

ORIGINAL RESEARCH

A low-cost Raspberry Pi based time domain reflectometer for fault detection in electric fences

Gabriel Kiarie¹  | Ciira wa Maina¹ | Kumbirayi Nyachionjeka²

¹Centre for Data Science and Artificial Intelligence (DSAIL), Dedan Kimathi University of Technology, Nyeri, Kenya

²Department of Electrical Engineering, University of Botswana, Gaborone, Botswana

Correspondence

Gabriel Kiarie, Centre for Data Science and Artificial Intelligence (DSAIL), Dedan Kimathi University of Technology, Private Bag-10143, Nyeri, Kenya.
Email: gabriel.kiarie@dkut.ac.ke

Funding information

Canada's International Development Research Centre (IDRC) through the Artificial Intelligence for Development in Africa (AI4D Africa) programme, Grant/Award Number: 109307-004; Data Science Africa (DSA) through the Affiliated Centres programme

Abstract

Electric fences used to create protected areas (PAs) are prone to faults that affect their operation. The conventional method of measuring the voltage of the fence periodically to detect faults and walking along the fence to locate the faults is inefficient and time consuming. This paper presents a low-cost Raspberry Pi time domain reflectometer (TDR) for fault detection and localisation in electric fences. The system is designed using cheap off-the-shelf components. It uses time domain reflectometry to detect hard (open and short circuit) faults in electric fences. Time domain reflectometry is a method of detecting and locating faults in electrical cables. The Raspberry Pi TDR is evaluated and it has successfully detected and located open circuit and short circuit faults in electric fences with a mean absolute error of 1.52 m. The Raspberry Pi TDR offers the potential to remotely monitor electric fences autonomously, hence improving their effectiveness.

1 | INTRODUCTION

The expansion of human activities has resulted in the loss of the natural habitat of wildlife. This, in turn, has led to the degradation of biodiversity and an increase in human–wildlife conflicts (HWCs) [1, 2]. To help solve these problems, governments and private partners have set aside protected areas (PAs) to safeguard biodiversity [3, 4]. PAs confine wildlife in a given area. By restricting the movement of animals, PAs help to reduce HWCs and make it easier to monitor and protect them [5, 6].

One way of establishing PAs is by erecting electric perimeter fences on large tracts of land. A basic electric fence comprises high tensile cables mounted on posts and connected to an energiser on one end. The energiser sends high-voltage pulses down the cables. When an animal touches the fence, it receives a non-lethal electric shock that scares it away. Over time, the animals learn to avoid the fence [7, 8].

Electric fences used in creating PAs are prone to faults that reduce their effectiveness. The faults should be detected, located, and repaired with the shortest possible downtime. The conventional way of detecting faults in electric fences involves periodic measuring of the voltage in the fence. If the measured

voltage is below a given value, the fence is most likely faulty. The condition of the power supply and output of the energiser is first checked. If the power supply and the energiser are working as expected, then the fence cables are diagnosed [9]. To locate a fault, an authorised person is required to walk along the fence measuring the voltage of the fence. The voltage will fall up to the faulty point and remain constant beyond that point [10]. This method of locating faults is time consuming and inefficient since the fences are exceptionally long [8, 11].

An alternative method of installing sensors at intervals on a fence has also been developed [12]. The sensors measure the voltage or current of the fence to detect faults. Once a fault is detected in a given section of the fence, the information is relayed wirelessly or using a communication line to the rangers. This reduces the section of the fence that needs to be inspected to locate a fault. The use of sensors greatly improves the process of detecting and locating faults in electric fences [11, 13]. However, several sensors are needed to cover all lines of the long fences, making this method expensive. The sensors and the communication line are prone to breakage, which means they also need to be monitored.

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial License](https://creativecommons.org/licenses/by-nc/4.0/), which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2024 The Authors. *IET Science, Measurement & Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

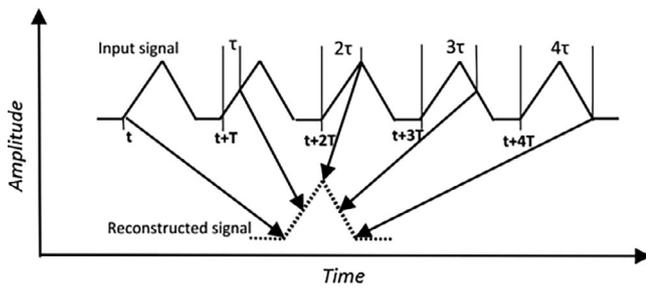


FIGURE 1 An illustration of equivalent time sampling [26].

This article proposes using time domain reflectometry to detect and locate faults in electric fences. A low-cost Raspberry Pi-based time domain reflectometer (TDR) has been developed to detect and locate open circuit and short circuit faults in electric fences. Time domain reflectometry is a method of detecting faults in electrical cables that involves applying a signal to a cable and analysing the reflected signal [14]. The reflection of electrical signals in circuits is caused by impedance mismatch. Faults introduce impedance mismatches in circuits. The magnitude and phase of the reflected signal are used to determine the type of the fault. The time delay between the incident and the reflected signals is used to compute the distance to the fault [15, 16].

The need for complex high-speed signal acquisition and processing systems makes state of the art TDRs expensive [17, 18]. Existing commercial TDRs retail at several hundreds to thousands US dollars. The TDR500/3P handheld TDR from Megger (which is a well-established brand known for making electrical test and measurement equipment), for example, costs about US\$910 at the time of writing this paper [19, 20]. There exists a need to develop simple and low-cost TDRs [21]. Several projects aimed at developing low-cost TDRs have been undertaken.

The main focus of the research that has been done on low-cost TDRs is development of inexpensive signal sampling systems. Signal sampling can be done using real-time sampling or equivalent time sampling (ETS) [22]. Real-time sampling involves acquiring the signal as it evolves. This method is, therefore, limited by the maximum sampling rate of the analogue to digital converter (ADC) in use [23]. ETS is a technique of sampling a signal in different periods using multiple triggers and then recombining the sampled data to reconstruct the original signal. ETS enables sampling of periodic signals at very high sampling rate using ADCs with low sampling rates. One basic requirement of ETS is the signal being sampled should be periodic [24]. There are two methods of ETS and they are sequential sampling and random sampling [25]. Figure 1 is an illustration of ETS.

Figure 1 shows a triangular signal with a period T is being sampled using ETS. A single sample of the signal is collected in every cycle. The ADC is triggered at sequentially increasing intervals. In Figure 1, the time interval is increased by τ each time. The samples collected are then used to reconstruct the original signal.

The concept of ETS has been used to design various TDRs. In reference [27], a Vernier clock generator comprising two crystal oscillators with slightly different frequencies (16 and 16.000312 MHz) is used to generate time delays to achieve slightly different sampling points of the incident and reflected signals. The system achieved a sampling rate of 59.5 giga samples per second (GSPS). The system uses two fixed oscillators making it less flexible to suit different applications that may require different frequencies. In reference [28], a low-cost microcontroller-based TDR has been developed for soil moisture measurement. The system uses the internal module of the ARM Cortex M4 on-board the STM32F429I to generate a 100.0079 kHz input signal and a direct digital synthesiser (DDS) to generate the ADC trigger signal. The TDR achieved a sampling rate of about 9 GSPS. The system is faced with the problem of low-quality signals hence a thousand measurements are required to compute the soil moisture. A microcontroller-based sequential sampling TDR is described in reference [29]. The TDR uses three delay lines to generate triggers for the sampling circuit and has achieved a sampling rate of 4 GSPS. One drawback of this system is that delay lines have limited number of programmable steps which results in very limited recording time that affects the accuracy of the reconstructed signal and limits the maximum length of the cable the TDR can be used to monitor. Delay lines also consume a lot of power and need to be calibrated [30]. In reference [30], an FPGA-based TDR that employs delta modulation has been developed. The use of a delta modulator in place of an ADC drastically reduces the amount of raw data that need to be collected. The system, however, suffers from the same drawback of inadaptability for long transmission line like the other low-cost ETS-based TDRs mentioned above.

ETS is a method of increasing the temporal resolution of signal acquisition systems [31]. ETS is used in state of the art digital storage oscilloscopes (DSOs) and TDRs. However, complex analogue circuits are needed to achieve the sophistication of state of the art DSOs and TDRs which drastically increase their cost [27, 28]. The low-cost ETS-based TDRs described before suffer from various shortcomings. First, synchronisation of the incident signal and the trigger signal is required which might increase the complexity of the system especially if independent signal generators are used. The choice of the signals also needs to be carefully done to avoid the wave missing phenomena [22]. Second, a lot of data needs to be collected to ensure the reconstructed signal is as close to the original signal as possible. This, in turn, results in large memory requirement and increases the time of acquisition [22, 30]. Third, the low-cost ETS were only efficient at very high frequencies. Low-frequency signals are, however, needed for long transmission lines to avoid superposing of multiple signals as a result of long time of flight of the incident and reflected signals. This makes the low-cost ETS described above suitable only for applications that have short transmission lines like testing PCBs and soil moisture [30].

To circumvent the drawbacks of ETS, a low-cost real-time sampling-based Raspberry Pi TDR was developed. The TDR attained a maximum sampling rate of 31 mega samples per second (MSPS). This translates to a maximum spatial resolution

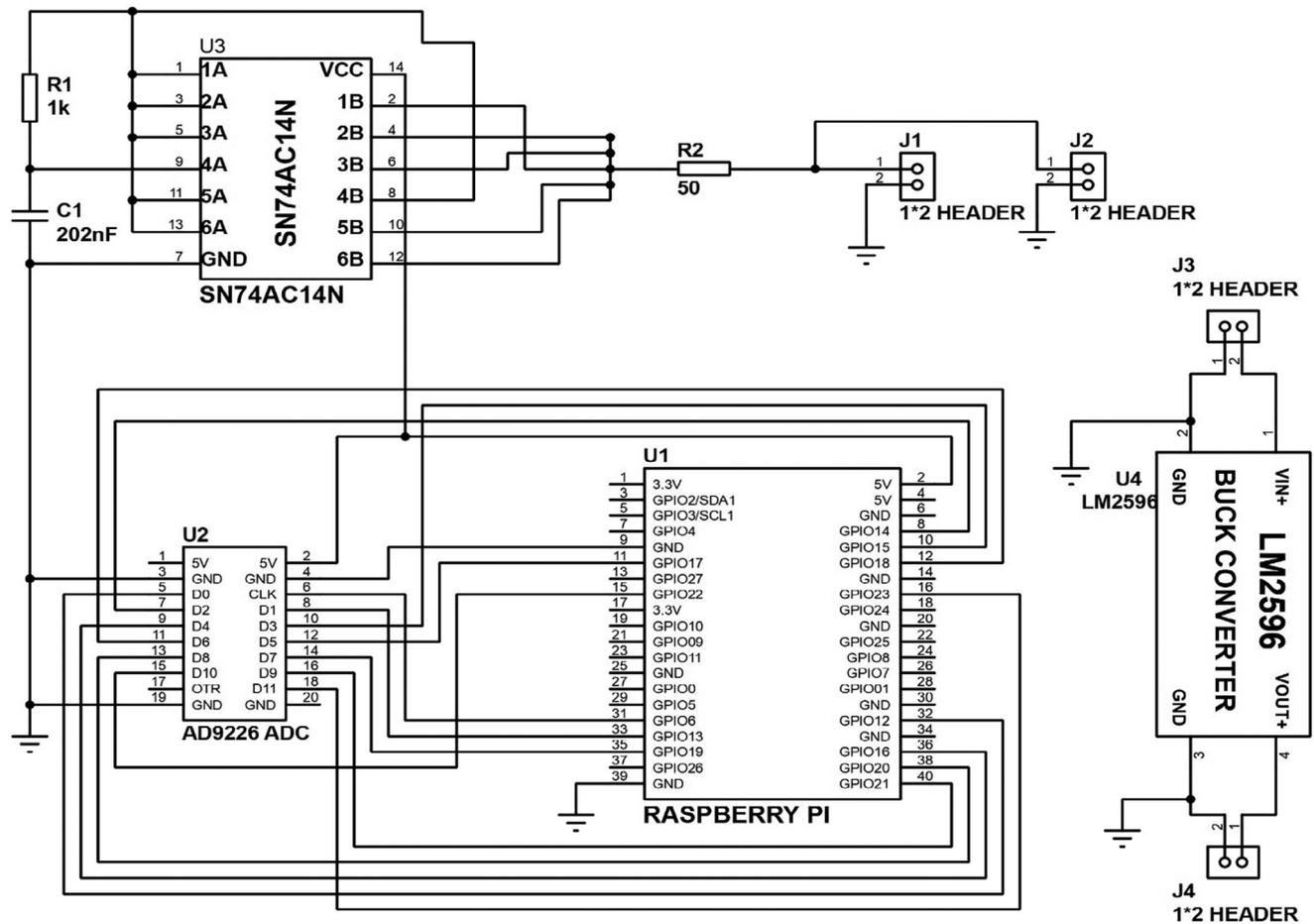


FIGURE 2 Schematic design of the Raspberry Pi time domain reflectometer.

of about 4.8 m. Due to the extensive lengths of the electric fences used in PAs it is difficult to determine the distances of points along the fences using sub-millimetre, sub-centimetre, or sub-metre resolution. From interacting with rangers from conservancies it was realised that the posts used in electric fences are mostly uniformly spaced and the rangers use this prior knowledge to estimate the distance of points along the fences from the starting point. A low-resolution TDR is, therefore, effective for monitoring electric fences. The distance to a fault predicted by the Raspberry Pi TDR will have a precision of about ± 4.8 m. This is a very short distance to search for a fault compared to the distance the rangers have to walk when using the conventional way of locating faults in electric fences. The Raspberry Pi TDR was developed using readily available cheap off-the-shelf components. The TDR uses very few components making it cost less than US\$100.

2 | METHODOLOGY

2.1 | System architecture

The Raspberry Pi TDR consists of hardware and software. The following subsections describe each of these parts.

2.1.1 | Hardware

The hardware components of the Raspberry Pi TDR include a Schmitt trigger signal generator, an AD9226 ADC module, a Raspberry Pi 4, a cable interface port and a buck converter. Figures 2 and 3 show the schematic design of the circuit and the parts of the TDR on a printed circuit board (PCB), respectively.

Schmitt trigger signal generator

A Schmitt trigger inverter was used to generate a fast-rising step signal to be applied to the cable under test (CUT). The step signal used in time domain reflectometry should have a fast rise time. An SN74AC14N Schmitt trigger inverter was used as the step signal generator. SN74AC14N has six independent inverters with a fast rise time [32]. The Schmitt trigger was configured into a bistable multivibrator by connecting an RC circuit between the output and the input of one inverter. The output of the inverter was then fed to the remaining five inverters. The outputs of the five inverters were combined to increase the power of the generated step signal. Combining the outputs of the five inverters increases the maximum power of the step signal the Schmitt trigger can generate from 0.12 to 0.6 W. Increasing the power of the step signal helps to compensate for

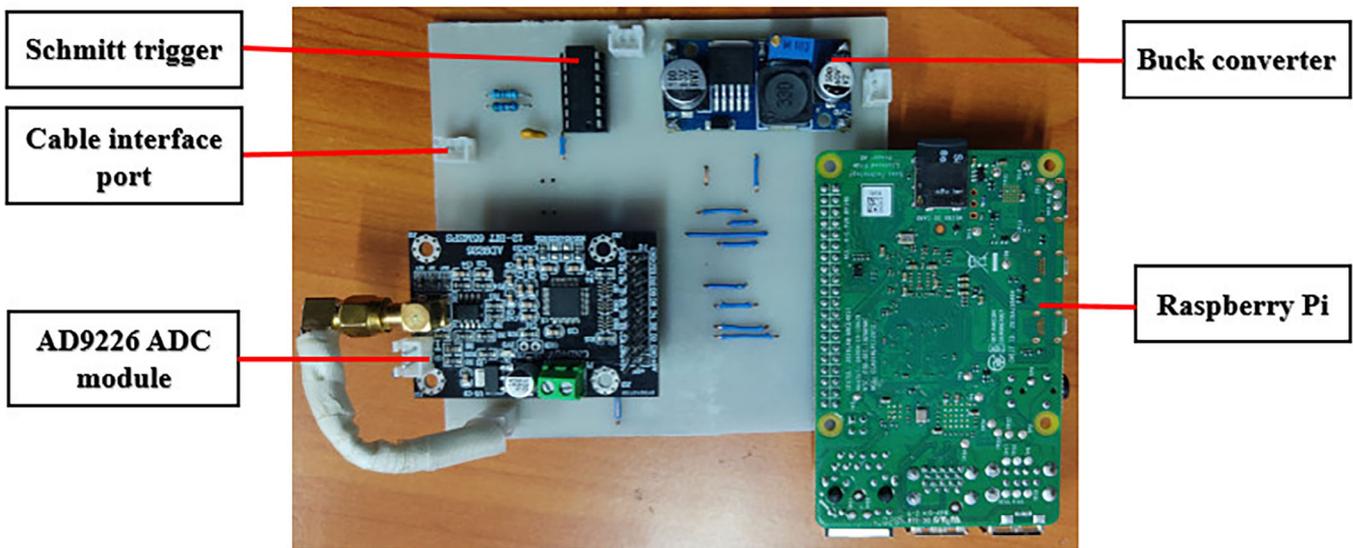


FIGURE 3 Parts of the Raspberry Pi time domain reflectometer.

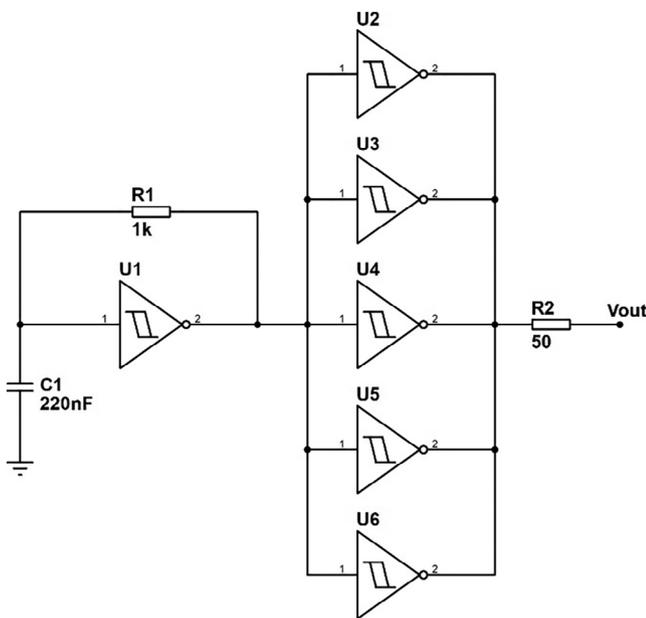


FIGURE 4 Configuration of the SN74AC14N Schmitt trigger inverter signal generator.

the effect of attenuation caused by the CUT. Figure 4 illustrates the configuration of the Schmitt trigger signal generator.

The values of $R1$ and $C1$ used are $1\text{ k}\Omega$ and 220 nF , respectively. The signal generator generates an 8.9 kHz step signal with an amplitude of 5 V and a rise time of 35.5 ns .

AD9226 ADC module

The AD9226 is a cheap fast 12-bit ADC with a parallel interface. It has a maximum sampling rate of 65 MSPS . It requires a 5 V supply and has a 3.3 V logic interface allowing it to be easily interfaced with the Raspberry Pi. The ADC can sample signals

of up to $\pm 5\text{ V}$ [33]. It samples the signal at the input of the CUT and feeds it to the Raspberry Pi.

The Raspberry Pi

The Raspberry Pi is the central device of the TDR system. It takes the output of the ADC and processes it to detect faults in cables. The ADC is interfaced with the Raspberry Pi using the Secondary Memory Interface (SMI). SMI is a general-purpose high-speed parallel interface that is found in all Raspberry Pi versions. The ability to grab all data bits at once in a single clock cycle makes SMI faster than the other serial interfaces found in the Raspberry Pi [34].

Direct memory access (DMA) was used to transfer the data from the general-purpose input/output (GPIO) pins to the storage. DMA is a method of moving data around a computer at high speeds independent of the CPU [35, 36]. Using a Raspberry Pi 4 interfaced with an AD9226 ADC module, a sampling rate of 31 MSPS was achieved.

Buck converter

The buck converter is used to step down the voltage from a $12/24\text{ V}$ battery to 5 V to power the TDR system. Some fences are powered by batteries, which can also be used to power the TDR system through the buck converter.

Table 1 shows the cost of components used to make the Raspberry Pi TDR.

2.1.2 | Software

The software for the Raspberry Pi TDR system is divided into two.

- (i) Data acquisition programmes
- (ii) Time domain reflectometry algorithm

TABLE 1 Cost of the components used to produce the Raspberry Pi time domain reflectometer.

Component	Number	Cost per unit (US\$)	Total cost (US\$)
Raspberry Pi 4	1	55	55
AD9226 ADC	1	35	35
SN74AC14N Schmitt trigger	1	0.67	0.67
50 Ω resistor	1	0.014	0.02
1 k Ω resistor	1	0.014	0.02
200 μF capacitor	1	0.036	0.04
LM2596 buck converter	1	1.44	1.44
JST connector	3	0.14	0.42
Female double-row header pin	1	0.22	0.22
14-Pin DIP IC socket	1	0.14	0.14
Copper clad board	1	0.86	0.86
Total cost (US\$)			93.83

Data acquisition programmes

The data acquisition programmes are responsible for driving the ADC and transferring its output to the storage of the Raspberry Pi. The programmes are written in C programming language to achieve high data transfer speed. The sampled signal is stored as a CSV file. Each file contains 10,000 samples, which is approximately equal to 0.3-ms-long sampled signal. The programmes used for data acquisition in the Raspberry Pi TDR were adapted from programmes developed for SMI by Bentham [22].

Time domain reflectometry algorithm

The time domain reflectometry algorithm processes the sampled signal to detect a fault and determine its type and location. The time domain reflectometry algorithm should be fast, accurate, and computationally cheap enough to run on the Raspberry Pi. To detect a fault, the time domain reflectometry algorithm checks for the presence of a reflected signal on the sampled signal. Figure 5 shows the reflection of an incident pulse by a short circuit and an open circuit.

Figure 5 shows the reflection of electric signals in a short circuit and an open circuit. The reflected signal is out of phase with the incident signal in a short circuit and in phase with the incident signal in an open circuit. The phase of the reflected signal is governed by Equation (1).

$$\Gamma = \frac{Z_1 - Z_c}{Z_1 + Z_c} \quad (1)$$

where Γ is the reflection coefficient, Z_c is the characteristic impedance of the cable, and Z_1 is the impedance of the load.

If a step signal is applied to an open-circuited or short-circuited cable, the reflected signal is also a step signal. Consider a single cycle of a step signal with an amplitude of 1 V, a period of T , and a 50% duty cycle as shown in Figure 6.

If the pulse is applied to an open circuit or a short circuit, it travels to the end of the circuit where it encounters a change in

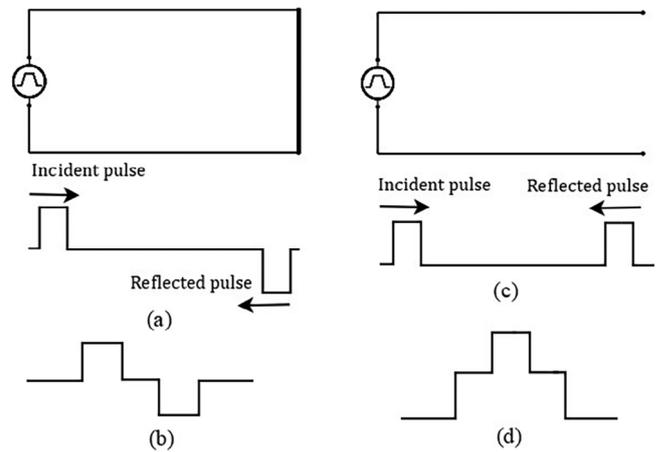


FIGURE 5 (a) Reflection of a step pulse by a short circuit; (b) resulting signal due to interference between the incident and the reflected signal in a short circuit; (c) reflection of a step pulse by an open circuit; and (d) resulting signal due to interference between the incident and reflected signal in an open circuit.

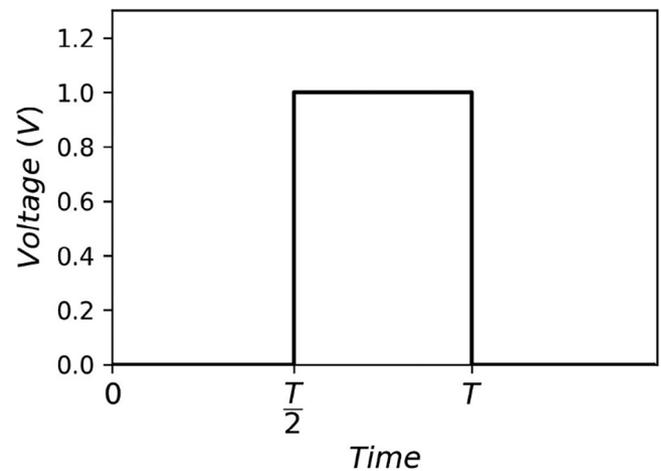


FIGURE 6 A step signal with an amplitude of 1 V, period of T , and a 50% duty cycle.

impedance resulting in its reflection. The reflected pulse travels from the end of the circuit to the input. Let Δt be the time taken by the pulse to travel from the input to the end of the circuit and back. If Δt is less than half the period of the pulse, that is, $\Delta t < 0.5T$, then the reflected pulse will arrive at the input while part of the incident pulse is still incoming. The word pulse is used to refer to the HIGH section of the step signal in this case.

The arrival of the reflected pulse at the input with part of the incident pulse still incoming, results in interference between the two. The interference between the incident and the reflected signals results in change points. The time domain reflectometry algorithm searches for these change points to detect the reflected signal. The pulse reflected by an open circuit results in constructive interference and that reflected by a short circuit results in destructive interference. Figures 7 and 8 show the interferences of the incident and reflected pulses in an open circuit and a short circuit, respectively.

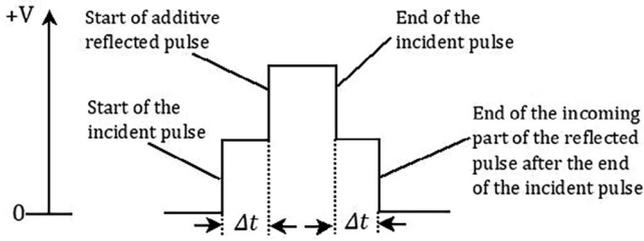


FIGURE 7 Resultant signal due to interference between an incident and a reflected pulse in an open circuit.

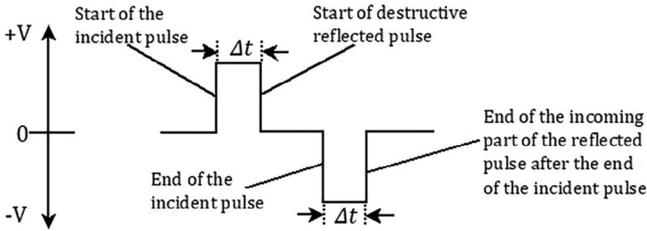


FIGURE 8 Resultant signal due to interference between an incident and a reflected pulse in a short circuit.

The interference between the incident and the reflected signals results in stepped signals as shown in Figures 7 and 8. The first change point on the signal is the rising edge of the incident pulse. The second change point occurs after Δt from the first change point due to the arrival of the reflected pulse. The third change point occurs at the falling edge of the incident pulse. The fourth change point occurs after Δt from the falling edge of the incident pulse and is the falling edge of the reflected pulse that is still incoming after the end of the incident pulse. These change points are used to detect the reflected pulse and determine its time delay.

The change points are stepped, meaning they can be detected using the method of edge detection in step signals. One way of detecting edges in step signals is using the method of discrete gradient [37]. The discrete gradient of a step signal is zero everywhere except at the rising and falling edges. The discrete gradient is computed by taking the finite difference of the sampled signal, which is a computationally cheap process.

If $\{v_t\}$, for $0 \leq t \leq T$ is a sampled signal that contains an incident and a reflected pulse, its finite difference, $\{\Delta v_t\}$, for $0 \leq t \leq T - 1$, is given by Equation (2).

$$\{\Delta v_t\} = |v_{t+1} - v_t| \text{ for } 0 < t \leq T - 1 \quad (2)$$

Figures 9 and 10 show the plots of signals with reflections due to an open circuit and a short circuit, respectively, and their discrete gradients for a line that is lossless and matched to the source.

The stem plots in Figures 9 and 10 show the discrete gradients of signals containing the incident and reflected pulses. The discrete gradient is zero everywhere except at the rising and falling edges of the incident and reflected pulses. This property is used to detect the change points of interest. Using the loca-

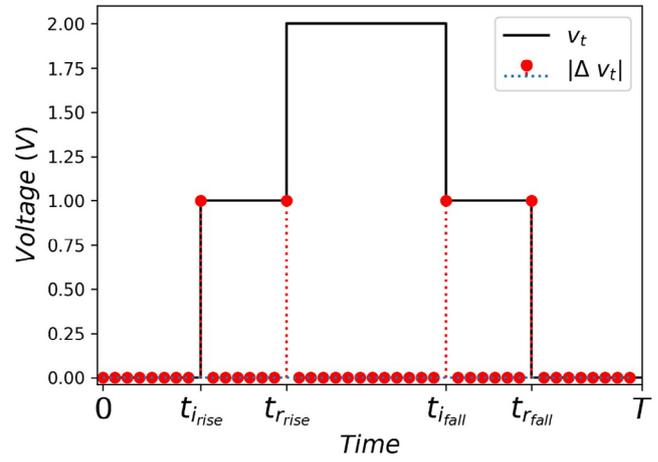


FIGURE 9 A plot of a signal containing an incident step pulse and a reflected step pulse due to an open circuit and its discrete gradient.

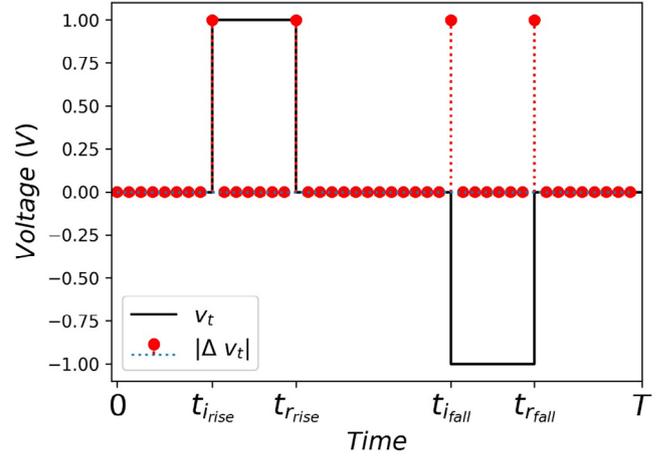


FIGURE 10 A plot of a signal containing an incident step pulse and a reflected step pulse due to a short circuit and its discrete gradient.

tions of the change points, the time delay of the reflected signal is given by Equation (3).

$$\Delta t = t_{r_{rise}} - t_{i_{rise}} = t_{r_{fall}} - t_{i_{fall}} \quad (3)$$

where $t_{i_{rise}}$ is the time of the rising edge of the incident pulse; $t_{r_{rise}}$ is the time of the rising edge of the reflected pulse; $t_{i_{fall}}$ is the time of the falling edge of the incident pulse; and $t_{r_{fall}}$ is the time of the falling edge of the reflected pulse. To compute the time delay of the reflected pulse, the correct pair of change points needs to be used as shown in Equation (3). The pairs are the rising edges of the incident and reflected pulses pair or the falling edges of the incident and reflected pulses pair.

In Figures 9 and 10, the incident signal is wholly reflected. For this to happen, the line needs to be lossless and matched with the source. It is, however, practically impossible to achieve a lossless line [38, 39]. A lossy line attenuates a signal as it travels in it [40]. The reflected pulse will therefore have a smaller amplitude than the incident pulse in a lossy line. This means that the

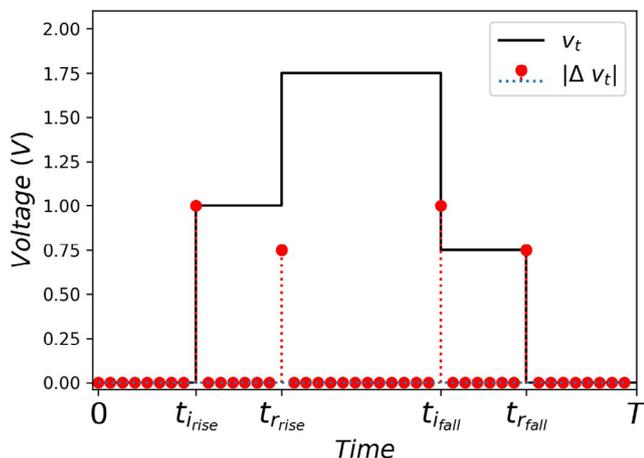


FIGURE 11 A plot of a signal containing an incident step pulse and a reflected step pulse due to an open circuit in a lossy line and its discrete gradient.

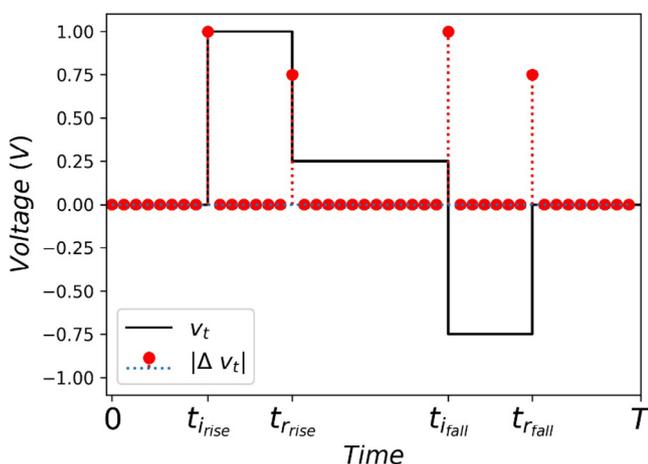


FIGURE 12 A plot of a signal containing an incident step pulse and a reflected step pulse due to a short circuit in a lossy line and its discrete gradient.

discrete gradients of the falling and rising edges of the reflected pulse will be less in magnitude than those of the edges of the incident signal.

Consider a case where the reflected pulse has an amplitude that is 75% of the amplitude of the incident pulse. Figures 11 and 12 show the plots of the signals with reflections due to an open circuit and a short circuit, respectively, and their discrete gradients.

The discrete gradients of the rising and falling edges of the incident pulse are large compared to those of the reflected pulse as shown in Figures 11 and 12. The time domain reflectometry algorithm uses this property to identify the edges of the incident signal and those of the reflected signal separately and then pair them appropriately to compute the time delay of the reflected signal.

Using a predetermined threshold, the indices of the rising and falling edges of the incident signal can be obtained from the discrete gradient of the sampled signal. Equation (4) is used to generate a binary vector, I , of 1s and 0s where the indices of 1s

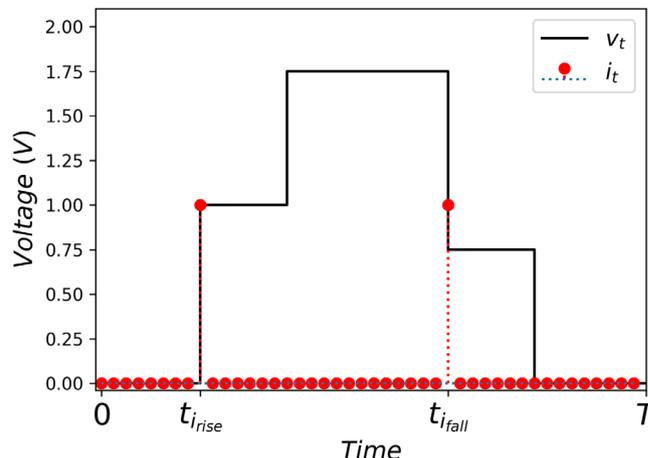


FIGURE 13 Generated binary vector showing the indices of the edges of the incident pulse in an open circuit.

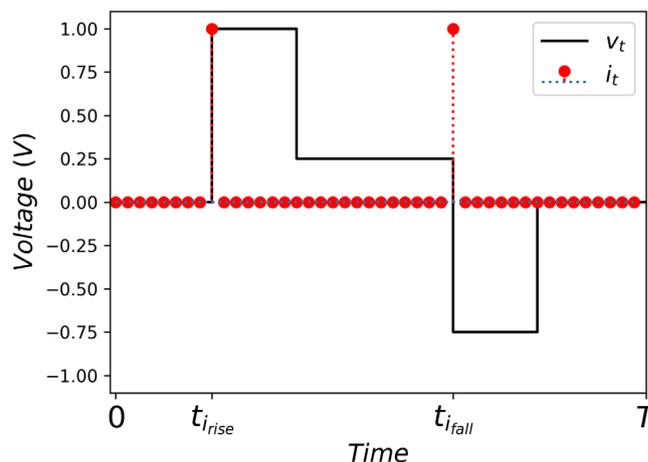


FIGURE 14 Generated binary vector showing the indices of the edges of the incident pulse in a short circuit.

represent the indices of the edges of the incident signal.

$$i_t = \begin{cases} 1, & \text{if } \Delta v_t > v_{th} \\ 0, & \text{otherwise} \end{cases} \quad \text{for } 0 \leq t \leq T - 1 \quad (4)$$

where v_{th} is the threshold value.

Figures 13 and 14 show the plots of signals and the obtained binary vectors, I , using a threshold of 0.75 that is equivalent to the ratio of the amplitude of the reflected pulse to the amplitude of the incident pulse.

The stem plots in Figures 13 and 14 show the binary vectors obtained from the discrete gradients of the signals from an open circuit and a short circuit, respectively. The binary vectors have values of zeros everywhere except at the indices of the rising and falling edges of the incident signals. Using the generated binary vector, the time domain reflectometry algorithm can locate the edges of the incident signal.

Next, the time domain reflectometry algorithm searches for the edges of the reflected pulses. The choice of the incident

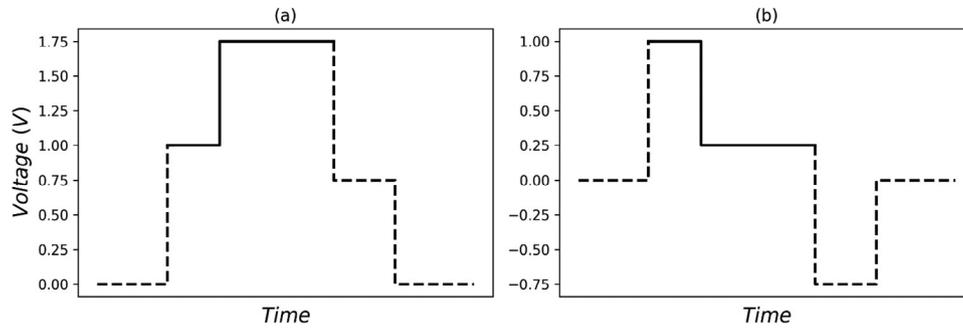


FIGURE 15 Signal segments generated from the end of the rising edge of the incident pulse to the start of the falling edge of the incident pulse for signals sampled from (a) an open circuit and (b) a short circuit.

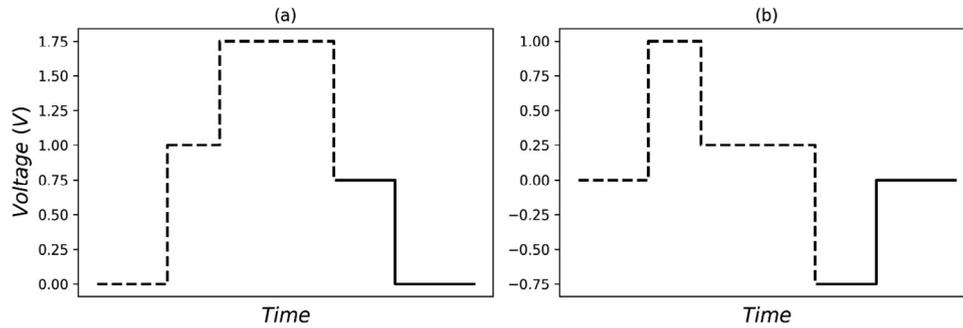


FIGURE 16 Signal segments generated from the end of the falling edge of the incident pulse to the start of the rising edge of the next cycle of the incident signal for signals sampled from (a) an open circuit and (b) a short circuit.

pulse used in the Raspberry Pi TDR is such that for a given cycle, the rising edge of the reflected pulse will be located within the HIGH section of the incident step signal and the falling edge will be located within the LOW section of the incident step signal as shown in Figures 7 and 8. This was achieved by using a signal with a duty cycle that is almost 50% and whose half period is greater than the expected approximated maximum time delay of the reflected signal. This property ensures that the reflected pulse that corresponds to a given incident cycle does not spread over to another cycle. This prior knowledge of the possible locations of change points due to the reflected signal makes it easy to search for them.

Using the located indices of the rising and falling edges of the incident signal, signal segments are generated. The segments are generated based on the knowledge of the possible locations of the change points due to the reflected signal. This makes it easier to pair the change points for the correct time delay computation. One signal segment is generated from the end of a rising edge to the start of the falling edge of the incident signal. This segment is expected to contain the rising edge of the reflected pulse. The second signal segment is generated from the end of the falling edge to the start of the rising edge of the next cycle of the incident signal. This segment is expected to contain the falling edge of the reflected pulse. Figures 15 and 16 show the signal segments that are generated to find the edges of the reflected pulses. The solid sections of the plots are the signal segments.

The number of samples, N , in each segment is given by Equation (5).

$$N = k \times \frac{1}{f} \times f_s \quad (5)$$

where f is the frequency of the step signal and f_s is the sampling rate of the signal.

$$k = \begin{cases} \text{duty cycle}, & \text{if duty cycle} \leq 0.5 \\ 1 - \text{duty cycle}, & \text{if duty cycle} > 0.5 \end{cases} \quad (6)$$

The same procedure for locating the edges of the incident signal is used to detect the edges of the reflected signal in the signal segments. If $\{s_n\}$, for $0 \leq n \leq N$ is a signal segment, its finite difference, $\{\Delta s_n\}$, for $0 \leq n \leq N - 1$, is given by Equation (7).

$$\{\Delta s_n\} = |s_n - s_{n+1}| \text{ for } 0 \leq n \leq N - 1 \quad (7)$$

Equation (8) is used to generate a binary vector, R , containing 0s and a 1 at the index of the edge of the reflected signal.

$$r_n = \begin{cases} 1, & \text{if } \Delta s_n \geq s_{th} \\ 0, & \text{otherwise} \end{cases} \text{ for } 0 \leq n \leq N - 1 \quad (8)$$

where s_{th} is the threshold value

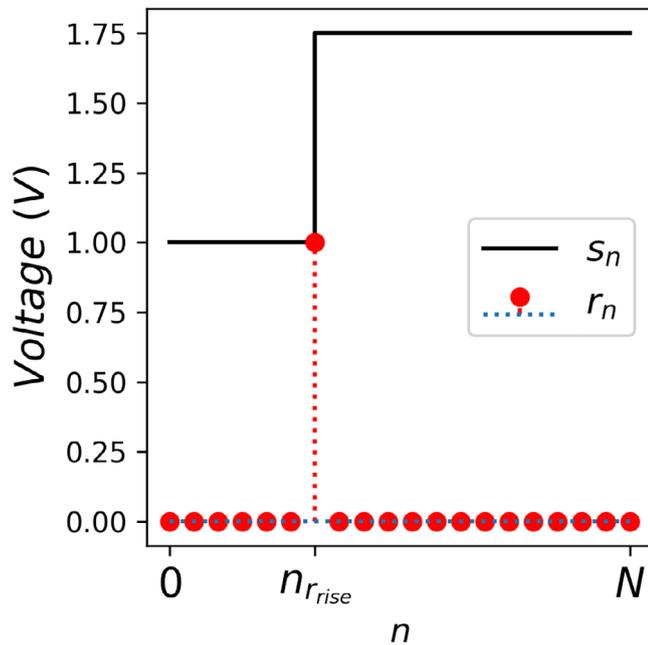


FIGURE 17 Computed binary vector showing the index of the rising edge of the reflected pulse in an open circuit.

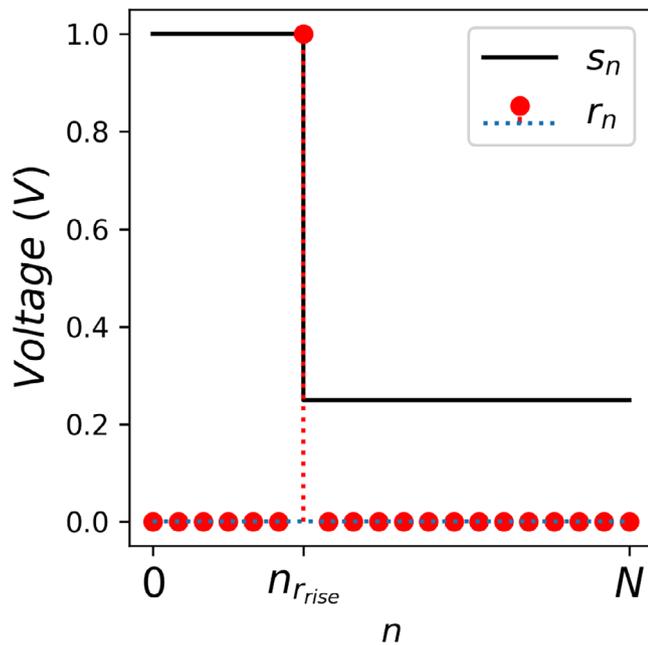


FIGURE 18 Computed binary vector showing the index of the rising edge of the reflected pulse in a short circuit.

Signal segments shown in Figure 15 will be used to explain the next steps. Figures 17 and 18 show the signal segments and the binary vectors, R , computed from them.

Figures 17 and 18 show the plots of R vectors calculated from signal segments that contain the rising edges of the reflected pulses. The R vectors contain zeros everywhere except at the edges of the reflected pulses. The spike in the stem plot of the binary vector corresponds to the index of the change point due

to the reflected pulse on a signal segment. The index represents the number of samples between the edges of the incident and reflected pulses. If n_{rise} and n_{fall} are the indices of the rising edge and falling edge, respectively, of the reflected pulse in a signal segment, the time delay, Δt , is given by Equation (9).

$$\Delta t = \frac{n_{\text{rise}}}{f_s} = \frac{n_{\text{fall}}}{f_s} \quad (9)$$

The time delay obtained in Equation (9) is used to calculate the distance to the fault. The distance to the fault, d_f , is given by Equation (10).

$$d_f = \frac{\text{VF} \times c \times \Delta t}{2} \quad (10)$$

where VF is the velocity factor of the CUT and c is the velocity of light in vacuum.

To determine the type of fault, the time domain reflectometry algorithm checks for the presence of negative going peaks. This is done by checking if the sampled signal has values that are less than -1 . Their presence means the fault is a short circuit and an open circuit if otherwise. Figure 19 shows the flowchart of the time domain reflectometry algorithm.

2.2 | Preliminary experiments

Preliminary tests were conducted to ascertain if the Raspberry Pi TDR can be used in fault detection and localisation. Two 15-m-long silicone-rubber-insulated copper cables, an oscilloscope, and the Raspberry Pi TDR were used in these experiments. The VF of a silicone-rubber-insulated copper cable is about 0.67 [41]. A step signal generated by the Schmitt trigger on the Raspberry Pi TDR was applied to one end of the cables, while open circuit and short circuit faults were simulated on the other end of the cables to reflect the incident signal. The incident and reflected signals were measured at the input using the oscilloscope and the Raspberry Pi TDR and the results were compared. Figure 20 shows the setup for testing time domain reflectometry using an oscilloscope, the Raspberry Pi TDR, and two 15-m-long silicone-rubber-insulated copper cables.

2.3 | Experiments with electric fences

Next, the Raspberry Pi TDR was tested with electric fences. A 106-m-long section of an electric fence at the Dedan Kimathi University of Technology Wildlife Conservancy (DeKUTWC) and a 280-m-long section of an electric fence at Ol Pejeta Conservancy were used for these experiments. A set of two cables for each of the fences was used to simulate open and short circuit faults. The entire fence section was treated as an open circuit in both cases to simulate an open circuit fault. Only two open circuit fault instances could be simulated. Short circuits were introduced at intervals along the fences by connecting the two cables with a wire.

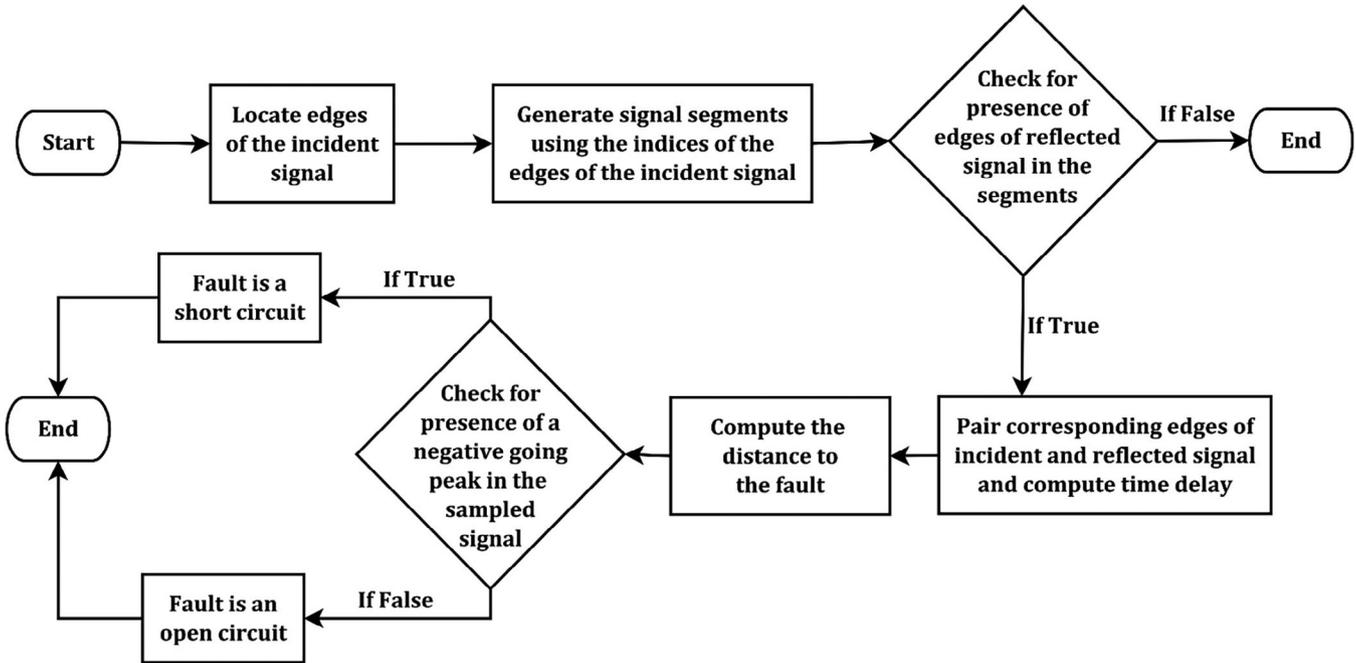


FIGURE 19 Flowchart of the time domain reflectometry algorithm.

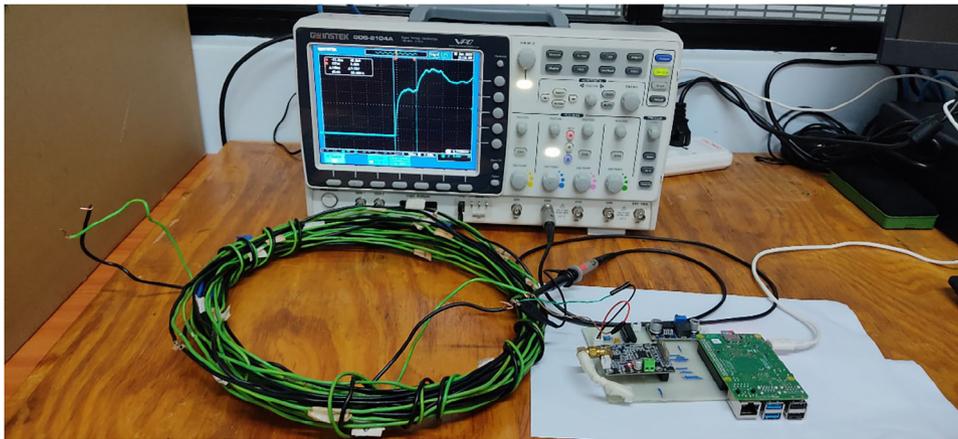


FIGURE 20 Setup for testing time domain reflectometry using an oscilloscope and the Raspberry Pi time domain reflectometer.

For the DeKUTWC fence, short circuit faults were created at points approximately midway between posts and immediately after each post. Nineteen short circuit fault instances were created using the DeKUTWC fence. For the Ol Pejeta Conservancy fence, short circuit faults were introduced after every five posts. Ten short circuit fault instances were created using the Ol Pejeta Conservancy fence.

Using the Raspberry Pi TDR, a step signal was applied to the cables and about 0.3-ms-long signal was sampled at the input and saved on the storage of the Raspberry Pi as a CSV file. For each fault simulation, several files were collected. The files collected from these experiments were 581. Forty-seven files are signals sampled from the open-circuited electric fence setups. The rest are signals sampled from the electric fences with short circuits at different points. Figure 21 shows the TDR connected to the electric fence at the DeKUTWC.

3 | RESULTS AND DISCUSSION

First, the results of the preliminary experiments with the oscilloscope and copper cables were analysed. Figures 22 and 23 show the signals measured by the oscilloscope when a step signal was applied to one end of the test cables, while the other end was open-circuited.

Figure 23 shows the rising edges of the incident and reflected signals. The interference of the reflected signal is constructive because it is in phase with the incident signal. The measured time delay of the reflected signal is 148 ns.

Figures 24 and 25 show the signals measured by the oscilloscope when a step signal was applied to one end of the test cables, while the other end was short-circuited.

Figure 24 shows the signal measured at the input of the short-circuited cables. The reflected signal is out of phase with the



FIGURE 21 The Raspberry Pi time domain reflectometer connected to the electric fence at the Dedan Kimathi University of Technology Wildlife Conservancy.

incident signal, hence the destructive interference as shown in Figure 25. The measured time delay of the reflected signal is also 148 ns. In both the open circuit and short circuit cases, there are secondary reflections that exhibit themselves as ripples after the first reflection due to a mismatch between the signal generator and test cables. The distance to the fault (length of the cables) is computed using Equation (11).

$$d_f = \frac{0.67 \times 3 \times 10^8 \times 148 \times 10^{-9}}{2} = 14.87 \text{ m} \quad (11)$$

The value computed in Equation (11) has an absolute error of 0.13 m.

Second, the signals collected using the Raspberry Pi TDR from the preliminary experiments with the copper cables were processed using the time domain reflectometry algorithm. Figure 26 shows the step signal generated and sampled by the Raspberry Pi TDR.

The step signal shown in Figure 26 has an amplitude of about 2.5 V, but the on-board signal generator of the Raspberry Pi TDR generates a step signal with an amplitude of 5 V. The measured amplitude of the step signal is less since the input resistance of the AD9226 ADC module and the output resistance of the signal generator are both about 50Ω forming a voltage divider. This means that the voltage across the input of the ADC will be approximately half the voltage produced by the signal generator.

Figures 27 and 28 show the signal sampled by the Raspberry Pi TDR during the simulation of an open circuit with the test cables.

Figure 27 shows the signal sampled at the input of the open-circuited cable. A zoomed section of the signal containing the rising edges of the incident and the reflected signal is shown in Figure 28. The reflected signal is additive since it is in phase with the incident signal.

Figures 29 and 30 show the signal sampled by the Raspberry Pi TDR during the simulation of a short circuit on the test cables.

Figure 29 shows the signal sampled at the input of the short-circuited cable. A zoomed section of the signal containing the rising edges of the incident and the reflected signal is shown in Figure 30. The reflected signal is subtractive since it is out of phase with the incident signal.

The signals sampled by the Raspberry Pi TDR are similar to the ones measured by the oscilloscope for both the open circuit and short circuit. The ripples after the first interference

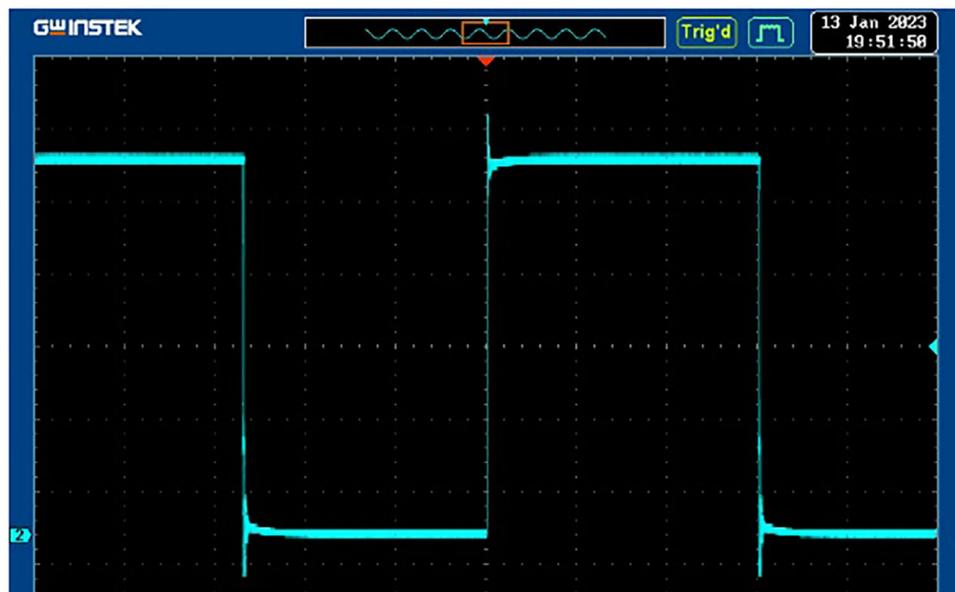


FIGURE 22 Signal measured at the input with test cables open-circuited.

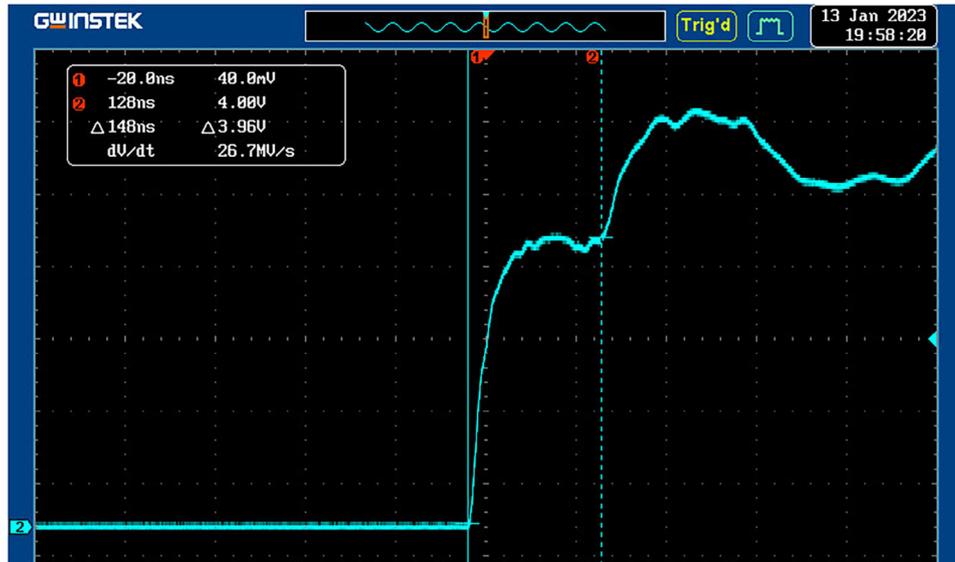


FIGURE 23 A zoomed section of the signal measured at the input with the cables open-circuited showing the additive reflected pulse.

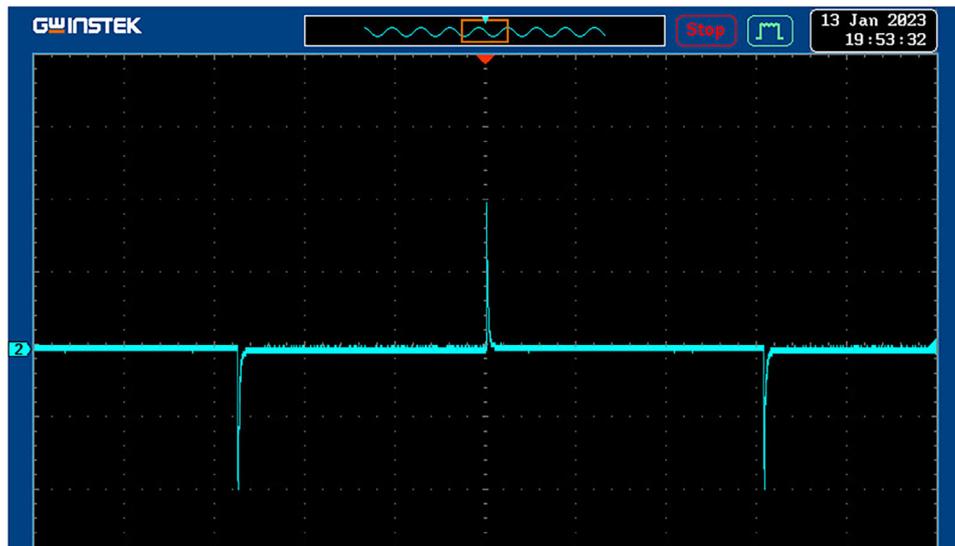


FIGURE 24 Signal measured at the input with test cables short-circuited.

are due to secondary reflections. The signal segments shown in Figures 28 and 30 will be used to show the results of the next steps of the time domain reflectometry algorithm.

The discrete gradients of the sampled signals were computed using the time domain reflectometry algorithm. Figures 31 and 32 show the discrete gradients of the signal segments shown in Figures 28 and 30, respectively.

The stem plots in Figures 31 and 32 show the discrete gradients of signal segments containing the incident and reflected signals. The discrete gradients increase at the edges of the incident and reflected signals. The discrete gradients are the largest at the edges of the incident signals. For each signal segment, the next largest discrete gradient is at the edge of the first reflected signal.

The binary vector, I , is computed from the discrete gradient using Equation (4) to locate the edges of the incident signal. The standard deviation of the step signal generated and sampled by the TDR (1.18) was used as the threshold to compute the binary vector. Figures 33 and 34 show the plots of the sampled signals along with their corresponding binary vectors.

Figures 33 and 34 show the plots of the binary vectors computed from the signals sampled from the open circuit and short circuit setups, respectively. The spikes in the binary vectors correspond to the indices of the falling and rising edges of the incident signals. The time domain reflectometry algorithm, thus, successfully located the rising and falling edges of the incident signals from the signals sampled from both the open and short circuit setups.

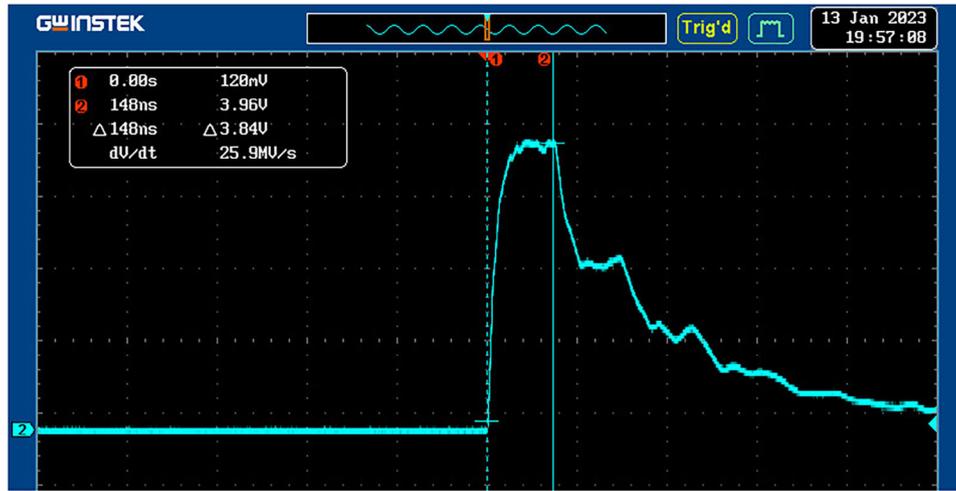


FIGURE 25 A zoomed rising edge of the signal measured at the input with the cables short-circuited showing the subtractive reflected pulse.

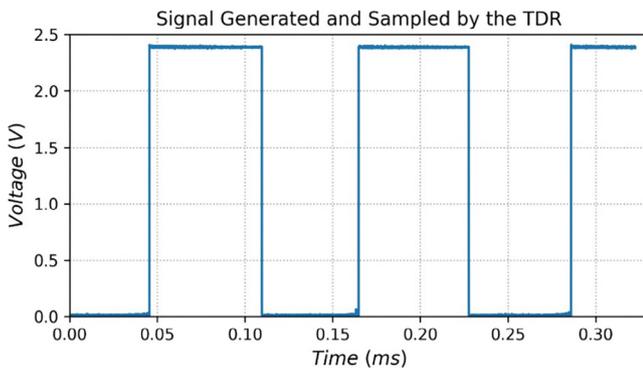


FIGURE 26 Signal generated and sampled by the Raspberry Pi time domain reflectometer system.

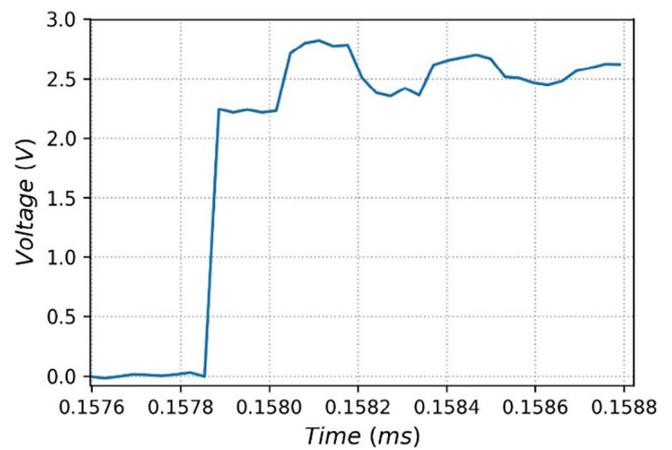


FIGURE 28 A zoomed section of the signal sampled at the input of the open-circuited test cables showing the additive reflected pulse.

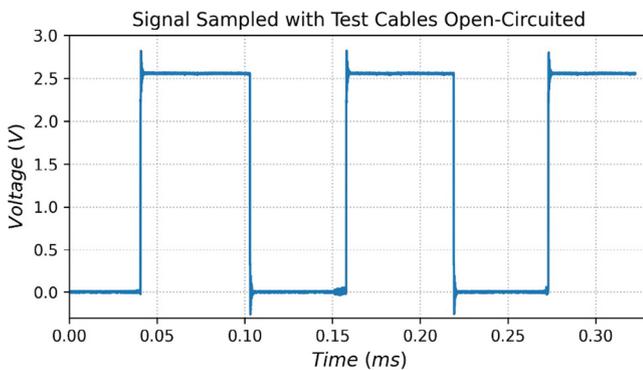


FIGURE 27 Signal sampled from the input of the open-circuited test cables.

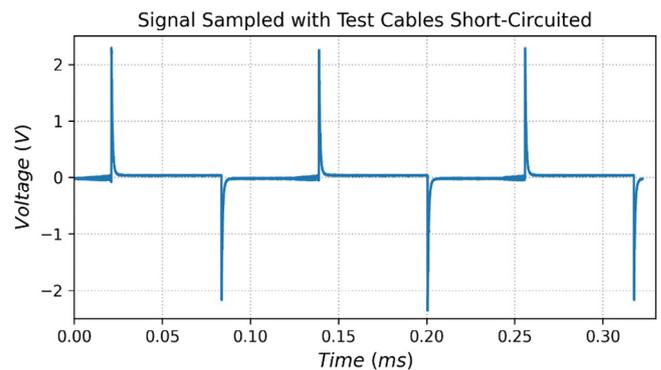


FIGURE 29 Signal sampled from the input of the short-circuited test cables.

Next, signal segments were generated using the obtained indices of the rising and falling edges of the incident signals. The step signal generated by the signal generator has a duty cycle of 51%. The size of a segment is given by Equation (12).

$$N = 0.49 \times \frac{1}{8.9 \times 10^3} \times 31 \times 10^6 \approx 1707 \text{ samples} \quad (12)$$

The discrete gradient of each segment was computed using Equation (7). The binary vector, R , was then computed from the discrete gradient using Equation (8). A threshold value of 0.3 was used. The value was determined from experimentation.

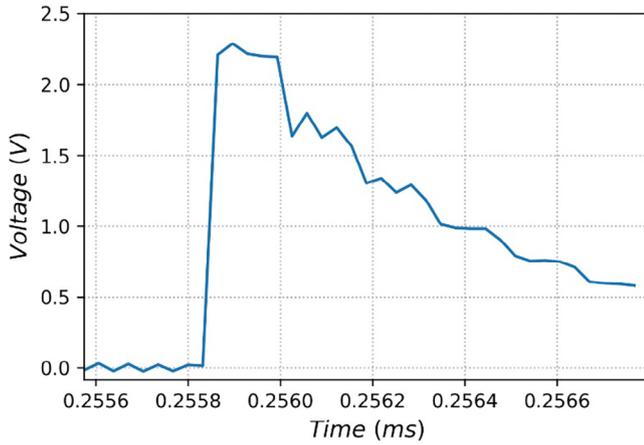


FIGURE 30 A zoomed section of the signal measured at the input of the short-circuited test cables showing the subtractive reflected pulse.

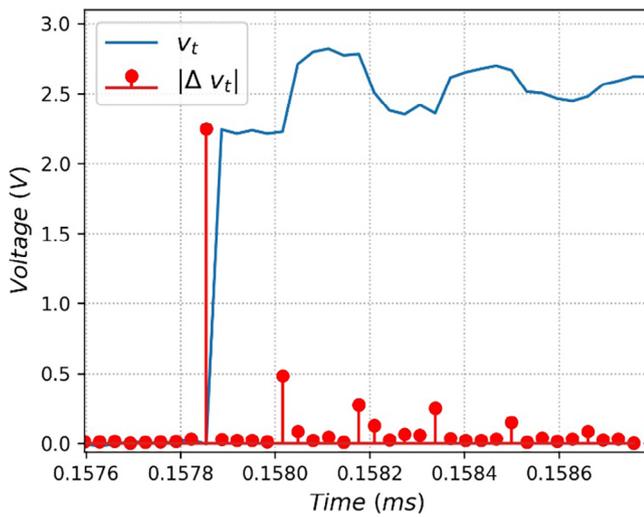


FIGURE 31 A plot of a sampled signal segment and its discrete gradient.

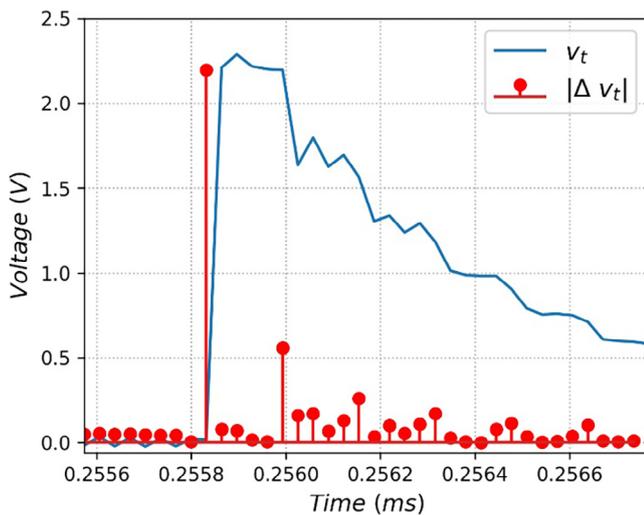


FIGURE 32 A plot of a sampled signal segment and its discrete gradient.

Figures 35 and 36 show the plots of signal segments, their discrete gradients, and the calculated binary vectors, R .

Figures 35 and 36 show signal segments that contain the edges of reflected signals from the open circuit and short circuit setups, respectively. The black stem plots with solid stems show the discrete gradients of the signal segments. The discrete gradients have the largest magnitudes at the edges of the first reflected pulses. Subsequent secondary reflections have smaller discrete gradients at their edges due to attenuation. The red stem plots with dotted stems show the binary vectors, R , computed from the discrete gradients of the signal segments. The vectors have values of 1s at the indices corresponding to the rising edge of the first reflected signals and 0s everywhere else. Figures 37 and 38 show the detected edges of both the incident and reflected signals.

Figures 37 and 38 show the time of the rising edges of the incident and reflected signals. The dash-dot lines show the time of the rising edges of the incident signals and the dotted lines show the time indices of the rising edges of the reflected signals. Using the pairs of the detected edges of the incident and reflected signals, the time delay was computed. The time delay of the reflected signal in both the open circuit and the short circuit setup was 161.3 ns. The distance to the fault is given by Equation (13).

$$d_f = \frac{0.67 \times 3 \times 10^8 \times 161.3 \times 10^{-9}}{2} = 16.21 \text{ m} \quad (13)$$

The value obtained from Equation (11) has an absolute error of 1.21 m.

The Raspberry Pi TDR can therefore be used to detect and locate faults in electrical cables. The distance to the fault obtained from the Raspberry Pi TDR has a larger absolute error than the one obtained from the oscilloscope since the Raspberry Pi TDR has a lower spatial resolution than the oscilloscope. This is attributed to the lower sampling rate of the Raspberry Pi TDR (31 MSPS) compared to that of the oscilloscope (2 GSPS).

Table 2 shows the results obtained from experiments conducted with the Raspberry Pi TDR, the oscilloscope and the copper cables.

Finally, the files collected from the experiments conducted on the electric fences were processed using the time domain algorithm to predict the distances to the simulated faults. Tables 3 and 4 show the results obtained from experiments conducted with the Raspberry Pi TDR and electric fences.

Tables 3 and 4 show the average predicted distances for the faults simulated at different points along the electric fences' cables at DeKUTWC and Ol Pejeta Conservancy. The values predicted by the time domain reflectometry algorithm are close to the expected values. The overall mean absolute error (MAE) of the time domain reflectometry algorithm on the data collected from the experiments with the electric fences is 1.52 m. This means that on average, the faults were predicted within ± 1.52 m of the actual distance. The error is very small compared to the distance the rangers have to walk while searching for faults. The predicted distance to the fault greatly narrows

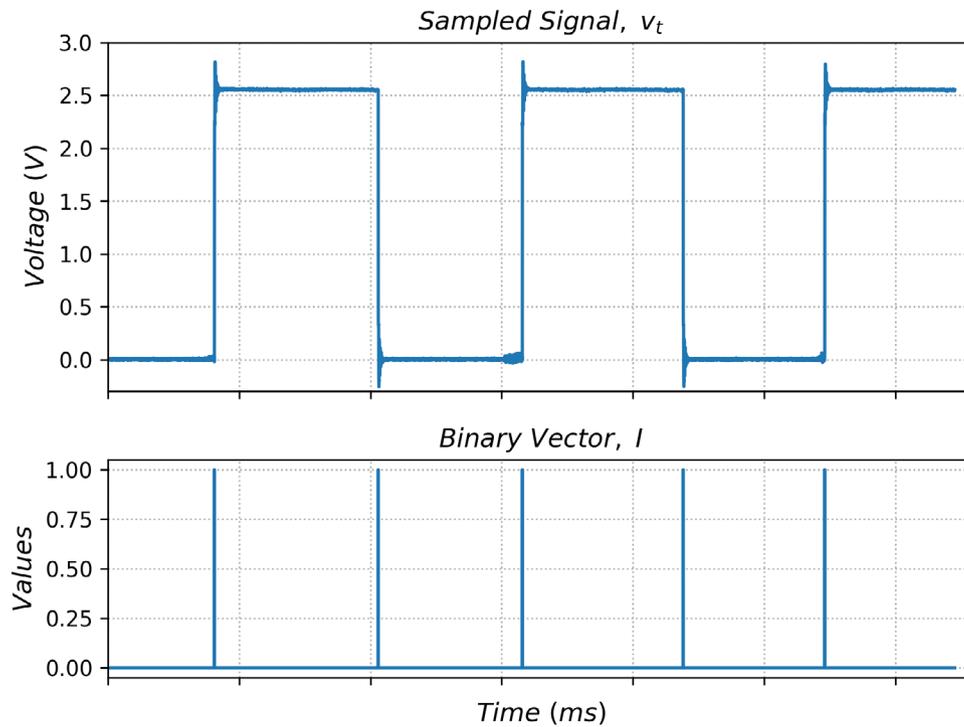


FIGURE 33 Plots of a signal sampled from the open-circuited cable setup and the binary vector computed from the discrete gradient of the signal.

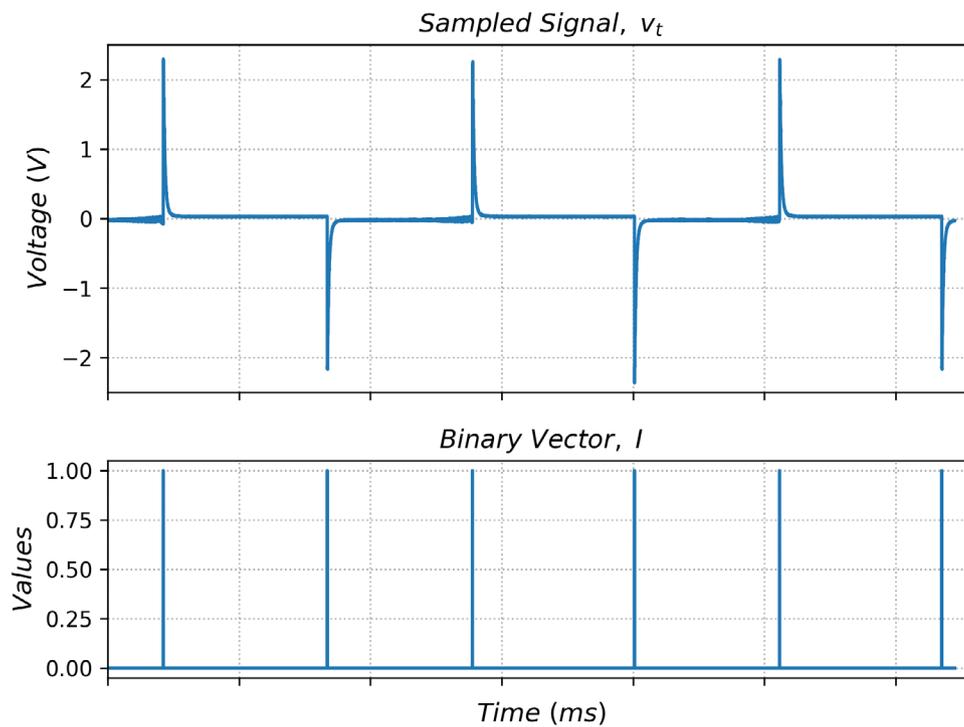


FIGURE 34 Plots of a signal sampled from the short-circuited cable setup and the binary vector computed from the discrete gradient of the signal.

down the section of the fence that the rangers need to look for the fault. The Raspberry Pi TDR can detect and locate faults in electric fences despite their nonuniform nature due to the multiple joints they have.

With the electric fences, the Raspberry TDR had a blind spot of about 12 m. A blind spot in time domain reflectometry is the shortest cable length below which a TDR cannot detect faults due to overlapping of the incident and reflected waves.

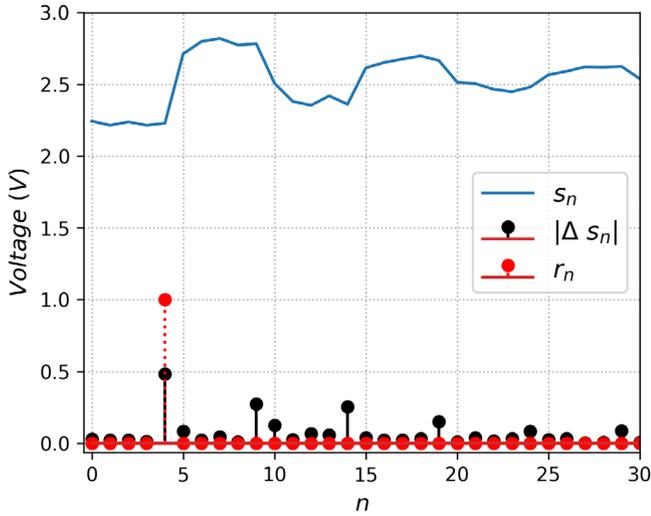


FIGURE 35 A plot of a signal segment containing change points due to interference by reflection caused by an open circuit. The dash-dot line is the finite difference of the signal segment, and the dotted line is the generated binary vector.

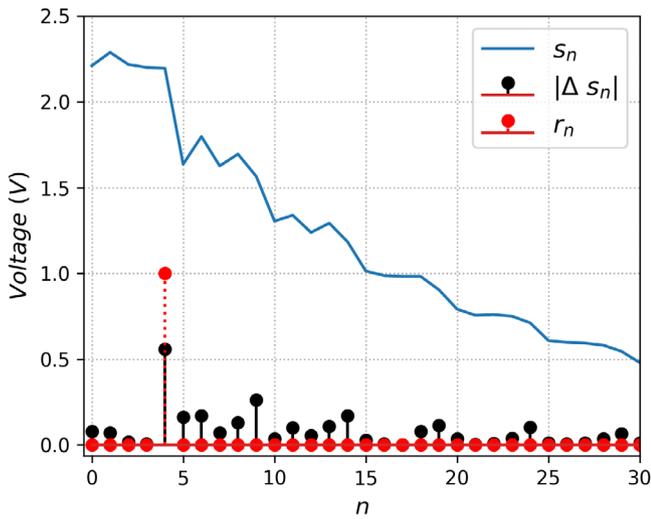


FIGURE 36 A plot of a signal segment containing change points due to interference by reflection caused by a short circuit. The dash-dot line is the finite difference of the signal segment, and the dotted line is the generated binary vector.

The problem of blind spots can be solved by using probes that are at least the length of the blind spot to connect a TDR to the CUT [42].

4 | WORK IN PROGRESS

The current version of the Raspberry Pi TDR is designed to test an electric fence that is not live. Electric fences operate at very high voltages (in the kilovolts range) that can damage the Raspberry Pi TDR. The user needs to switch off the energiser and then test the fence using the Raspberry Pi TDR. Future work will include designing a protection circuit to inter-

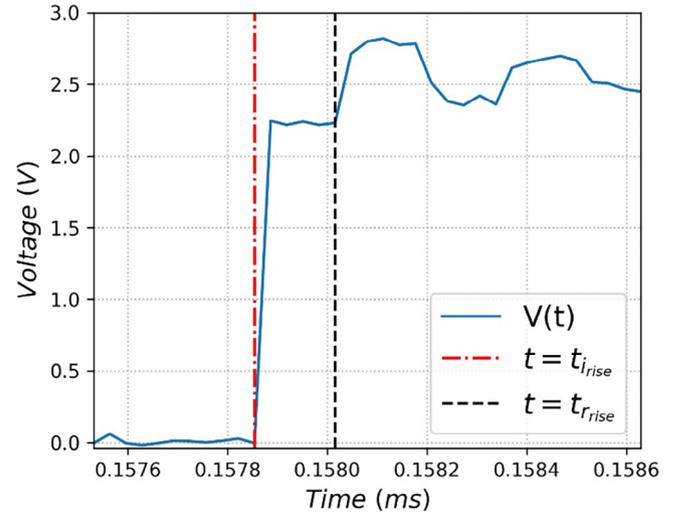


FIGURE 37 Detected time indices of the incident and reflected signals' edges in an open circuit.

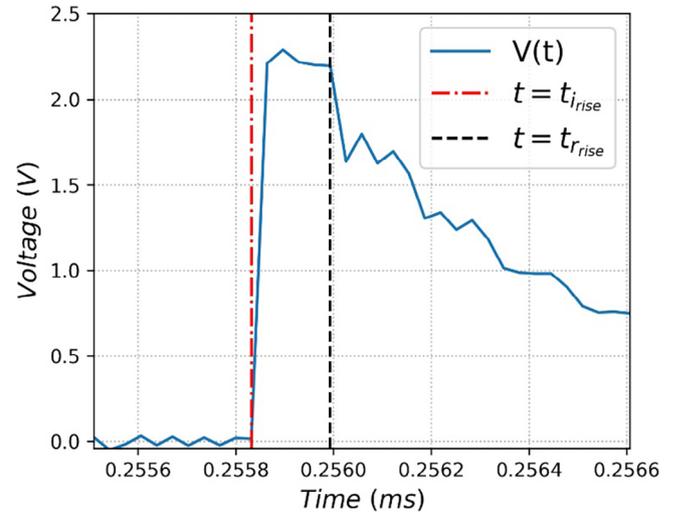


FIGURE 38 Detected time indices of the incident and reflected signals' edges in a short circuit.

TABLE 2 Results obtained from preliminary experiments with the Raspberry Pi time domain reflectometer and the oscilloscope.

Instrument	Fault	Distance to the fault (m)	Predicted distance (m)	Absolute error (m)
Oscilloscope	Open circuit	15	14.87	0.13
	Short circuit	15	14.87	0.13
Raspberry Pi TDR	Open circuit	15	16.21	1.21
	Short circuit	15	16.21	1.21

face the Raspberry Pi TDR with a live fence. The circuit will filter out the low-frequency high-voltage pulses generated by the energiser but allow the TDR signal to pass. The use of the high-voltage pulses of the energiser for arc reflection to detect high impedance faults in electric fences will also be explored.

TABLE 3 Results obtained from processing files collected from Dedan Kimathi University of Technology Wildlife Conservancy.

Type of fault	Actual distance to fault (m)	Number of files collected	Average predicted distance (m)	Mean absolute error (m)
Short circuit	12.41	21	14.30	1.89
Short circuit	17.44	21	18.60	1.16
Short circuit	22.38	21	25.53	1.51
Short circuit	26.86	21	28.41	1.55
Short circuit	32.24	21	33.20	0.96
Short circuit	37.61	20	38.71	1.10
Short circuit	42.15	21	43.50	1.35
Short circuit	47.35	25	48.81	1.47
Short circuit	52.11	21	53.77	1.66
Short circuit	57.08	22	58.58	1.50
Short circuit	62.05	22	63.24	1.19
Short circuit	66.93	21	67.49	0.56
Short circuit	72.10	20	74.02	1.92
Short circuit	77.00	21	79.20	2.20
Short circuit	82.03	21	84.05	2.02
Short circuit	87.52	21	88.71	1.19
Short circuit	93.26	21	94.58	1.32
Short circuit	98.56	21	99.41	0.85
Short circuit	104.26	22	105.97	1.71
Open circuit	106.10	30	108.81	2.71

The Raspberry Pi TDR can also be used to detect faults in other electrical cables as shown during the tests with the copper cables.

5 | CONCLUSION

This paper presents a low-cost Raspberry Pi-based TDR for fault detection and localisation in electric fences. Fault detection and localisation in electric fences are necessary to ensure their effectiveness. Conventional methods of fault detection and localisation are inefficient. The Raspberry Pi TDR uses the principle of time domain reflectometry to detect and locate open circuit and short circuit faults in electric fences. The system has been evaluated and it has successfully detected and located open circuit and short circuit faults in electric fences. From the experiments carried out with electric fences, the Raspberry Pi TDR was able to compute the distances to faults with an MAE of 1.52 m. The error is very small compared to the distance the rangers have to walk to locate faults in electric fences. The Raspberry Pi TDR has the potential to make the process of fault detection and localisation in electric fences more efficient, thus improving their effectiveness.

TABLE 4 Results obtained from processing files collected from Ol Pejeta Conservancy.

Type of fault	Actual distance to fault (m)	Number of files collected	Average predicted distance (m)	Mean absolute error (m)
Short circuit	12.20	10	14.16	1.96
Short circuit	36.60	10	35.41	1.19
Short circuit	67.10	13	65.92	1.18
Short circuit	97.60	14	96.73	0.87
Short circuit	128.10	11	126.64	1.46
Short circuit	158.60	12	156.79	1.81
Short circuit	189.10	12	188.30	0.80
Short circuit	219.60	15	217.79	1.81
Short circuit	250.10	13	249.46	0.64
Short circuit	280.60	20	278.51	2.09
Open circuit	280.60	17	278.56	2.04

AUTHOR CONTRIBUTIONS

Gabriel Kiarie: Conceptualisation; investigation; methodology; software; writing—original draft preparation. **Ciira wa Maina:** Supervision; funding acquisition; writing—review and editing. **Kumbirayi Nyachionjeka:** Supervision; writing—review and editing.

ACKNOWLEDGEMENTS

This work was supported by Canada's International Development Research Centre (IDRC) through the Artificial Intelligence for Development in Africa (AI4D Africa) programme and Data Science Africa (DSA) through the Affiliated Centres programme. The authors wish to thank Anthony Munuve, Dishan Otieno, and the Rangers at DeKUTWC and Ol Pejeta Conservancy for their assistance during the development and testing of the Raspberry Pi TDR.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

The data used for this research is found in the repository: <https://github.com/DeKUT-DSAIL/electric-fence-fault-detector-and-localiser>

ORCID

Gabriel Kiarie  <https://orcid.org/0000-0003-1413-7270>

REFERENCES

- Feuerbacher, A., Lippert, C., Kuenzang, J., Subedi, K.: Low-cost electric fencing for peaceful coexistence: An analysis of human-wildlife conflict mitigation strategies in smallholder agriculture. *Biol. Conserv.* 255, 108919 (2021)
- Köpke, S., et al.: Human–elephant conflict in Sri Lanka: A critical review of causal explanations. *Sustainability* 13(15), 8625 (2021)
- López, J.J., Mulero-Pázmány, M.: Drones for conservation in protected areas: Present and future. *Drones* 3(1), 10 (2019)

4. Acreman, M., Hughes, K.A., Arthington, A.H., Tickner, D., Dueñas, M.-A.: Protected areas and freshwater biodiversity: A novel systematic review distills eight lessons for effective conservation. *Conserv. Lett.* 13(1), e12684 (2020)
5. Mousumi, G.-H., et al.: Protected areas and biodiversity conservation in India. *Biol. Conserv.* 237, 114–124 (2019)
6. Nguyen, M.H., Jones, T.E.: Building eco-surplus culture among urban residents as a novel strategy to improve finance for conservation in protected areas. *Humanit. Soc. Sci. Commun.* 9(1), 1–15 (2022)
7. Rajalashmi, K., Hemachandira, V.S., Saravanan, S., Chandru, M., Kaviyadevi, R.S.: Monitoring & Tracking System for Elephants Using GPS/GSM with Smart Electric Fencing. In: *IOP Conference Series: Materials Science and Engineering*, vol. 1084, p. 012067 (2021)
8. Al-Bahadly, I., Simpson, R.: Contactless electric fence fault detection system. In: *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Auckland, New Zealand (2019)
9. Fault finding on your electric fence. Electric Fence. <https://www.electrifice-online.co.uk/blogs/blogs/electric-fence-fault-finding/>
10. Palmer, M., Thacker, E., Cromwell, S., Heaton, K., Carter, K.: High Tensile Permanent Electric Fence, Electrifying the Fence, Utah State University Extension, Utah (2020)
11. Dissanayake, S., Fernando, D., Suduwella, C., Dabare, M., De Zoysa, K., Keppetiyagama, C.: Sensor based fence breakage detection system. In: *2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Colombo, Sri Lanka (2018)
12. Jayasuriya, N., et al.: 'Wire is not dead': Wired-backscatter communication for breakage detection in electric fences. In: *International Conference on Embedded Wireless Systems and Networks (EWSN)*, Uppsala, Sweden (2017)
13. Chandrasiri, K.A.D.I., Sagara, H.V.S.S., Ilandara, U.A.I.B., Sanjeewa, S.D.A.: Designing and development of smart electric fence fault finding system. In: *International Research Symposium 2022*, Dehiwala-Mount Lavinia, Sri Lanka (2022)
14. Haddad, D., Kallel, A.Y., Amara, N.E.B., Kanoun, O.: Multiple faults detection and location in bus-shaped cable networks by distributed time-domain reflectometry. *IEEE Sens. Lett.* 6(5), 1–4 (2022)
15. Saleh, M.U., et al.: An overview of spread spectrum time domain reflectometry responses to photovoltaic faults. *IEEE J. Photovolt.* 10(3), 844–851 (2020)
16. Jayakumar, N.K.T., et al.: Postprocessing for improved accuracy and resolution of spread spectrum time-domain reflectometry. *IEEE Sens. Lett.* 3(6), 1–4 (2019)
17. Sommervogel, L.: Various models for faults in transmission lines and their detection using time domain reflectometry. *Prog. Electromagn. Res.* 103, 123–135 (2020)
18. Auzanneau, F.: Multiple soft defect signature magnification in electrical cables with binary time-domain reflectometry. *IEEE Sens. Lett.* 4(7), 1–4 (2020)
19. TDR500/3. Megger. <https://megger.com/handheld-tdr-tdr500/3>
20. Megger TDR500/3 cable fault locator. Test-meter. <https://www.test-meter.co.uk/megger-tdr500-3-cable-fault-locator>
21. Xu, Z., Mauldin, T., Yao, Z., Hefferman, G., Wei, T.: A high-performance, reconfigurable, fully integrated time-domain reflectometry architecture using digital I/Os. *IEEE Trans. Instrum. Meas.* 70, 1–9 (2021)
22. Sasaki, Y., Zhao, Y., Kuwana, A., Kobayashi, H.: Highly efficient waveform acquisition condition in equivalent-time sampling system. In: *2018 IEEE 27th Asian Test Symposium (ATS)*, Hefei, China (2018)
23. Svatos, J., Fischer, J., Holub, J.: High-speed equivalent-time sampling virtual instrument based on microcontroller ADC. *Measurement* 220, 113392 (2023)
24. Zang, X., Zhao, J., Lu, Y., He, Q.: Precision measurement system of high-frequency signal based on equivalent-time sampling. *Electronics* 11(13), 2098 (2022)
25. Zhijian, D., Qi, W., Xi, Y., Yijiu, Z.: The application of random equivalent sampling in acquisition system with 5Gsp/s real-time sampling. In: *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, Yangzhou, China (2017)
26. Ballal, T., Al-Naffouri, T.Y.: Low-sampling-rate ultra-wideband channel estimation using equivalent-time sampling. *IEEE Trans. Signal Process.* 62(18), 4882–4895 (2014)
27. Lee, D., Sung, J., Park, J.: A 16ps-resolution random equivalent sampling circuit for TDR utilizing a Vernier time delay generation. In: *2003 IEEE Nuclear Science Symposium. Conference Record (IEEE Cat. No. 03CH37515)*, Portland, OR, USA (2003)
28. Sulthoni, M.A., Rizqulloh, M.A.: A low cost microcontroller-based time domain reflectometer for soil moisture measurement. In: *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, Bandung, Indonesia (2019)
29. Negrea, C., Rangu, M.: Sequential sampling time domain reflectometer. In: *2009 15th International Symposium for Design and Technology of Electronics Packages (SIITME)*, Gyula, Hungary (2009)
30. Trebbels, D., Kern, A., Fellhauer, F., Huebner, C., Zengerle, R.: Miniaturized FPGA-based high-resolution time-domain reflectometer. *IEEE Trans. Instrum. Meas.* 62(7), 2101–2113 (2013)
31. Svatos, J., Fischer, J., Holub, J.: High-speed equivalent-time sampling virtual instrument based on microcontroller ADC. *Measurement* 220, 113392 (2023)
32. SN74AC14N. Texas Instruments. <https://www.ti.com/product/SN74AC14N>
33. AD9226. Analog Devices. <https://www.analog.com/en/products/ad9226.html#product-overview>
34. Bentham, J.P.: Raspberry Pi Secondary Memory Interface (SMI). Lean2. <https://iosoft.blog/2020/07/16/raspberry-pi-smi/>
35. Bentham, J.P.: Raspberry Pi DMA programming in C. Lean2. <https://iosoft.blog/2020/05/25/raspberry-pi-dma-programming/>
36. Bentham, J.P.: Fast data capture with the Raspberry Pi. Lean2. <https://iosoft.blog/2020/06/11/fast-data-capture-raspberry-pi/>
37. Signal edge detection. Scilab. <https://www.scilab.org/signal-edge-detection>
38. Dutta Roy, S.C.: Lossy transmission lines revisited. *IETE J. Edu.* 64(1), 15–16 (2023)
39. The characteristic impedance of lossless and lossy transmission lines. Cadence. <https://resources.system-analysis.cadence.com/blog/msa2021-the-characteristic-impedance-of-lossless-and-lossy-transmission-lines>
40. Steer, M.: The lossy terminated line. LibreTexts Engineering (2022). [https://eng.libretexts.org/Bookshelves/Electrical_Engineering/Electronics/Microwave_and_RF_Design_II_-_Transmission_Lines_\(Steer\)/02%3A_Transmission_Lines/2.05%3A_The_Lossy_Terminated_Line](https://eng.libretexts.org/Bookshelves/Electrical_Engineering/Electronics/Microwave_and_RF_Design_II_-_Transmission_Lines_(Steer)/02%3A_Transmission_Lines/2.05%3A_The_Lossy_Terminated_Line)
41. Velocity factor general table. Ground Rules. <https://www.qsl.net/ac5jw/isbp/vf-gen.html>
42. Lee, C.-K., Chang, S.J.: A method of fault localization within the blind spot using the hybridization between TDR and wavelet transform. *IEEE Sens. J.* 21(4), 5102–5110 (2020)

How to cite this article: Kiarie, G., wa Maina, C., Nyachionjeka, K.: A low-cost Raspberry Pi based time domain reflectometer for fault detection in electric fences. *IET Sci. Meas. Technol.* 1–18 (2024). <https://doi.org/10.1049/smt2.12183>