blood flow is calculated from the global search algorithm data.

I. Initiate background research on Transcranial doppler ultrasound (TCD), including past, current, and conceptual designs.
II. Simulate controller model using MATLAB capturing relevant data for benchmarking.
III. Assess data for optimization areas.
IV. Prioritize optimization points select areas that will give the greatest performance increase.
V. Create an optimised controller model that locates the maximum blood flow using the least number of steps.
VI. Process and evaluate experimental data.
VII. Evaluate the effectiveness of the new controller.
VIII. Provide recommendations for any further improvements.

Specifically, the objective is to develop an algorithm that uses the data from the signal plots and locates the maximum blood flow within the given search area.

1.3 Outcome Evaluation

Ultimately the outcome that will be assessed is the comparison of several local search algorithms, the number of steps each uses to evaluate data, including lines of code and number of loop iterations. And which method is the most efficient in producing the maximum blood flow velocity in the localised area.

1.4 Resource Requirements

Simulation of the auto TCD probe controller model will be implemented in MATLAB Simulink. Raw data input has been provided by USQ, which is raw data taken from a TCD probe. This has been gathered using finite element method (FEM), covered in section 4.2. This data will be used to implement the second stage of the controller producing the result.

Table 1.0 Resource Requirements

Item	Source	Costs AUD$	Comment
MATLAB	Student	59	Already Purchased
SIMSCAPE	Student	12	Already Purchased
Microsoft Office	Student	25	Already Purchased
Datasheets	Internet	0	Datasheets.com
Datasheets	Delica/USQ	0	Provided by USQ
Datasets	Delica/USQ	0	Provided by USQ

1.5 Chapter Summary

This introductory chapter has given the broad objectives of the research project, justification, and limitations of current design. Presenting the need of an optimised TCD controller. Clearly defined project objective steps, acting as guide for the project.

Background Information

This chapter covers the information on the Trans Doppler Ultrasound, and which algorithms can be used to process the raw data, it is provided to help understand where the algorithms researched work in the system.

2.1 Background

The following chapter will cover general and technical aspects of the TCD probe. Short examination of the TCD probe and the function of the device. Including physical components, patient, and operator interaction. The auto TCD probe will be described in further detail, with design of the units driving the probe and software design, particular attention will be given to the algorithms used to process and analyse signals from the probe.

2.2 General Background

TCD is used as a non-invasive approach to measure the cerebral blood flow velocity in the major intracranial arteries (Naqvi et al. 2013). The TCD probe is positioned at the thin bone windows of temporal bone, the juncture of the frontal, parietal, temporal, and sphenoid (Naqvi et al. 2013), low frequency US waves in the order of 2 MHz or less, are used to produce a high-resolution imaging of CBF-V and vessel plasticity. This form of ultrasound compact design makes it portable coupled with relatively inexpensive offers considerable advantage over other forms of US, its use in constant bedside monitoring for intensive care applications is an attribute of these advantages.

Limiting the effectiveness of the TCD is the dependency on operator skill. Long training programs and significant practical training are required for adequate results, coupled with the complexity of understanding the cerebrovascular anatomy in the third dimension (Naqvi et al. 2013). Some applications of the TCD are used in brain stem death, head injury, cerebral pressure autoregulation and subarachnoid haemorrhage. It is adaptable to adults and children and has been in use since 1982.

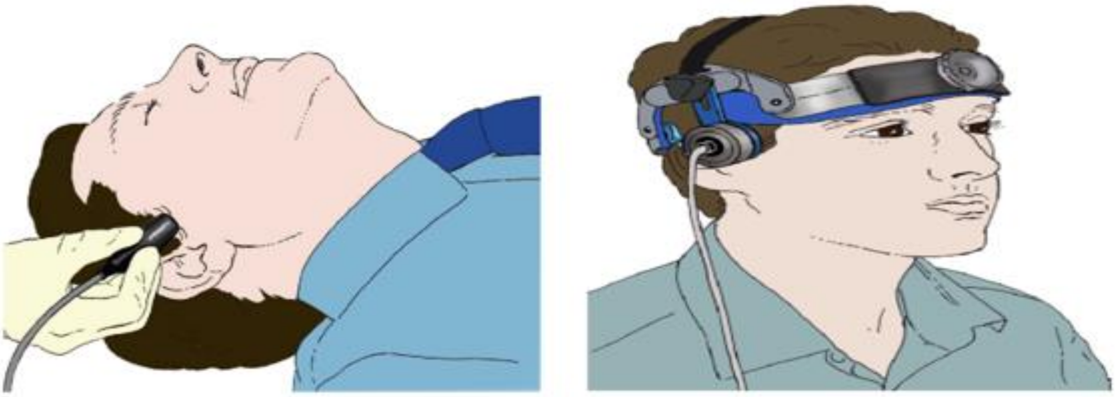


Figure 2.1 Manual TCD (left) and Auto TCD (right) application for patient use (Naqvi et al. 2013)

Based on the doppler effect, where a sound wave strikes a moving object. The reflected waves frequency will be changed, this is referred to as a doppler shift (fd) and the change in frequency is directionally proportional to the velocity (v) of the moving object (Naqvi et al. 2013). Derived in *Figure 2.1*, (c) is the speed of the wave, (fo) the incident pulse frequency and (θ) is the angle of the moving object relative to the TCD probe.

$$v = \frac{(c \times f_d)}{2 \times f_0 \times \cos\ \theta},$$

Figure 2.2 Equation for calculating CBF-V (Naqvi et al. 2013).

The TCD emits and receives waves and uses the received signal to calculate fd and *v* to produce a spectral waveform. This waveform is used to produce medical imaging *(Refer Figure 3)* of the cerebral arteries within the brain.

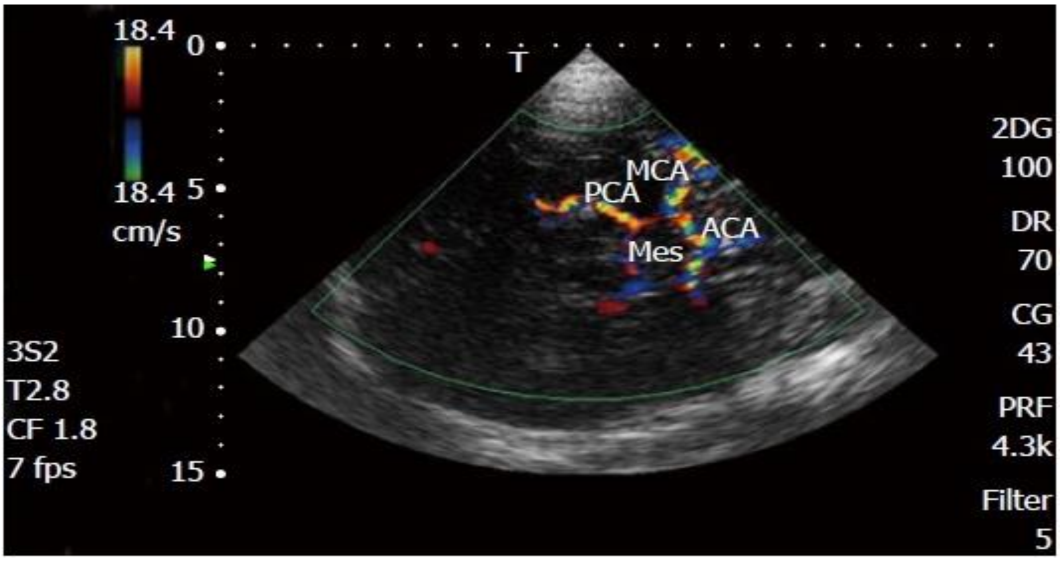


Figure 2.3 Imaging of the cerebral arteries peak systolic velocity (PSV) and end diastolic velocity (EDV) (Drsquo & Andrea 2016)

2.3 Technical Background

Several components for an Auto TCD Probe will be examined, with the main interest being of the algorithms that process the global search data to locate the maximum blood flow velocity. Currently locating a detailed examination of the entire system is proving difficult, intellectual property protection is understandable for working systems. However even with course descriptions of Auto TCD probes, various systems overviews follow a similar flowchart.

i. Automatic scanning of the entire Circle of Willis
ii. Processing of the data, numerically representing the cerebral arteries.
iii. Scanning areas returning a strong signal to improve the US image.

iv. Classification of signals to produce the different arteries and velocity of the cerebral arteries.

The following information is drawn from an auto TCD probe design entitled *“An Auto TCD Probe Design and Visualization”* by Yi Huang, Paul Wen, Bo Song and Yan Li.

An auto TCD probe will position individual transducers to scan and locate cerebral arteries without human intervention (Huang et al. n.d.). There are two main components that make up the auto TCD probe. The first is the harness that holds the assembly together, this also holds the devices to the patient’s head *(Refer Figure 1)* and keeps the probes located correctly on the temporal window.

The second component is the TCD US probe, using a 2Mhz frequency to emit ultrasound waves. The device emits a single ultrasound wave at a time. Controlling deflection angles of the waves are two servo motors positioning the transducers in the X and Y axis, with the Z axis being the direction of the wave (Huang et al. n.d.).

The fundamental frequency of the US probe is 2MHz. This frequency will allow scans of depths up to 80mm with an acceptable level of attenuation to the signal. The design only scans a single side of the Circle of Willis through the left transtemporal window as in theory it is symmetrical. Scanning depth of 80mm will cover one side of the scanning window up to the centre of the cerebral arteries. Sampling depths will range from 40mm to 80mm, with a step size of 5mm. The deflection angle of the two servo motors is 1 degree with the range from -70 to 70 degrees with a resolution of 1 degree. For each servo motor this will present 141 positions, giving 19881 points in the cartesian plane at 5mm increments from 40mm to 80mm this presents a total 178929 measurements.

The first algorithm for controlling the positioning of each servo motor is named Global Search Algorithm. This uses the US to scan the Circle of Willis within the brain to locate target cerebral blood vessels. According to MRI and MRA procedures, obtaining the position of each cerebral artery, then using this data with the finite element method (FEM). Giving an accurate location of each artery inclusive of the location of the ultrasound window in the skull.

Initially the processing of the FEM data and origin coordinate translation is required. The initial data point of the FEM has its origin based at (0,0,0) X = 0, Y = 0 and Z = 0. Once data has been collected and processed the origin can be manipulated to the left side of the temporal window. For each axis there will be 118 FEM points. The central point of the left ultrasound window can be found by taking the average of these 118 points for each axis. With the central window defined the origin of the coordinates can be transformed to produce the vector key, in the form of $(x0,\ y0,\ z0)$, all the coordinates of the cerebral arteries need to be transformed using the conversion vector. Once complete the coordinates of the arteries can be repositioned based on the coordinates of the ultrasound window. Finally, the depth of ultrasound wave and deflection angles of the X and Y servo motors is set, controlling the reflection angles of transmission (Huang et al. n.d.). with 141 detection points for each motor, producing 19981 detection points per scannable depth.

The global search's purpose is defining the focal zone of the ultrasound at a specified depth. After the search has completed the Circle of Willis at a certain depth, the K-nearest neighbour search (KNN) is used in this case to find the nearest point in scale space, using echoes generated by arteries from the US. By calculating the Euclidean distance between the detected US signals and the near sample points of arteries will give an indication of the strength of signals, in this case 10597 FEM points are used to construct the arteries. Next the distance between test signals and samples from all scanning signals is calculated.

The second algorithm is the local search approach, this is to improve the accuracy of the searching outputs. It takes the strongest signal from the global search, then using the two equations in *figure 2.3*, creates the deflection angle for x y respectively. With the deflection angles calculated, the scan changes the deflection angles of x and y along with the servo motors to scan within a circle with x0, y0, z0 been the centre point. The 10 closest neighbouring points are calculated using the KNN as are the Euclidean distances of these points. This will ensure a stable signal spectrum is established as the signal from the global search can be intermittent with exception to certain areas providing continuous signals. It will also aid in confirming if the strong signal from the global search is to be excluded from the global search as the local search further examines the true or false signal.

$$\theta_y = -(\arcsin(\frac{y_1}{D})) \times 360°/2\pi$$

$$\theta_x = -(\arctan(\frac{x_1}{z_1})) \times 360°/2\pi$$

Figure 2.4 Deflection and for X and Y of detected signal (Huang et al. n.d.).

Classifying the signals uses the KNN method, *(reviewed in detail in section 3.2)*, and is used as the reference algorithm in this project. Firstly, the distance between a detected signal and each relevant data point, then sorted as the relationship and distance increases. Next the points with the least distance are compared with the initial K points. Finally, the category of K points with the highest frequency before the extrapolation of classification of data.

2.4 Chapter Summary

The Auto TCD probe scans the area of the Circle of Willis at a set frequency at different distances to map a set of data points, which are then extrapolated to provide a more detailed representation of the cerebral arteries.

The positioning of the TCD, servo motor deflection and to a limited extent the global search function is almost arbitrary for this report. The intention is to give a reasonable description of the system. The main interest is the results from the global search and the second search and how the data is processed. In this case the KNN algorithm, which will be reviewed with the Decision Tree, Naive Bayes and the Golden Section Search algorithms in the literature review.

Literature Review

The following chapter covers several algorithms that will be potentially used in the project, providing detailed analysis of how the algorithms are compiled and how they are used for function optimisation.

3.1 Literature Review

The following section is a review of various sources of information reviewing differing methods to identify the maximum values in the results of the global search function, including methods for interpolation. Also, there is a general review of algorithms in the physical scanning of the cerebral arteries.

3.2 Finite Element Method (FEM)

FEM is a numerical method for solving problems of engineering and mathematical physics. It is used in problems that have complex geometries that require numerical solutions as well as loading and properties of materials. Common applications include Structural and stress analysis, fluid flow, heat transfer, electromagnetic fields, and acoustics modelling. The crux of the method is that many engineering problems can be defined governing equations and boundary conditions (Kim & De Weck 2004). Taking any of the systems, the governing or differential equation coupled with the boundary condition will provide a numerical representation of the system.

The FEM takes the model required for analysis and divides it into a system of many smaller models or bodies. These smaller bodies are connected at two or more common points referred to as nodes, boundary lines or surfaces and can be represented as a simultaneous algebraic equation.   (*Introduction to Finite Element Analysis (FEA) or Finite Element Method (FEM)* n.d.)

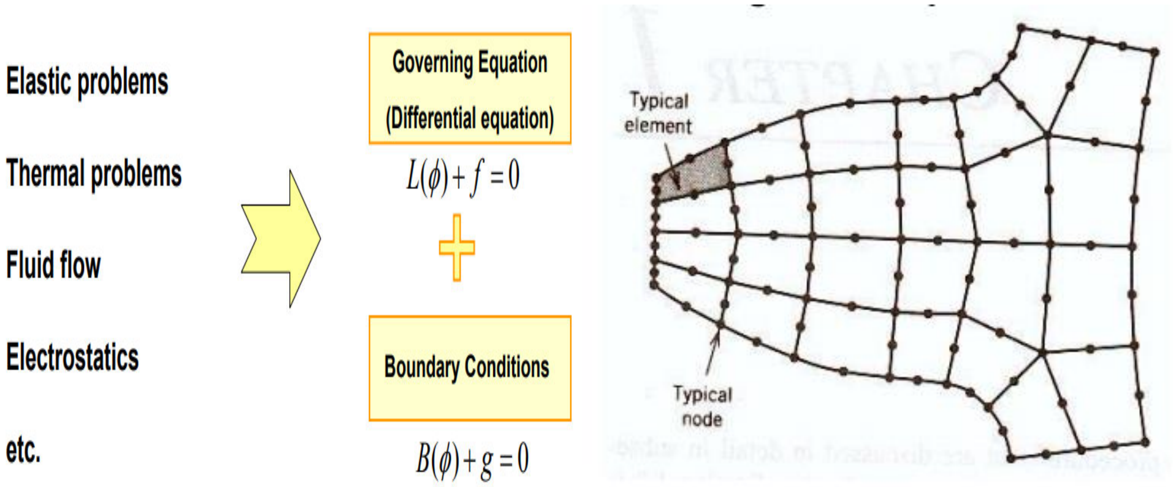


Figure 3.1 - Numerical Conversion of Engineering Phenomenon (Olivier De Weck, Il & Kim 2004).

The method can be defined in three routines with several subroutines to form the numerical solution. The first is pre-processing, where the geometric domain, element types, material properties, geometric properties, element connectivity and physical constraints. Secondly is the solution routine, where the unknown variables of the primary field are computed. These values are then used by back substitution to compute additional derived variables such as element stresses and flow. The final routine is post processing where sophisticated processes are used to sort and plot the relevant results from the finite element solution (Olivier De Weck, Il & Kim 2004).

3.3 K Nearest Neighbour Algorithm (KNN)

The KNN algorithm uses the assumption that similar things exist close in proximity (Harrison 2018). The abbreviated process for the KNN algorithm using the equation

$$d = nk = x1 \times 1k - x2 \ \times k2$$

Figure 3.2 – KNN Algorithm

Initialise data for processing.

Select a value of k, the chosen number of neighbouring points in this case 10.

Calculate the distance between the current point and the queried point.

The distance and the index of the point to an array, sorted in distance from nearest to furthest.

Use the value of the K entry to return the mean or the mode dependent on application.

The value of K is dependent on several variables, the value wants to keep the number of errors in calculation to a minimum while being able to make reliable predictions about unknown data. Decreasing the value of K produces fewer stable predictions, while increasing to threshold improves predictions. However, once the threshold is reached the number of errors will begin to increase. The speed at which the algorithm is executed slows as the number points, predictors, or independent variables increases (Varghese 2019).

Advantages

i. Simple to implement.
ii. Small number of parameters to optimise.
iii. Produces results in real time.

Disadvantages

i. Value of K requires careful selection.
ii. Requires large amounts of computing resources and increases exponentially with dataset size.
iii. Longer runtimes than other algorithms

3.4 Decision Tree

Uses a tree-based structure to solve regression and classification problems (Varghese 2019). Derived from the independent variables, each node having a condition to produce a feature. Where each node decides on how to navigate based on a given condition. Once the final stage is reached, called a leaf node, an output is predicted. Making the tree efficient requires selecting the correct conditions, while information gain is used to select node conditions. With tree structure is developed using a recursive, greedy algorithm (Varghese 2019).

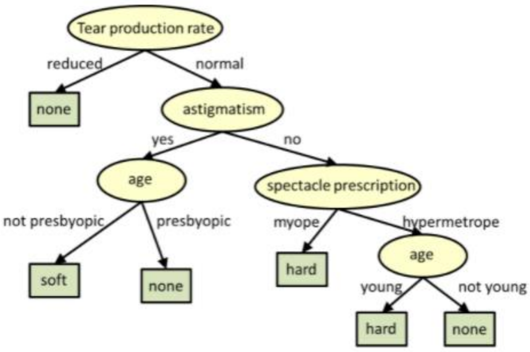


*Figure 3.2 Decision Tree Structure with nodes and leaves (*Decision Trees *2020)*

Gini index is used for classification and regression trees, used for calculating how mixed data points are (Varghese 2019).

$$Gini\ Index = 1 - P2t$$

Figure 3.3 – Gini Index

Entropy and information gain for selecting the next attribute.

$$Hs = \ -PC * \log^{f(s)}(PC)$$

Figure 3.4 - Entropy

$$Igs = hs - tPt * H(t)$$

Figure 3.5 – Information Gain

H(s) is entropy and IG(s) is information gain used for calculating entropy difference of parent and child nodes.

Advantages

1. No data pre-processing.
2. Data distribution requires no assumptions.

3. Efficient with collinearity

Disadvantages

1. Overfitting the model without decision tree pruning.
2. Prone to outliers
3. May become overly complex with large datasets.
4. Losses critical information with continuous variables.

3.5 Naïve Bayes

Naive Bayes assumes that there is no dependence amongst attributes, referred to as class conditional independence (Ashari, Paryudi & Min 2013). It is used to solve classification problems throughout many industries and is based on the Bayes theorem, an extension of conditional probability which uses one conditional probability to calculate another. The equation representing Bayes Theorem.

$$PB = PA * \frac{P(A)}{P(B)}$$

Figure 3.6 – Bayes Theorem

The probability of A given that B has occurred, the right-hand side the probability of B given A has occurred, multiplied by the probability of event A to event B (Ashari, Paryudi & Min 2013).

Advantages

1. Performs better than other algorithms if the assumption of independent predictors is true.
2. Straightforward to implement.
3. Requires a small amount of training data to estimate test data (Ashari, Paryudi & Min 2013).

Disadvantages

1. The assumption of independent predictors is the main limiting factor of Naïve Bayes, it assumes that all attributes are independent. In the physical world the likelihood of a set of independent predictors is extremely low.
2. A categorical variable in the test data sheet not in the training data set will result in a prediction being unable to be made. This is called zero frequency and can be smoothed using techniques such as Laplace estimation.

3.6 Golden Section Search

The golden section search is used for finding the maximum or minimum of a dataset inside an interval specified by the user, usually for a unimodal function, one that contains only one maximum or minimum on the interval. It can be adopted as a line search method for univariate problems by progressively subdividing the regions until the difference between these is less than the specified tolerance. Once this has been accomplished the maximum or minimum is found (Hanson, 2021). The conditions that need to be satisfied so that the correct points between the bounds are examined are:

1. The distance between the optimised bounds is the same whichever side they are removed from.
2. This distance or proportion is to remain constant on every iteration, using one of the existing calculated points on the next iteration.

Referring to *figure 3.7*, initially three points are assigned, xi, $x1$ and *xu* where ($xi<x1<xu$). The values of the function correspond to $f(xi), f(x1)$ *and* $f(xu)$ and the maximum value lies between these points. A fourth point $x2$ is used between the larger intervals of [xi, $x1$] and [xi, xu], if the larger interval is $f(x2)>f(x1)$ then the new coordinates will be $xl<x2<x1$. This process continues until the distance of the outer point reaches the specified tolerance (Yalcin and Kaw, 2021).

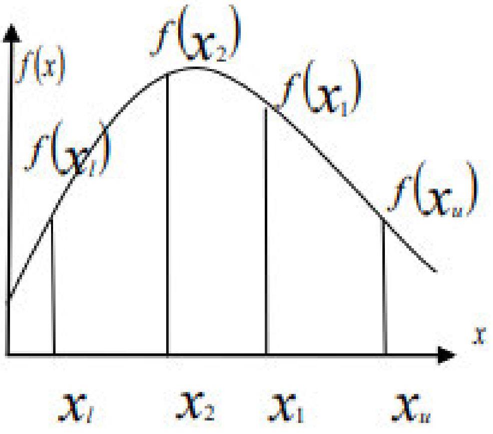


Figure 3.7 - Cross section of a function (Yalcin and Kaw, 2021)

Determining the distance of the intermediate points for the Golden Section Search, is based on the Golden Ratio. This is a ratio that has been used in mathematics and engineering since 300 BC and is the ratio of two values whose sum of the larger of the two quantities is the same as their ratio. Mathematically it is expressed as a+ba = ab where a is greater than b and both are greater than zero. Alternately let R = ab then the equation can be written as:

$$R2 + R - 1 = 0$$

$$Then\ use\ the\ quadratic\ fromula\ to\ solve$$

$$R = \ -1 + 1 - 4(-1)2$$

$$R\ = 0.61803$$

Figure 3.8 – Deriving the Golden Ratio

The distance of the intermediate points to the boundaries of the search region is as close to the golden ratio as practical as demonstrated in *figure 4.7*, where *xl and x2* are chosen to ensure the distance from the boundaries represents this ratio.

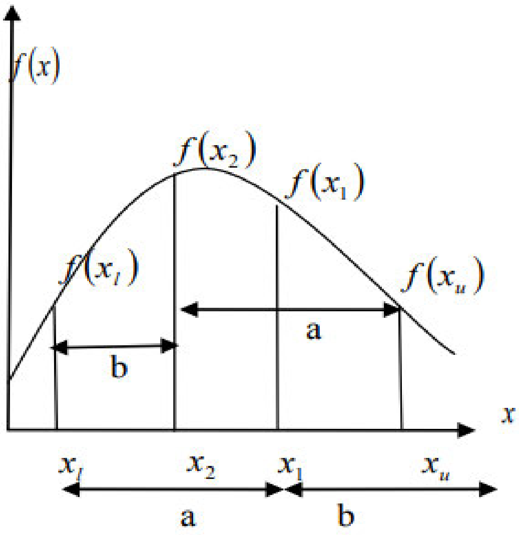


Figure 3.8 - Cross section of a function (Yalcin and Kaw, 2021)

After the first two intermediate points are determined, a smaller interval must be chosen where the maximum or minimum of the function exists. This interval is located at either [*xl*, *x2, x1*] or [*x2*, *x1, xu*], which of these is used will be found by evaluating the function at points *x2 and x1*. In the case that *f(x2)>f(x1)* the next interval will be [*xl*, *x2, x1*], else this not been the case then the next interval will be [*x2*, *x1, xu*]. In the case of *figure 4.7, indeed f(x2)>f(x1)* giving new interval at [*xl*, *x2, x1*]. The boundary of the smaller interval is *xl* and *x1* with one of the intermediate points already located at *x2*, which has a ratio to the boundary of the golden ratio and the second interval at *xl* (Hanson, 2021).

3.7 An Unknown Algorithm

Most algorithms evaluate a function at several points, with the location of the test points based on the value of the function at the previous test, with many algorithms halting on local maxima or minima. Majority of algorithms that evaluate functions this way can be very time consuming and require large amounts of processing power. With optimisation algorithms it is essential that the maximum or minimum be found using the minimum amount of test points. If the known, then the search can conclude once a test point returns the maximum value. The following algorithm is based on a method developed by H.J Kushner's versatile stochastic model function, whose form is unknown and time varying. If the global maximum or upper boundary of a univariate function is known, the method results in a simpler algorithm that does not regard the differentiability of the function (Haskell, Castelino and Mirshab, 2021).

Figure 3.10 represents a univariate function g(z), each interaction is performed on the interval $[\mathrm{Zmin}, \mathrm{Zmax}]$ and it is assumed that the maximum value of the function is G, with the other parameters defined as.

$$dmax = G - g(Zmax)$$

$$dmin = G - g(Zmin)$$

Figure 3.9 – dmax and dmin values

Each iteration of the algorithm at a point within the interval is selected as a new possibility for the maximum represented as ẑ.

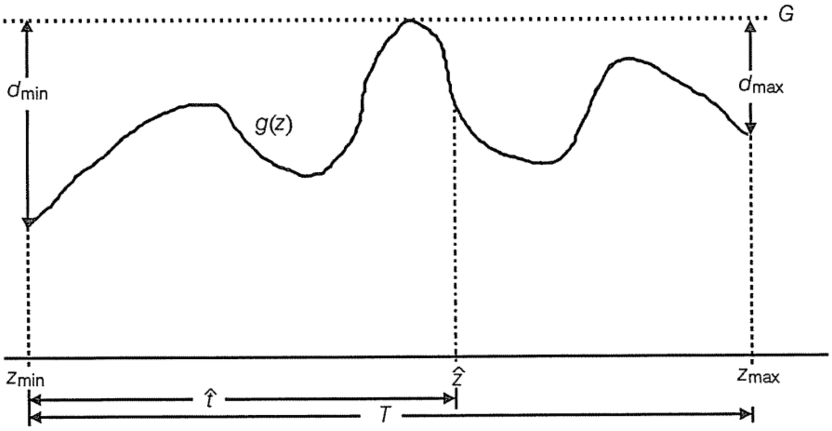


Figure 3.10 - An Arbitrary Function (Haskell, Castelino and Mirshab, 2021).

The algorithm iterates through a series of steps using the initial test interval [Zmin, Zmax], and can be described as:

1. $\mathrm{If\ g(zmin) = G\ or\ g(zmax) = G\ then\ END}$

2. $\mathrm{Do\ the\ following\ n\ times\ or\ until\ the\ maximum\ has\ been\ found}$

3. $\mathrm{Compute}\ \hat{z}$

$\hat{Z} = Zmin + t$

Where, $t = dminT\ (dmin + dmax)$

And $T = Zmax - Zmin$

4. If $g(\hat{z}) = G$ the END
5. Reduce interval into smaller intervals - $[Zmin, \hat{z}]$ and $[\hat{z}, Zmax]$
6. $\hat{Z}$ becomes new Zmax for the first interval and the new Zmin for the second
7. Calculate Amin

   $Amin = (dmin\ dmax)/T$

8. Each interval is added to a segment list containing Amin, Zmin, Zmax, dmin and dmax.
9. The entry with the smallest Amin in the segment list, the new interval corresponding

   [Zmin,Zmax]

The value of n can be determined based on some arbitrary value or as a percentage of the total search space, if no maximum is found after the value of n is reached the result is the closest to the value of the maximum (Haskell, Castelino and Mirshab, 2021).

3.8 Equal Interval Search Method

The equal interval search method is one of the simplest methods for finding a local maximum or minimum of a univariate function. To find the maximum of a function, f(x), where the maximum exists in the interval [a, b]. Referring to figure 4.8, using an interval of an arbitrary value e at which it is assumed the maximum occurs, two intervals can be calculated.

$$f\left(\frac{(a+b)}{2}+\frac{e}{2}\right) and\ f\left(\frac{(a+b)}{2}-\frac{e}{2}\right)$$

$$if\ f\left(\frac{(a+b)}{2}+\frac{e}{2}\right) \geq \left(\frac{(a+b)}{2}-\frac{e}{2}\right)$$

$$Then\ the\ interval\ where\ the\ maximum\ occurs\ is$$

$$[\frac{(a+b)}{2}-\frac{e}{2}, b]$$

$$else\ maximum\ occurs$$

$$[\frac{(a+b)}{2}+\frac{e}{2}, a]$$

Figure 3.11 – Where the Maximum Occurs Equal Interval Method

The process is repeated until the interval is reduced to a predetermined size (Yalcin and Kaw, 2021). From the data points A, M-, M+, and B, the value of the function increases from A to M-. then decreases from M- to M+. When there is a sudden change in function value as this, the new upper and lower bracket values can be evaluated. The lower bound remains at Point A, with the new upper boundary at M+.

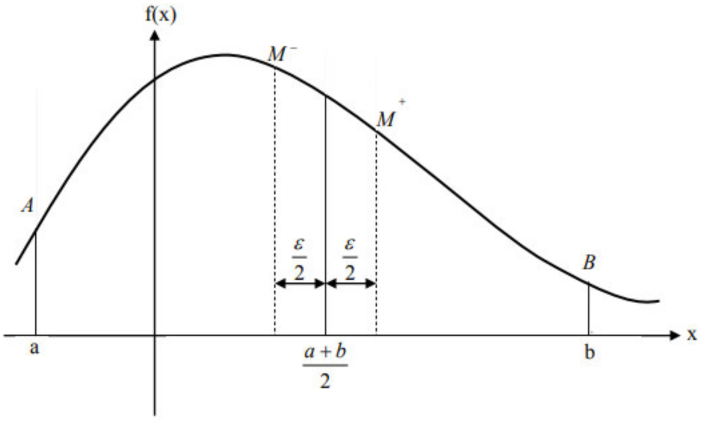


Figure 3.12 - Cross section of a function (Yalcin and Kaw, 2021)

3.9 Chapter Summary

While six different algorithms with methods of optimising functions were examined only three will be implemented and evaluated for optimising the data from the global search. These will be the Golden Section Search, and Unknown Algorithm and the Equal Section Search. The K-nearest Neighbour, Naive Bayes and the Decision Tree are complex algorithms and after evaluation would require many resources to adapt and compile for the purpose of locating the maximum of a function.

The equal interval search is a very basic and potentially inefficient algorithm for evaluating the maximum of a function. As the algorithm starts in the midpoint of the function, then takes values either side of the midpoint at a pre-determined distance. Then comparing these values follows the direction in which the function is increasing. Even if the function has a single a maximum then the function should arrive at a maximum, if the function has multiple maximums or multiple values close to the maximum, the algorithm may struggle to locate the true maximum.

An unknown algorithm overcomes the issue of multiple maxima, as it references a predicted maximum value. While this ensures faster convergence it is another step in the algorithm, as the value must be calculated before the algorithm can locate the maximum.

The golden section search is also intended to find a single maxima or minima of a function. The limitations apply as with the previous algorithms, as there may be multiple maxima or values close to the maximum which could cause errors in processing the raw data. The main advantage of the golden section is no pre-calculating of maxima is required or size of interval.

Methodology

This chapter outlines how each algorithm will process the raw data in MATLAB to determine the maximum blood flow velocity in the local search results. The result of this will be analysed for the number of steps taken to reach the outcome, which will be the metric for measuring the performance of the algorithm.

4.1 MATLAB Introduction

MATLAB will be used to simulate the algorithms as performing the calculations manually would be very time consuming and inefficient use of time. MATLAB allows a high number of calculations of every second, whilst offering extensive support for math-based applications and is used extensively in academic institutions.

4.2 Raw Data Evaluation

Raw data from the global search has been provided by the University of Southern Queensland. It is presented in a three-dimensional plane, refer to *figure 4.9.1*, and there are three different data points to make up the complete data set:

I. X coordinate - Deflection angle of the probe located at the horizontal plane
II. Y coordinate - Deflection angle of the probe located at the vertical plane
III. Z coordinate - Strength of the signal from the doppler ultrasound

While the algorithms presented are implemented to examine a single plane, X and Y coordinates, where the X axis increases evenly, and the Y value is potentially varying.

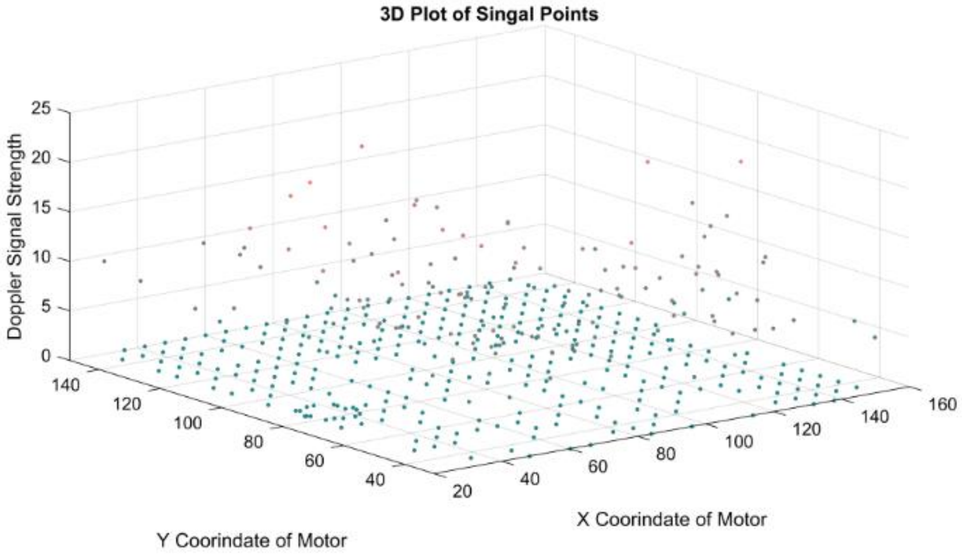


Figure 4.1 - 3D Plot of Raw Global Search Data (USQ 2021)

Referring to *Figure 4.2*, the first plot is the position of the X probe as it sweeps at various angles through a set range, the angle of the X probe is varying constantly. We can see from the second plot that the Y axis probe goes through step changes approximately every 21 iterations of the X probe. Finally, the third plot is the result of the doppler signal at both points X and Y and is the strength of the doppler signal.

The focus of the optimisation will be on the Z doppler strength signal. Other data is predetermined an Z is essentially the unknown. Algorithms will examine each data set of 21 values of the Z value and attempt to locate the maximum value in the least number of iterations of loops as possible. Even one or two iterations less over a large dataset can result in much quicker processing of the raw data.

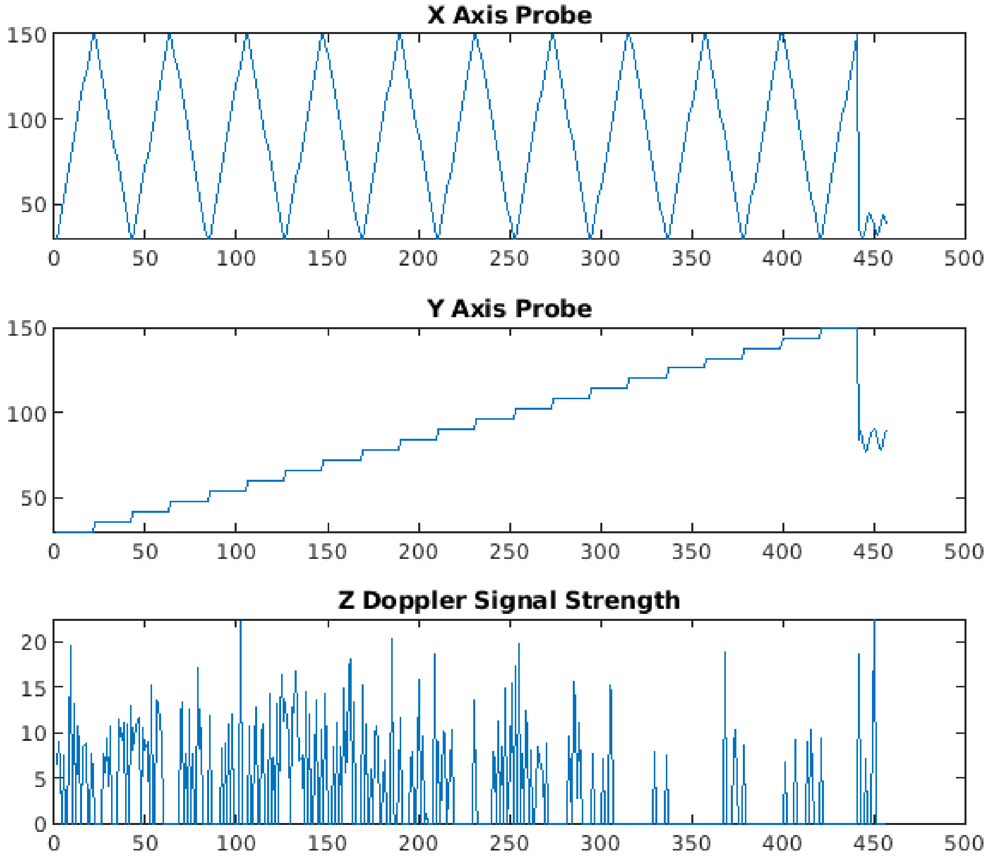


Figure 4.2 - Separate Plots of Global Search data

4.3 Data Integration

As the data from the global search is presented in three dimensions, with the X, Y and Z planes, the local maximum will be evaluated within two planes, X and Y only. This will be achieved by breaking the global data into smaller sections and optimising these smaller sections individually. In the preceding chapter it was identified that The Y axis probe remained static through twenty iterations of the X probe. Based on this, the X value can represent the X plane, as for twenty-one iterations it is linear. The Z values can be represented on the Y plane as their value is random and the maximum is unknown. Referring to Figure 5.2.1, twenty-one X probe values are plotted, on the Y axis twenty-one Z plane data points. The selected algorithms will be applied to these smaller blocks of data points to find the maximum point. There is a total of four hundred and fifty-seven data points in the global search data set, however the last sixteen data points appear corrupted and will be omitted from the data processing. This brings the total amount of data blocks to twenty-one, the selected algorithms will be applied to the separate blocks which represent different planes in the ultrasound image.

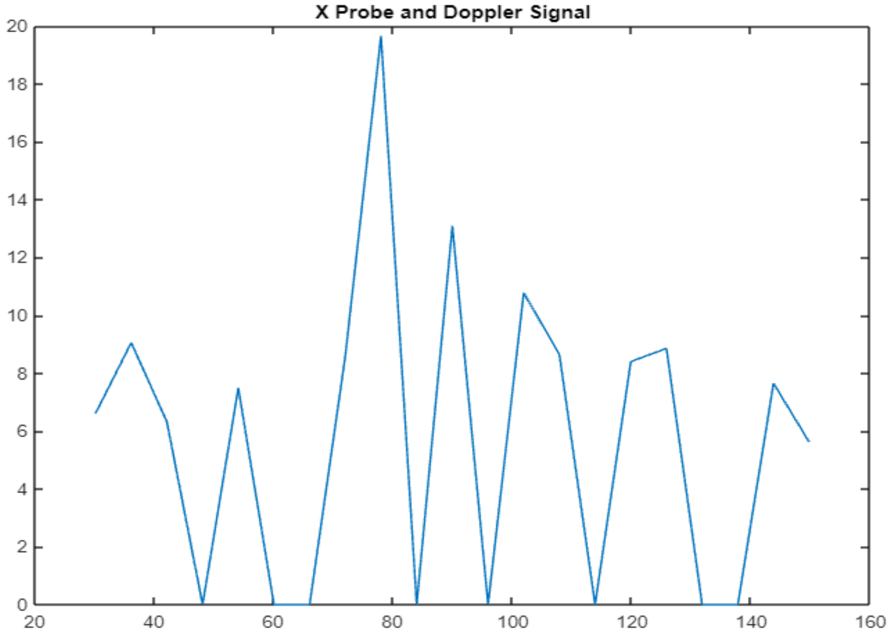


Figure 4.3 – Global Search Data with Y Plane Removed

With the global data separated into separate planes, with the maximum in each plane calculated, the data can be transferred to a new array or matrix with the maximum value and the X and Y coordinates of the maximum point.

4.4 The Golden Section Search MATLAB Implementation

The Golden Section Search is normally used to evaluate a function with a known maxima or minima. In the case of the global search data, there is no known maximum for the data, and as it appears, is not represented by a known function. Following will be the steps of using MATLAB to implement the Golden Section Search for the global data.

Firstly, as established the function will be replaced with one of the data planes, with the upper and lower bounds as xU and xL. The golden ratio can also be declared in relation to the upper and lower boundaries denominated by D, referring to *Figure 3.8*:

$$R2 + R - 1 = 0$$

$$Then\ use\ the\ quadratic\ fromula\ to\ solve$$

$$R = -1 + 1 - 4(-1)2$$

$$R = 0.61803$$

Figure 3.8 – Deriving the Golden Ratio

Results in

$$D = R(xu - xL)$$

Figure 4.4 – Golden Ratio in Relation to Upper and Lower Boundaries

Using pseudo code this will be represented as:

F = single global data plane

Xu = the upper bound of the data set, in this case the lowest deflection angle of the X probe

Xl = the lower bound of the data set, in this case the highest deflection angle of the Y probe

Fu = value of the data at Xu

Fl = value of the data at Xl

D = the ratio of the upper boundary less the lower boundary in the golden ratio

With the boundary ratio and the upper and lower limits determined, then the values of the global data at the initial boundary ratios can be calculated:

X1 = the upper limit Xu, minus the golden ratio of the boundaries D

X2 = the lower limit XL, plus the golden ratio of the boundaries D

F1 = the value of the global data at the point X1

F2 = the value of the global data at the point X2

With the ratio and the values along the X axis of the data plane may differ, they will be rounded to match the available data points, or the resolution of the x axis may be increased to meet the requirements. The next stage is to establish conditions for the algorithms on how to proceed through the search process to find the maximum value.

If F1 > F2

Then the upper boundary moves to X2

Xu = X2

Fu = X1

New value for X2 assigned

X2 = X1

F2 = F1

New value for X1 assigned

D = R*(Xu-Xl) re-evaluates distance between the bounds

X1 = XU - D

Evaluate F1

F1 = value of data plane at point X1

Otherwise, if the value of F1<F2 the lower bound moves to X1

Xl=X1

Fl = F1

New value assigned to X1

X1 = X2

Fl = F2

New value assigned to X2

D = R*(xU-xL) re-evaluates distance between the bounds

X2 = Xl + D

Evaluate F2

F2 = value of data plane at point X2

There is a possibility that F1 and F2 may be equal, in the case that they both contain a maxima or minima, in this case they are both the result

Xl = (X1 + X2)/2 midpoint of X1 and X2

Xu = Xl this will return an error for the result

Then convergence can be confirmed

Error = the absolute value of the span of the upper and lower bound relative to the centre position of the lower boundary, where:

Error = absolute value of Xu =Xl

With the variables and values established the program will have to iterate through the search function until it can find the maximum value, achieved by using a loop with set conditions to terminate the loop once a value in the tolerance is found. The loop can be constructed as follows:

Tol = tolerance of the loop relative to the error, while the error is greater than the tolerance the loop will continue, upon the tolerance exceeding the error, then the result is evaluated.

Xe = Result where (X1 + X2)/2 averaging the values of X1 and X2.

At this point the values must be referenced back to the Y probe deflection angle, which is constant through the selected range, the X probe deflection angle is relative to the result as is the maximum value.

4.5 Implementing an Unknown Algorithm in MATLAB

The Unknown Algorithm, like the Golden Section Search, requires a known local maximum or minimum, although not a known function. The value of G normally represents the maximum, in this case it will represent the estimated upper boundary. An arbitrary univariate function would normally be represented by g(z), in this application let g(z) represent the global data. Each iteration of the variable is performed on the boundaries Zmin and Zmax (Haskell, Castelino and Mirshab, 2021), dmax and dmin is the value of the global data evaluated at points Zmin and Zmax.

The algorithm will use the values of Zmin and Zmax as initial test intervals and iterate through a loop until the maximum value of iterations is reached. The pseudo code for this algorithm in MATLAB is as follows.

Declares the value G, equal to or greater than the maximum value of the global search data, at this point this will be formed for small datasets manually, for larger data sets there are functions available in MATLAB for finding a maximum, this will affect the performance of the algorithm.

If g(Zmin) = G or g(Zmax) = G, then the maximum is located at these points

Value for variable n, can be set manually or set as a percentage of the number of data points in the global data, as it is assumed that the maximum must have been found by this point.

Calculate dmin and dmax, these are the values of Z of the global search data, evaluated at the minimum and maximum of the X axis.

dmin = G - g(Zmax)

dmax = G - g(Zmin)

Calculate the value T, ṱ, and ẑ.

T produces the span of the interval

T = Zmax - Zmin

ṱ calculates a new smaller interval proportional to the total span

ṱ = dmin T / (dmin + dmax)

Creates the new upper boundary for the new interval, also minimum for the next interval

ẑ = Zmin + ṱ

At this point if g(ẑ) = G the maximum is found, this is the value of the function evaluated at the new upper boundary, also the next intervals lower boundary

Divide the interval into two smaller intervals

[Zmin, ẑ] and [Zmax, ẑ] where ẑ becomes the Zmax for the first interval and Zmin for the second interval.

Calculate a new parameter, Amin, for both new intervals, this evaluates how close the values at dmax and dmin are to the maximum value, the smaller this value the closer to the maximum both points are.

Amin1 = (dmin*dmax)/T

The new segment where Amin is the smallest, becomes the new test interval with the corresponding [zmin, zmax].

The number iteration can either be predetermined, this will start at a higher number during testing and reduced until a result is produced before error.

4.6 Equal Interval Search MATLAB Implementation

As this is the simplest method of finding the local maximum or minimum of a known function it will be used as a baseline for the other methods for comparison of optimisation. The size of the interval is predetermined and is represented by ε and will use this same size interval throughout the entire evaluation of the global search data.

Firstly the range evaluated will be represented by [a,b], where a is the lower boundary of the global search data and b is the upper boundary.

a = lower boundary x value of global search data

b = upper boundary x value of global search data

Let ε = 1 for the initial interval size, this will be adjusted through subsequent testing for optimal number of steps.

The first interval

F1 = Z((a + b) / 2 + ε/2) lower interval evaluates the value of Z plane from the data at this point

F2 = Z((a + b) / 2 - ε/2) upper interval evaluates the value of Z plane from the data at this point

If F1 > F2 then the maximum lies in this interval

Alternatively

If F1 < F2 then the maximum lies in this interval

The interval will then be moved to coincide with the side that the maximum was found

If F1 > F2

Create a new interval where the upper boundary is b and the lower boundary is F2

a = F2

F2 = F1

F1 = Z((a + b) / 2 + ε/2)

If F1 < F2

Create a new interval where the upper boundary is F1, and the lower boundary is a

b = F1

F1 = F2

F1 = Z((a + b) / 2 - ε/2)

Iterate through this until the specified number of loops is reached, initially this will be five, with further testing this number may be adjusted with regards to error rate of the algorithm.

Results and Discussion

The following chapter provides how the results from test data have been analysed and evaluated. Using MATLAB to simulate algorithms and evaluate effectiveness of each by taking into consideration the number of steps of code, the number of iterations the code requires to reach the maximum and the error rate of the function. The number of steps can be considered secondary as the number of iterations required as this will act a multiplier each time the iteration increases. The erratic nature of the raw data has also been taken into consideration as the multiple number of minima and maxima can make it difficult for algorithms.

5.1 Results Summary

Processed data has been tabulated after algorithm optimisation. The first five data sets (each of 21 data points) have been recorded. The first column is the range of the data set each of twenty-one points, the second is the number of times the algorithm has processed the data referred to as iterations. The third and fourth columns indicated the location and maximum value of that dataset and the fifth and sixth are the results returned from the algorithm. The final two columns are the difference between the actual maximum value and the algorithms calculated values.

The second set of data tables is the same process with data smoothing applied in MATLAB *(Figure 5.1)*, this eliminates the outliers and possibly reduces the value of maxima and minima values proportionally. This was implemented to assist with the algorithms locating the maxima more efficiently, removing multiple maximum and minimum values so the search could follow the increase in function value easier.

Further to this the averages of the position error and function value error are presented in a table to give some comparison between the algorithms results of the smoothed and raw data.

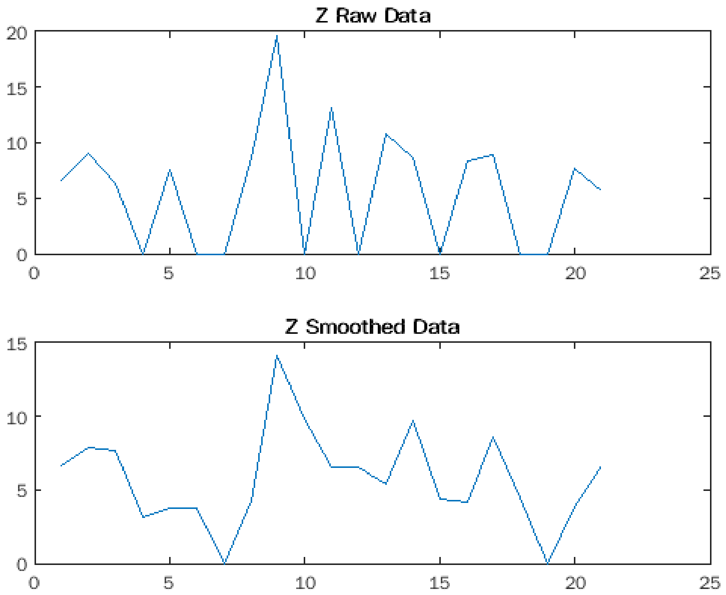


Figure 5.1 Comparison of Raw Data and Smoothed Data

5.2 Equal Interval Method

Initial finding of the equal section search is that's the error rate of the raw data is high, with loop count varying from one to five which it was found that there was little effect on the result after four iterations, in some cases the error rate increased with larger number of iterations moving further away from the maximum, also the zero values have been offset by one as zero returns a false result.

This algorithm had the highest error rate for position error, the change in slope can cause the algorithm to return false results and become stuck on a certain value, even though this is not the maximum. While the average function value error is not as high as the other algorithms, this is less significant than producing an incorrect location.

The number of steps required for the algorithm to complete one iteration is twenty-nine. Based on five iterations this potentially evaluates the maximum at one hundred and forty-five steps. With a high error rate and an inaccurate evaluation of the maximum location this method in this form cannot reliably find a local maximum.

Table 5.1 Averaged Result Data Equal Interval Algorithm

Dataset	Maximum Value	Maximum Location for X	Average Function Value Error	Average Position Error (Raw Data)	Average Position Error (Smoothed Data)
1:21	19.655	11	10.16164	1.2	3.8
22:42	12.897	21	6.85276	14.2	12
43:63	15.218	11	2.20308	3.6	3.8
64:84	17.059	16	4.6496	9.2	8
85:105	22.194	18	13.3605	8	8
Total Averaged	~	~	7.445516	7.24	7.12

5.3 An Unknown Algorithm

Error rate for the Unknown Algorithm is 11.36421 for the value and 4.52 for the X position (*Table 5.2). This is a minimal improvement over the Equal Interval Search with one of the five data sets been calculated accurately. Examining the full results *(Table B.3),* the algorithm cannot move past a local maximum, as it returns the same location for the maximum location in each of the datasets with exception to one, the 43:63 range in which it only changed once. The final three data sets returned a zero result, which is the minimum, although one of multiple points of this value through the dataset. Main source of error could be that the algorithm fails to move off the local maximum as the search iterates.

This is evaluated in the literature review as this algorithm was designed to avoid this result. Examining the algorithm, the error appears to occur when a zero value is returned for the variable Zd, this occurs if the initial maximum is at the Zmin or Zmax point. Placing an offset of one for G provides some improvement, although the algorithm still fails to locate the maximum on majority of iterations.

Lines of code required for the algorithm are thirty-four and based on number of iterations to reach a maximum for the full results is four. Resulting in one hundred and thirty-four steps to reach a maximum.

Table 5.2 Averaged Result Data an Unknown Algorithm

Dataset	Maximum Value	Maximum Location for X	Average Function Value Error	Average Position Error (Raw Data)	Average Position Error (Smoothed Data)
1:21	19.655	11	6.5716	2	12
22:42	12.897	21	0	0	0
43:63	15.218	11	10.99564	4.6	7.2
64:84	17.059	16	17.0594	5	5
85:105	22.194	18	22.1944	11	3
Total Averaged	~	~	11.36421	4.52	5.44

5.4 The Golden Section Search

The Golden Section Search returned the lowest error rate of all three algorithms and found a maximum in the least number of loop iterations normally converging after three iterations as can be found in *Table B.5*. As with the other algorithms with the erratic nature of the raw data it can become stuck on a false maximum, given some improvements with data smoothing the error rate was reduced.

The algorithm is the least dependent on prior information in that it forms all its intervals of the basis of the Golden ratio applied at the upper and lower bounds of the function. While the value function error was smaller, the smoothed data position error proved the strongest at 3.96. The number of steps for each iteration was forty-two, resulting in one hundred and twenty-six total steps. This gives the Golden Section Search the least number of steps for all algorithms.

Although the algorithm remained stuck on false maximums potentially requiring more work to locate the true maximum.

Table 5.3 Averaged Data for the Golden Section Search

Dataset	Maximum Value	Maximum Location for X	Average Function Value Error	Average Position Error (Raw Data)	Average Position Error (Smoothed Data)
1:21	19.655	11	2.2118	02	0
22:42	12.897	21	4.88482	12.6	8.8
43:63	15.218	11	5.90478	2.4	4.4
64:84	17.059	16	3.51556	0.4	1.6
85:105	22.194	18	22.1944	7	5
Total Averaged	~	~	7.74227	4.12	3.96

5.5 Chapter Summary

While intermittently all three algorithms managed to locate the maximum, on most occasions all three became stuck on a false maximum. The nature of the algorithms is that it would narrow the focus onto an upward direction of a function, if the function had a higher maximum after a down trend than often this was missed. Although unsuccessful the highest performing was the Golden Section Search, in which with more refining could possibly produce more accurate results.

Conclusions and Further Works

6.1 Conclusion

The primary objective was to produce a time optimal controller for a transcranial ultrasound, by developing an algorithm that would locate the maximum blood flow in the least number of steps. While the algorithms did not produce accurate results for the maximum value of a function, there is potential with improvements there could be an increase in accuracy. It was found that all of the researched algorithms are capable of locating the maximum of a univariate function, however with the raw data provided there are several values at near maximum that resulted in high error rates.

6.2 Further Works

Further work to either smooth the raw data, or variations of the algorithms to help to locate the maximum. It may also be prudent to extrapolate the raw data, although this may increase the processing time with no benefit.

Appendix A

Project Specification

For: Kieran Higo

Title: The time-optimal controller design & implementation for an auto TCD probe

Major: Electrical and Electronic Engineering

Supervisors: Paul Wen and Bo Song

Enrolment: ENG4111 – EXT S1, 2021

ENG4112 – EXT S2, 2021

Project Aim: Design and implement a controller to increase speed and efficiency of the TCD probe reducing discomfort to the user.

Program: Version 1, 15th March 2021

Initiate background research on Transcranial doppler ultrasound (TCD), including past, current, and conceptual designs.

Create a model from current servo unit used in TCD.

Simulate controller model using MATLAB capturing relevant data for benchmarking.

Assess data for optimization areas.

Prioritize optimization points select areas that will give greatest performance increase.

Create optimised controller model for simulation.

Evaluate optimised data based on the model.

Implement new controller in existing servo unit.

Simulate TCD unit with optimised controller.

Process and evaluate experimental data.

Evaluate the effectiveness of new controller.

Provide recommendations for any further improvements.

If time and resource permit:

Collect feedback from field users and service reps for any improvements recommended.

Appendix B

Algorithm Test Results

The following appendix will provide full test results of the Equal Interval Search, Unknown Algorithm, and the Golden Section Search Algorithm.

Table B.1 - Equal Interval Search Results

Dataset Range of X	Iterations	Max value X	F(x)	Calculated Value	F(x)	Position Error	Error Value
1:21	1	X = 9	19.6554	X = 11	13.0832	2	6.5722
1:21	2	X = 9	19.6554	X = 8	8.5964	1	11.059
1:21	3	X = 9	19.6554	X = 8	8.5964	1	11.059
1:21	4	X = 9	19.6554	X = 8	8.5964	1	11.059
1:21	5	X = 9	19.6554	X = 8	8.5964	1	11.059
22:43	1	X = 21	12.8973	X = 0	0	21	12.8973
22:42	2	X = 21	12.8973	X = 14	11.4414	7	1.4559
22:42	3	X = 21	12.8973	X = 7	5.5456	14	7.3517
22:42	4	X = 21	12.8973	X = 7	5.5456	14	7.3517
22:42	5	X = 21	12.8973	X = 6	7.6901	15	5.2072
43:63	1	X = 11	15.2176	X = 11	15.2176	0	0
43:63	2	X = 11	15.2176	X = 5	11.5458	6	3.6718
43:63	3	X = 11	15.2176	X = 5	11.5458	6	3.6718
43:63	4	X = 11	15.2176	X = 5	11.5458	6	3.6718
43:63	5	X = 11	15.2176	X = 11	15.2176	0	0
64:84	1	X = 16	17.0594	X = 11	12.5738	5	4.4856
64:84	2	X = 16	17.0594	X = 7	13.2313	11	3.8281
64:84	3	X = 16	17.0594	X = 6	12.0813	10	4.9781
64:84	4	X = 16	17.0594	X = 6	12.0813	10	4.9781
64:84	5	X = 16	17.0594	X = 6	12.0813	10	4.9781
85:105	1	X = 18	22.1944	X = 0	0	18	22.1944
85:105	2	X = 18	22.1944	X = 14	11.9534	4	10.241
85:105	3	X = 18	22.1944	X = 14	11.9534	4	10.241
85:105	4	X = 18	22.1944	X = 14	11.9534	4	10.241
85:105	5	X = 18	22.1944	X = 8	8.3093	10	13.8851

Table B.2 Equal Section Search Smoothed Data

Dataset Range of X	Iterations	Max value X	F(x)	Calculated Value	Position Error
1:21	1	X = 9	19.6554	X = 11	2
1:21	2	X = 9	19.6554	X = 6	3
1:21	3	X = 9	19.6554	X = 5	4
1:21	4	X = 9	19.6554	X = 2	7
1:21	5	X = 9	19.6554	X = 6	3
22:43	1	X = 21	12.8973	X = 11	10
22:42	2	X = 21	12.8973	X = 11	10
22:42	3	X = 21	12.8973	X = 11	10
22:42	4	X = 21	12.8973	X = 6	15
22:42	5	X = 21	12.8973	X = 6	15
43:63	1	X = 11	15.2176	X = 11	0
43:63	2	X = 11	15.2176	X = 3	8
43:63	3	X = 11	15.2176	X = 5	6
43:63	4	X = 11	15.2176	X = 8	3
43:63	5	X = 11	15.2176	X = 9	2
64:84	1	X = 16	17.0594	X = 11	5
64:84	2	X = 16	17.0594	X = 11	5
64:84	3	X = 16	17.0594	X = 6	10
64:84	4	X = 16	17.0594	X = 6	10
64:84	5	X = 16	17.0594	X = 6	10
85:105	1	X = 18	22.1944	X = 0	18
85:105	2	X = 18	22.1944	X = 14	4
85:105	3	X = 18	22.1944	X = 14	4
85:105	4	X = 18	22.1944	X = 14	4
85:105	5	X = 18	22.1944	X = 8	10

Table B.2 - An Unknown Algorithm Results

Dataset Range of X	Iterations	Max Value X	F(x)	Calculated Value X	Calculated Value F(x)	Position Error	Error Value
1:21	1	X = 9	19.6554	X = 11	13.0838	6.5716	2
1:21	2	X = 9	19.6554	X = 11	13.0838	6.5716	2
1:21	3	X = 9	19.6554	X = 11	13.0838	6.5716	2
1:21	4	X = 9	19.6554	X = 11	13.0838	6.5716	2
1:21	5	X = 9	19.6554	X = 11	13.0838	6.5716	2
22:42	1	X = 21	12.8973	X = 21	12.8973	0	0
22:42	2	X = 21	12.8973	X = 21	12.8973	0	0
22:42	3	X = 21	12.8973	X = 21	12.8973	0	0
22:42	4	X = 21	12.8973	X = 21	12.8973	0	0
22:42	5	X = 21	12.8973	X = 21	12.8973	0	0
43:63	1	X = 11	15.2176	X = 7	10.5549	4.6627	4
43:63	2	X = 11	15.2176	X = 7	10.5549	4.6627	4
43:63	3	X = 11	15.2176	X = 6	0	15.2176	5
43:63	4	X = 11	15.2176	X = 6	0	15.2176	5
43:63	5	X = 11	15.2176	X = 6	0	15.2176	5
64:84	1	X = 16	17.0594	X = 21	0	17.0594	5
64:84	2	X = 16	17.0594	X = 21	0	17.0594	5
64:84	3	X = 16	17.0594	X = 21	0	17.0594	5
64:84	4	X = 16	17.0594	X = 21	0	17.0594	5
64:84	5	X = 16	17.0594	X = 21	0	17.0594	5
85:105	1	X = 18	22.1944	X = 7	0	22.1944	11
85:105	2	X = 18	22.1944	X = 7	0	22.1944	11
85:105	3	X = 18	22.1944	X = 7	0	22.1944	11
85:105	4	X = 18	22.1944	X = 7	0	22.1944	11
85:105	5	X = 18	22.1944	X = 7	0	22.1944	11

Table B.4 - An Unknown Algorithm Results Smoothed Data

Dataset Range of X	Iterations	Max Value X	F(x)	Calculated Value X	Position Error
1:21	1	X = 9	19.6554	21	12
1:21	2	X = 9	19.6554	21	12
1:21	3	X = 9	19.6554	21	12
1:21	4	X = 9	19.6554	21	12
1:21	5	X = 9	19.6554	21	12
22:42	1	X = 21	12.8973	21	0
22:42	2	X = 21	12.8973	21	0
22:42	3	X = 21	12.8973	21	0
22:42	4	X = 21	12.8973	21	0
22:42	5	X = 21	12.8973	21	0
43:63	1	X = 11	15.2176	7	4
43:63	2	X = 11	15.2176	3	8
43:63	3	X = 11	15.2176	3	8
43:63	4	X = 11	15.2176	3	8
43:63	5	X = 11	15.2176	3	8
64:84	1	X = 16	17.0594	21	5
64:84	2	X = 16	17.0594	21	5
64:84	3	X = 16	17.0594	21	5
64:84	4	X = 16	17.0594	21	5
64:84	5	X = 16	17.0594	21	5
85:105	1	X = 18	22.1944	21	3
85:105	2	X = 18	22.1944	21	3
85:105	3	X = 18	22.1944	21	3
85:105	4	X = 18	22.1944	21	3
85:105	5	X = 18	22.1944	21	3

Table B.5 Golden Section Search Results

Dataset Range of X	Iterations	Max Value X	F(x)	Calculated Value X	Calculated Value F(x)	Position Error	Error Value
1:21	1	X = 9	19.6554	X = 8	8.5964	1	11.059
1:21	2	X = 9	19.6554	X = 9	19.6554	0	0
1:21	3	X = 9	19.6554	X = 9	19.6554	0	0
1:21	4	X = 9	19.6554	X = 9	19.6554	0	0
1:21	5	X = 9	19.6554	X = 9	19.6554	0	0
22:42	1	X = 21	12.8973	X = 8	9.3020	13	3.5953
22:42	2	X = 21	12.8973	X = 6	7.6901	15	5.2072
22:42	3	X = 21	12.8973	X = 6	7.6901	15	5.2072
22:42	4	X = 21	12.8973	X = 6	7.6901	15	5.2072
22:42	5	X = 21	12.8973	X = 6	7.6901	15	5.2072
43:63	1	X = 11	15.2176	X = 7	10.5549	4	4.6627
43:63	2	X = 11	15.2176	X = 9	9.0023	2	6.2153
43:63	3	X = 11	15.2176	X = 9	9.0023	2	6.2153
43:63	4	X = 11	15.2176	X = 9	9.0023	2	6.2153
43:63	5	X = 11	15.2176	X = 9	9.0023	2	6.2153
64:84	1	X = 16	17.0594	X = 16	17.0594	0	0
64:84	2	X = 16	17.0594	X = 17	8.2705	1	8.7889
64:84	3	X = 16	17.0594	X = 17	8.2705	1	8.7889
64:84	4	X = 16	17.0594	X = 16	17.0594	0	0
64:84	5	X = 16	17.0594	X = 16	17.0594	0	0
85:105	1	X = 18	22.1944	X =11	0	7	22.1944
85:105	2	X = 18	22.1944	X =11	0	7	22.1944
85:105	3	X = 18	22.1944	X =11	0	7	22.1944
85:105	4	X = 18	22.1944	X =11	0	7	22.1944
85:105	5	X = 18	22.1944	X = 11	0	7	22.1944

Table B.6 Golden Section Search Smoothed Data

Dataset Range of X	Iterations	Max Value X	F(x)	Calculated Value X	Position Error
1:21	1	X = 9	19.6554	9	0
1:21	2	X = 9	19.6554	9	0
1:21	3	X = 9	19.6554	9	0
1:21	4	X = 9	19.6554	9	0
1:21	5	X = 9	19.6554	9	0
22:42	1	X = 21	12.8973	10	11
22:42	2	X = 21	12.8973	10	11
22:42	3	X = 21	12.8973	10	11
22:42	4	X = 21	12.8973	10	11
22:42	5	X = 21	12.8973	10	11
43:63	1	X = 11	15.2176	8	3
43:63	2	X = 11	15.2176	9	2
43:63	3	X = 11	15.2176	8	3
43:63	4	X = 11	15.2176	8	3
43:63	5	X = 11	15.2176	9	2
64:84	1	X = 16	17.0594	13	3
64:84	2	X = 16	17.0594	17	1
64:84	3	X = 16	17.0594	17	1
64:84	4	X = 16	17.0594	17	1
64:84	5	X = 16	17.0594	17	1
85:105	1	X = 18	22.1944	12	6
85:105	2	X = 18	22.1944	12	6
85:105	3	X = 18	22.1944	12	6
85:105	4	X = 18	22.1944	12	6
85:105	5	X = 18	22.1944	12	6

Appendix C

Consequential Effects

Consequential effects of this project are limited to any person who may read the report. Listed are the possible effects associated with the report.

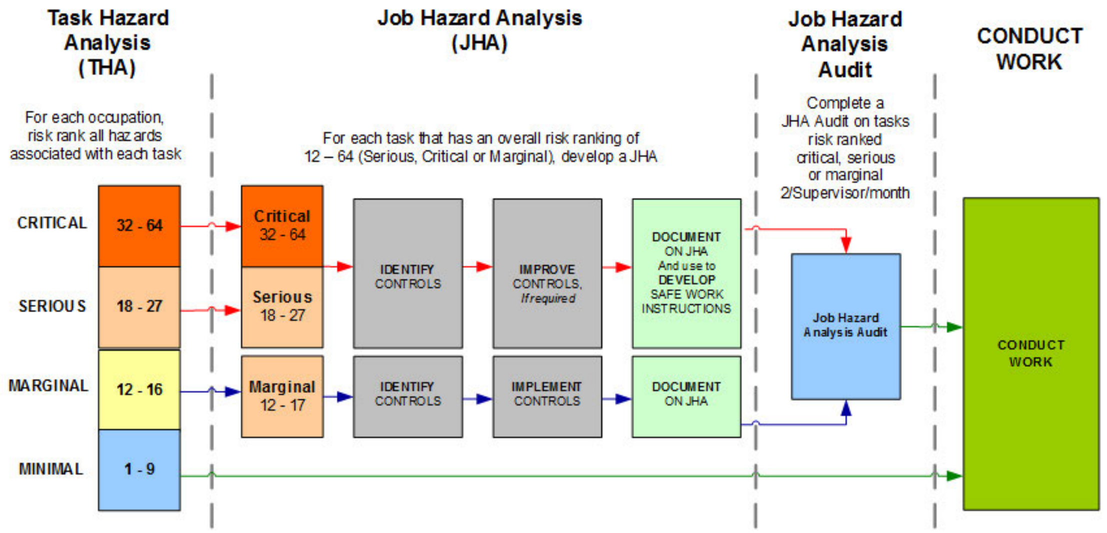
1. Giving or providing false information on a topic by not thoroughly researching and using reliable information.
2. Intellectual property breaches by sharing information publicly not intended for.
3. Not stating the intention of the report in the unlikely event that it is taken as approved research and used in the field.
4. Failing to seek peer/supervisor review.
5. Not giving credit to information sources
6. Ensuring that fraudulent activity is not committed, creating a report.
7. Ensure that outcomes of dissertation are giving thought to sustainability.

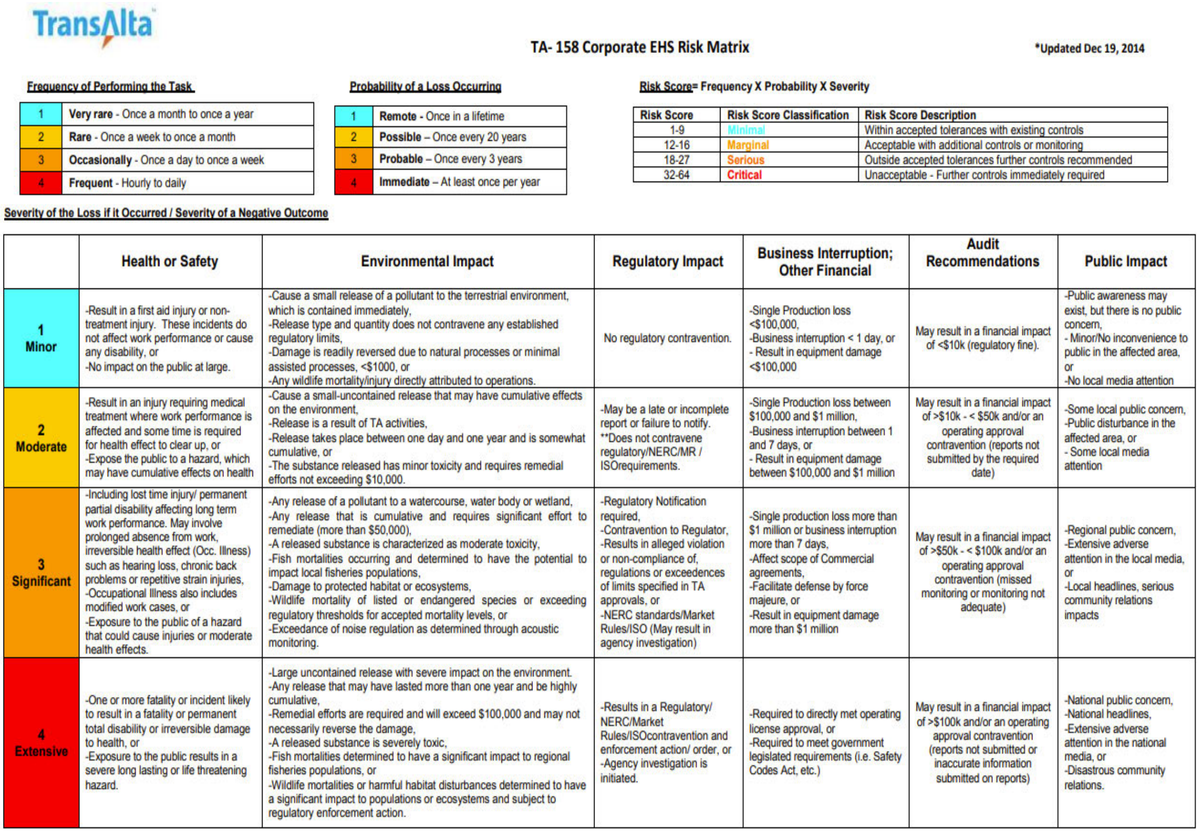
Appendix D

Risk Analaysis

Risk analysis system is provided by TransAlta corporation and remains property of TransAlta Energy Australia. This is the standard for risk assessment used by engineering and maintenance.

Process for THA/JSA Hazard Assessment and Control


Task Hazard Analysis (THA)
For each occupation, risk rank all hazards associated with each task
Job Hazard Analysis (JHA)
For each task that has an overall risk ranking of 12 – 64 (Serious, Critical or Marginal), develop a JHA
Job Hazard Analysis Audit
Complete a JHA Audit on tasks risk ranked critical, serious or marginal 2/Supervisor/month
CONDUCT WORK
CRITICAL
32 - 64
SERIOUS
18 - 27
MARGINAL
12 - 16
MINIMAL
1 - 9
Critical 32 - 64
Serious 18 - 27
Marginal 12 - 17
IDENTIFY CONTROLS
IMPROVE CONTROLS, If required
DOCUMENT ON JHA And use to DEVELOP SAFE WORK INSTRUCTIONS
IDENTIFY CONTROLS
IMPLEMENT CONTROLS
DOCUMENT ON JHA
Job Hazard Analysis Audit
CONDUCT WORK



TransAlta

TA- 158 Corporate EHS Risk Matrix

*Updated Dec 19, 2014

Frequency of Performing the Task

1	**Very rare** - Once a month to once a year
2	**Rare** - Once a week to once a month
3	**Occasionally** - Once a day to once a week
4	**Frequent** - Hourly to daily

Probability of a Loss Occurring

1	**Remote** - Once in a lifetime
2	**Possible** – Once every 20 years
3	**Probable** – Once every 3 years
4	**Immediate** – At least once per year

Risk Score= Frequency X Probability X Severity

Risk Score	Risk Score Classification	Risk Score Description
1-9	Minimal	Within accepted tolerances with existing controls
12-16	Marginal	Acceptable with additional controls or monitoring
18-27	Serious	Outside accepted tolerances further controls recommended
32-64	Critical	Unacceptable - Further controls immediately required

Severity of the Loss if it Occurred / Severity of a Negative Outcome

	Health or Safety	Environmental Impact	Regulatory Impact	Business Interruption; Other Financial	Audit Recommendations	Public Impact
1 Minor	-Result in a first aid injury or non-treatment injury. These incidents do not affect work performance or cause any disability, or -No impact on the public at large.	-Cause a small release of a pollutant to the terrestrial environment, which is contained immediately, -Release type and quantity does not contravene any established regulatory limits, -Damage is readily reversed due to natural processes or minimal assisted processes, <$1000, or -Any wildlife mortality/injury directly attributed to operations.	No regulatory contravention.	-Single Production loss <$100,000, -Business interruption < 1 day, or - Result in equipment damage <$100,000	May result in a financial impact of <$10k (regulatory fine).	-Public awareness may exist, but there is no public concern, - Minor/No inconvenience to public in the affected area, or -No local media attention
2 Moderate	-Result in an injury requiring medical treatment where work performance is affected and some time is required for health effect to clear up, or -Expose the public to a hazard, which may have cumulative effects on health	-Cause a small-uncontained release that may have cumulative effects on the environment, -Release is a result of TA activities, -Release takes place between one day and one year and is somewhat cumulative, or -The substance released has minor toxicity and requires remedial efforts not exceeding $10,000.	-May be a late or incomplete report or failure to notify. **Does not contravene regulatory/NERC/MR / ISOrequirements.	-Single Production loss between $100,000 and $1 million, -Business interruption between 1 and 7 days, or - Result in equipment damage between $100,000 and $1 million	May result in a financial impact of >$10k - < $50k and/or an operating approval contravention (reports not submitted by the required date)	-Some local public concern, -Public disturbance in the affected area, or - Some local media attention
3 Significant	-Including lost time injury/ permanent partial disability affecting long term work performance. May involve prolonged absence from work, irreversible health effect (Occ. Illness) such as hearing loss, chronic back problems or repetitive strain injuries, -Occupational Illness also includes modified work cases, or -Exposure to the public of a hazard that could cause injuries or moderate health effects.	-Any release of a pollutant to a watercourse, water body or wetland, -Any release that is cumulative and requires significant effort to remediate (more than $50,000), -A released substance is characterized as moderate toxicity, -Fish mortalities occurring and determined to have the potential to impact local fisheries populations, -Damage to protected habitat or ecosystems, -Wildlife mortality of listed or endangered species or exceeding regulatory thresholds for accepted mortality levels, or -Exceedance of noise regulation as determined through acoustic monitoring.	-Regulatory Notification required, -Contravention to Regulator, -Results in alleged violation or non-compliance of, regulations or exceedences of limits specified in TA approvals, or -NERC standards/Market Rules/ISO (May result in agency investigation)	-Single production loss more than $1 million or business interruption more than 7 days, -Affect scope of Commercial agreements, -Facilitate defense by force majeure, or -Result in equipment damage more than $1 million	May result in a financial impact of >$50k - < $100k and/or an operating approval contravention (missed monitoring or monitoring not adequate)	-Regional public concern, -Extensive adverse attention in the local media, or -Local headlines, serious community relations impacts
4 Extensive	-One or more fatality or incident likely to result in a fatality or permanent total disability or irreversible damage to health, or -Exposure to the public results in a severe long lasting or life threatening hazard.	-Large uncontained release with severe impact on the environment. -Any release that may have lasted more than one year and be highly cumulative, -Remedial efforts are required and will exceed $100,000 and may not necessarily reverse the damage, -A released substance is severely toxic, -Fish mortalities determined to have a significant impact to regional fisheries populations, or -Wildlife mortalities or harmful habitat disturbances determined to have a significant impact to populations or ecosystems and subject to regulatory enforcement action.	-Results in a Regulatory/ NERC/Market Rules/ISOcontravention and enforcement action/ order, or -Agency investigation is initiated.	-Required to directly met operating license approval, or -Required to meet government legislated requirements (i.e. Safety Codes Act, etc.)	May result in a financial impact of >$100k and/or an operating approval contravention (reports not submitted or inaccurate information submitted on reports)	-National public concern, -National headlines, -Extensive adverse attention in the national media, or -Disastrous community relations.

Appendix E

Project Risks

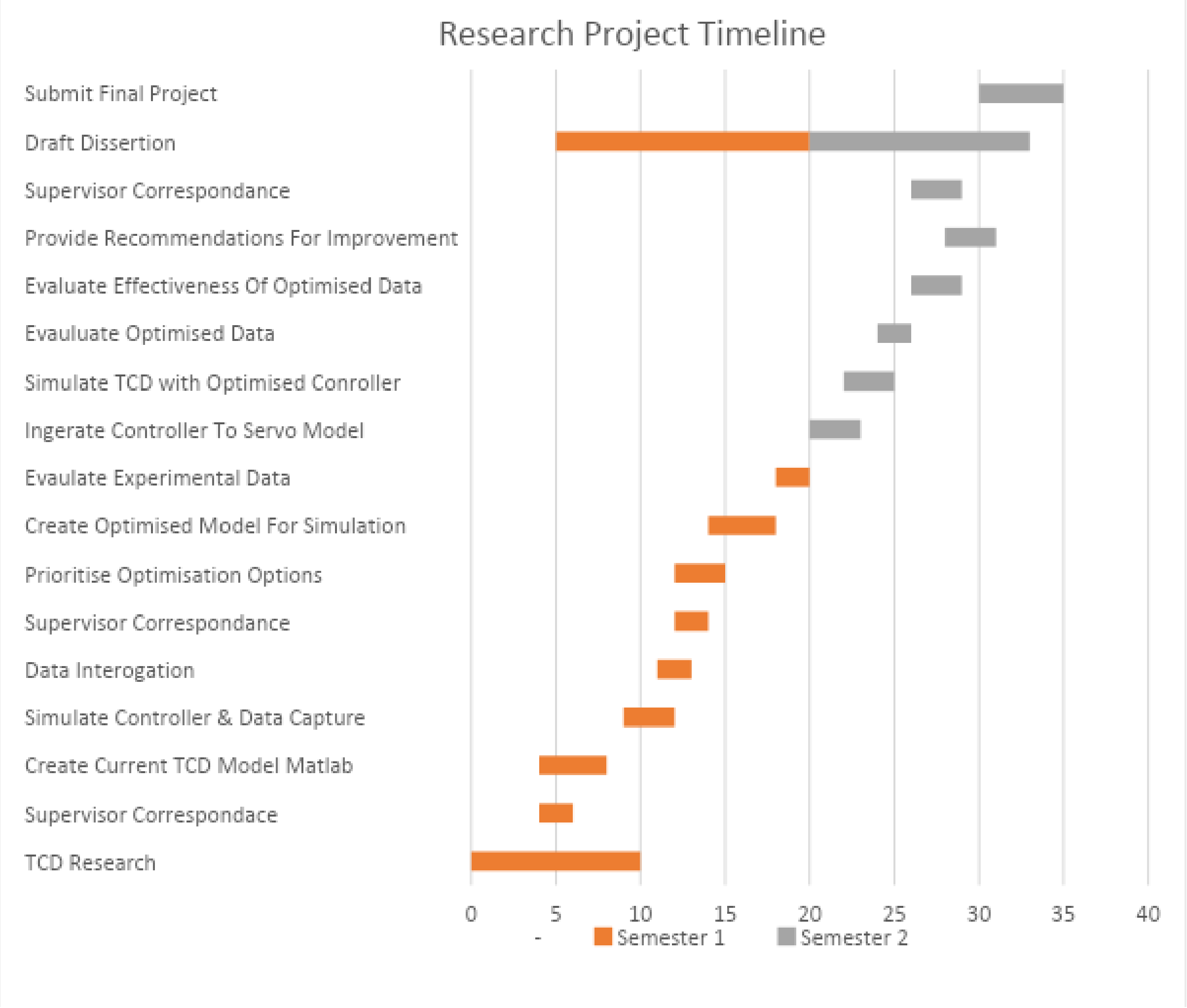
Research and Project Risks

Risk Description	Risk Mitigation	Pre-score	Post-score
Preliminary Procedures			
Incorrect review of information leading to incorrect project objectives	Review careful, confirm with course supervisor to ensure correct pursual	18	8
Expired software licenses	Ensure license is current before start of semester	13	6
Incorrect Interpolation Algorithms	Select algorithms suited to TCD US	14	7
Data Preparation			
Research insufficient, incomplete, or erroneous data	Research multiple quality sources of information	19	9
Programming mistakes causing false results	Test code multiple times with multiple inputs	32	12
Run code with incorrect inputs	Check inputs and step through program to check calculated values at each stage	33	11
Compile Data and Analyse Results			
Incorrect data cross referencing or recording	Cross reference data and double check	19	9
Determination of Controller			
Incorrect estimate of optimisation	Check against known working model	21	9
Compile Report			
Timeline not met	Keep track of progress using project plan and update constantly	33	12
Supervisor or engineering not available for review	Schedule appointments in advance and be flexible with times	23	11
Miscellaneous			
Ergonomic issues	Take regular breaks and work at ergonomic workstation	16	9
Losing focus on work requirements	Prioritize work and study balance	16	9

Risk Description	Risk Mitigation	Pre-score	Post-score
Preliminary Procedures			
Incorrect Interpolation of data resulting in incorrect medical imaging	Thorough QA/QC testing	30	8
Algorithms do not actually process data faster.	Failed project, wasted resource	13	6
Algorithms do not actually process data faster.	Legal issues with falsity	14	7
Intellectual Property protection of raw data and proposed algorithms	Legal Issues	14	7

Appendix F

Project Timeline



Appendix G

Project Stages

Table I.1 - Project Stages Summary

Stage 1	**Preliminary Procedures**
1.1	Review project description from USQ, review all course information available, review timeline and project objectives
1.2	Update and check all software for correct operation and compatibility as well as current licenses
1.3	Initiate contact with Supervisor
Stage 2	**Data Preparation**
2.1	Research and obtain data on manufacturers equipment for general overview
2.2	Research methods of data interpolation relative to TCD US
2.3	Cross reference data to ensure correctness before simulation
2.4	Run simulation and record data before and after all variable changes
2.5	Cross reference results with known values form research, confirm if program is functioning correctly
Stage 3	**Compile Data and Integrate Controller**
3.1	Review and process methods for Interpolation
3.2	Build a program with different interpolation methods, preferably 3.
Stage 4	**Results**
4.1	Compare efficiency and speed of controller
4.2	Determine if controller is optimised
Stage 5	**Compile Report**
5.1	Draft report
5.2	Review draft with course supervisor
5.3	Review draft with work engineering team
5.4	Complete Final Report

Appendix H

MATLAB code

H.1 Equal Interval Search

```
%Equal Section Search for Global Search Raw Data

%Initialise

close all;
clc;
clear all;

OD = importdata('Data3OrginalDataExtracted.mat'); %Import Global Search Data

fZ = OD(85:105,3);    %Seperate Z values

%Find maximum of data set for algorithm

E = 2; %Interval Size

a = 1;        %Lower boundry value of X
b = 21;       %Upper boundry value of X

A = fZ(1);         %Lower boundry value of Z
B = fZ(21);        %Upper boundry value of Z

n = 0;      %Record number of loops for performance evaluation
X = 0;
Max = 0;

while n < 5

        P = ceil(abs((a+b)/2 + E/2)+1);
        N = ceil(abs((a+b)/2 - E/2)+1);

        Mp = fZ(abs(P)); %Upper value of E
        Mn = fZ(abs(N)); %Lower value of E

        if Mp>Mn

            a = Mn;

            Max = Mp; %Change Upper Boundry to M+
            X = P;

        elseif Mn>Mp

                b = Mp;

                Max = Mn; %Change Lower Boundry to M-
                X = N;

        elseif Mn==Mp
```

```
                a = N-1;
                b = P+1;
        end

     n = n+1;

end

Max
X
```

H.2 An Unknown Algorithm

```
%Unknown Search for Global Search Raw Data

%Initialise

close all;
clc;
clear all;

OD = importdata('Data3OrginalDataExtracted.mat'); %Import Global Search Data

fZ = OD(85:105,3);    %Seperate Z values

%fZr = OD(1:21,3);
%N = 4;
%fZ = repelem(fZr, N); % for Optmisation

%Find maximum of data set for algorithm

G = max(fZ); %G value maximum of data set

Zmin = 1;         %Lower boundry value of X
Zmax = 21;        %Upper boundry value of X

gZmin = fZ(1);          %Lower boundry value of Z
gZmax = fZ(21);         %Upper boundry value of Z

dmin = G - gZmin; %G minus lower boundry of Z value
dmax = G - gZmax; %G minus upper boundry of Z value

n = 0;      %Record number of loops for performance evaluation

while n < 4

        T = Zmax-Zmin;
        td = dmin*T/(dmin+dmax);
        Zd = Zmin+td;

        if Zd<1

            Zd = Zd+1;
        end

        Zh = round(Zd);

        dmaxl = G - (fZ(Zh));
        dminu = dmaxl;

        %For the new interval 1 - lower

        AminL  = (dmin*dmaxl)/T;

        %For the new interval 2 - upper

        AminU  = (dminu*dmax)/T;
```

```
     %end

     if AminL < AminU

        dmax = dmaxl;
        Zmax = Zh;

     elseif AminL > AminU

        dmin = dminu;
        Zmin = Zh;

     end

     if dmax==G || dmin == G
     n = 10;
     end

     n = n+1;

end

fZ(Zmax)   %Display maximum of function
Zmax       %Display X value
```

H.3 Golden Section Search

```
%Golden Section Search for Global Search Raw Data
%Initialise
close all;
clc;
clear all;
OD = importdata('Data3OrginalDataExtracted.mat'); %Import Global Search
Data
fX = OD(1:21,1);    %Separate X probe values
fY = OD(1:21,2);    %Separate Y probe values
fZ = OD(1:21,3);    %Separate Z values
f = [fX,fZ];        %Place X and Z values in a single matrix
xL = 1;%fX(1);         %Lower boundary value of X
xU = 21;%fX(21);        %Upper boundary value of X
fL = fZ(1);         %Lower boundary value of Z
fU = fZ(21);        %Upper boundary value of Z
R = 0.5*(sqrt(5) - 1);  %Golden ratio
D = round(R*(xU - xL)); %Golden ratio applied to upper and lower boundaries
x1 = xU - D;            %Upper boundary evaluated with golden ratio
x2 = xL + D;            %Lower boundary evaluated with golden ratio
f1 = fZ(x1);    %Value of Z at adjusted boundary
f2 = fZ(x2);    %Value of Z at adjusted boundary
tol = 1e-4; %Tolerance value
err = inf;  %Error, set at infinity to ensure loop executed
n = 0;      %Record number of loops for performance evaluation
while err > tol %Loop while the err is greater than tolerance

   if f1 > f2 %When the value of f1 is greater than f2 perform the
following

       xU = x2;    %Change upper boundary
       fU = f2;    %Change upper value of Z

       x2 = x1;    %Change lower boundary
       f2 = f1;    %Change lower value of Z
       D = round(R*(xU - xL)); %Update ratio of upper and lower boundaries
       x1 = xU - D;            %Upper boundary evaluated with golden ratio

       f1 = fZ(x1);            %Value of Z at adjusted boundary

   elseif f1 < f2  %As the case above, in the case of f1 been greater than
f2

       xL = x1;
       fL = f1;

       x1 = x2;
       f1 = f2;

       D = round(R*(xU - xL));
       x2 = xL + D;

       f2 = fZ(x2);

   else
       xL = (x1 + x2)/2;
       xU = xL;

   end

   n = n+1;                        %update value of n for performance
tracking
   err = 2*abs(xU - xL)/(xU + xL); %adjust error value for loop increment

end
xef = floor((x1+x2)/2); %For rounding the value to the lower value of X
xec = ceil((x1+x2)/2);  %For rounding the value to the upper value of X
cf = fZ(xef);%The Z value at the point
cc = fZ(xec);
if cf > cc  %Taking the larger value of cf and cc

   C = cf;
```

```
else

   C = cc;

end
```

References

Ashari, A, Paryudi, I & Min, A 2013, ‘Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool’, *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 11.

Brady, AP 2016, ‘Error and discrepancy in radiology: inevitable or avoidable?’, *Insights into Imaging*, vol. 8, no. 1, pp. 171–182.

Decision Trees 2020, Cmu.edu.

Drsquo, A & Andrea, MC 2016, ‘Transcranial Doppler ultrasonography: From methodology to major clinical applications’, *World Journal of Cardiology*, vol. 8, no. 7, pp. 383–400, viewed 13 April 2021, <https://www.wjgnet.com/1949-8462/full/v8/i7/383.htm>.

Finite Element Method Basics - MATLAB & Simulink - MathWorks Australia n.d., au.mathworks.com, viewed 8 June 2021, <https://au.mathworks.com/help/pde/ug/basics-of-the-finite-element-method.html#:~:text=The%20finite%20element%20method%20describes>.

Guilford-Blake, R 2018, *Wait. Will AI Replace Radiologists After All?* www.radiologybusiness.com.

Hanson, 2021. *MCS 471 Lecture Nine*. [online] Homepages.math.uic.edu. Available at: <http://homepages.math.uic.edu/~jan/MCS471/Lec9/lec9.html> [Accessed 8 September 2021].

Harrison, O 2018, *Machine Learning Basics with the K-Nearest Neighbors Algorithm*, Medium, Towards Data Science, viewed 6 March 2021, <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>.

Haskell, R., Castelino, G. and Mirshab, B., 2021. *An Efficient Algorithm for Locating the Global Maximum of an Arbitrary Univariate Function*. [online] Soton.mpeforth.com. Available at: <http://soton.mpeforth.com/flag/jfar/vol5/no3/article1.pdf> [Accessed 3 September 2021].

Huang, Y, Song, B, Wen, P & Li, Y 2018, *An Auto TCD Probe Design and Visualization*, *University Southern Queensland*, University of Southern Queensland, Toowoomba Qld Australia, pp. 1–12, viewed 28 March 2021, <https://eprints.usq.edu.au/35907/2/Author%20submit%20version.pdf>.

Introduction to Finite Element Analysis (FEA) or Finite Element Method (FEM) n.d.,.

Jalaleddini, K, Thorpe, SG, Canac, N, Dorn, AY, Thibeault, CM, Wilk, SJ & Hamilton, RB 2019, ‘Method for Modeling Residual Variance in Biomedical Signals Applied to Transcranial Doppler Ultrasonography Waveforms’, *Neural Analytics Inc*.

Kenyhercz, MW & Passalacqua, NV 2016, *K Nearest Neighbor - an overview | ScienceDirect Topics*, www.sciencedirect.com, viewed 8 June 2021, <https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/k-nearest-neighbor>.

Kim, Y & De Weck, O 2004, *Finite Element Method*.

Naqvi, J, Yap, KH, Ahmad, G & Ghosh, J 2013, *Transcranial Doppler Ultrasound: A Review of the Physical Principles and Major Applications in Critical Care*, International Journal of Vascular Medicine.

Varghese, D 2019, *Comparative study on Classic Machine learning Algorithms*, Medium, viewed 10 June 2021, <https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222>.

Yalcin, A. and Kaw, A., 2021. *Golden Search Method*. [online] Mathforcollege.com. Available at: <http://mathforcollege.com/nm/mws/gen/09opt/mws_gen_opt_txt_goldensearch.pdf> [Accessed 8 September 2021].