USDOT Tier 1
University Transportation Center
on Improving Rail Transportation
Infrastructure Sustainability and Durability

Final Report UNLV-7

FAST RAIL TRANSIT DEGRADATION MONITORING AND FAILURE PREDICTION OF CARBON STRIP IN PANTOGRAPH

By

Robert A. Schill, Jr., PhD, Professor/Director of EMITION
Energy Materials Interation Technology Initiative of Nevada Center (EMITION)
Department of Electrical and Computer Engineering
University of Nevada – Las Vegas
Robert.Schill@unlv.edu

June 10, 2021

Grant Number: 69A3551747132

DISCLAIMER

ABSTRACT

The weakest link in powering high-speed rail locomotives is the carbon strip of a pantograph collector which makes physical contact between the overhead power line and the electrical supply wires of the locomotive. This research effort aims to detect and distinguish the degradation effects of the carbon strip by monitoring the locomotive's input current line. A pulsed-power carbon strip degradation test stand was designed, built, and tested based on static degradations. Using pulsed power techniques, peak currents attained are on the order of maximum AC/DC currents drawn by the locomotive. An extensive circuit theory (implemented with MATLAB), an independent computer aided design, analysis, and circuit simulation tool (LTSpice), and an in-line current measuring resistor were used to study the charge and discharge properties of the test stand and to corroborate experimental results using the EM dot. The EM dot is a UNLV patented time varying electric and magnetic field sensor. All four studies showed reasonably good agreement. The EM dots were sensitive enough to distinguish the resistance of six different grooves for train loads between 1.1 (and less; not tested) Ω and 2.2 Ω. For a coarser carbon strip thickness distribution, the EM dots can resolve carbon strip thicknesses for train loads between 2.2 Ω and 5.0 Ω. The carbon strip resistance ranged between $3m\Omega \leq R_c \leq 28m\Omega$ where the short state occurs at R_c=3 mΩ (measured) and the new, unblemished, carbon strip state occurs at R_c=28 mΩ (measured). The generation of carbon thickness curves, carbon resistor curves, wear rates, and carbon states in-transit are discussed in detail with an example. Further, it is discussed how to use the curves to monitor the carbon strip in-transit and to predict its end-of-life time or remaining train distance in-transit.

Keywords: Materials, electromagnetic pulses, electric trains, railroad electrification

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The weakest link in powering high-speed rail locomotives is the carbon strip of a pantograph collector which makes physical contact between the overhead power line and the electrical supply wires of the locomotive. This research effort aims to detect and distinguish between degradation effects of the carbon strip by monitoring the locomotive's input current line. At the onset of this research effort, there was no in-transit means of monitoring the carbon strip's integrity and wear at contact.

It was hypothesized that the transient nature of the current flow in the locomotive's input power line and hence the magnetic field it generates can be used to characterize and monitor the carbon strip's integrity and wear rate in-transit. This information allows one to predict the lifetime of the strip. Further, it was hypothesized that the UNLV patented electromagnetic dot (EM dot) sensor is sensitive enough to monitor the magnetic field and changes in the magnetic field generated by the current in the locomotive's input power lines.

A pulsed-power carbon strip degradation test stand was designed, built, and tested based on static degradations. The test stand was designed to operate in single pulse mode and in multi-pulsed mode. Peak currents generated in the test stand were on the order of the maximum AC/DC currents drawn by a locomotive. Depending on the train load, the test stand can supply up to roughly 1160 A peak with an average current of 737 A in a 10 microsecond pulse width with a typical repetition period of 135 ms.

Driven by a 5 kV source set at 2.5 kV, a 3.3 Ω F capacitor (rated at 25 kV DC) releases its energy to an approximate 0.1 to 15 Ohm liquid copper sulfate ($CuSO_4$) resistor by way of a discharge tube switch (in self-breaking mode) in series with a carbon strip. The liquid resistor represents the train load or train load with wire and wire contact resistances. Because of stability issues especially for accuracy and verification purposes, the copper sulfate liquid coaxial resistor and the in-line manganin rod were replaced with pulsed-power solid-state resistors.

Operating in multi-pulsed mode, the pulsed-power resistors (resistances between 1 Ω and 10 Ω and and between 10 mΩ and 30 mΩ) can exceed temperatures ranging from 500 ^{o}F to 1000 ^{o}F. At 200 ^{o}F, the pulsed-power resistors have a 11% (worst case scenario) change or less from their room temperature resistance. To preserve the resistors' integrity, experiments are conducted in five second intervals with an off time of 200 s to allow for cooling. Experimental studies show a less than 10% (worst case) maximum deviation and a less than 6% (worst case) minimum deviation in the pulsed-power resistor's resistance may result. This is acceptable.

Based on Paschen effects, the self-breaking switch exhibits closure at about 2 kV (measured more like 1.9 kV) at tube pressures between 5 and 6 Torr. An extensive circuit theory (with MATLAB), an independent computer aided design, analysis, and circuit simulation tool (LTSpice), and an in-line current measuring resistor were used to study the charge and discharge properties of the test stand and to corroborate experimental results as measured with the EM dot. All four studies showed reasonably good agreement.

The EM dots were sensitive enough to distinguish the resistance of six different grooves for train loads between 1.1 (and less; not tested) Ω and 2.2 Ω. For a coarser carbon strip thickness distribution, the EM dots can resolve carbon strip thicknesses for train loads between 2.2 Ω and 5.0 Ω. The carbon strip resistance ranged between $3m\Omega \leq R_c \leq 28m\Omega$ where the short state occurs at R_c=3 mΩ and the new, unblemished state occurs at R_c =28 mΩ.

The generation of carbon thickness curves, carbon resistor curves, wear rates, and carbon states in-transit are discussed in detail. Two techniques are presented on how to generate the characteristic curves of the carbon strip. The first technique uses the fast rail in-transit for a particular set of train parameters. Discontinuous changes in state and varying carbon strip wear rates are allowed. The second technique uses a discrete, static-generated carbon thickness curve with interpolation developed in a laboratory setting based on the assumption that the carbon strip has a continuous state with varying carbon strip wear rates (varying slopes). Statistical deviations from baseline characteristic curves (monitoring) and end-of-life predictions are addressed.

An EM dot sensor is not sensitive to steady-state, low frequency AC (60 Hz) and DC fields. To apply the pulse power techniques to a DC or AC supply line, one would have to couple a high voltage pulse to the electrical line. To minimize dispersion and attenuation losses, the overhead power line supplying energy to the train can also supply energy to pulse generating equipment housed in the train. Consequently, one can control both the pulse strength, its shape, and on/off states.   The probing pulse would only need to propagate the length of the train in contrast to that propagating from a power station to the train. The nature of most pulses is frequency rich. The high frequencies of the pulse will drive the EM dot. The pulse durations can be short and the pulse period relatively long. Other options of adapting a probing signal signature into a DC and an AC sourced train are also examined.

It is recommended that different stages of this research be tested on a train.

THE PROBLEM

Typically, high-speed rail electric locomotives employ pantographs with carbon strips as current collecting systems to maintain electrical contact with overhead catenary power lines even when the lines are not perfectly centered with the vehicle. The pantograph's carbon strip is long which helps to dissipate heat and distribute wear. The weakest link in powering high-speed rail locomotives is the carbon strip of a pantograph collector which makes physical contact between the overhead power line and the electrical supply wires of the locomotive (Yang, et al., 2017). Contact quality between the catenary and the pantograph dictates the current collection quality which in turn affects the locomotive speed. An excessive force promotes wear and fatigue of the contact wire and pantograph collector (carbon strip). An insufficient contact force can lead to arcing and an interruption of electric power (Yang. et al., 2017, Xin, et al., 2020). Premature wear of the carbon strip is a consequence of arc ablation, mechanical friction, thermal shock, rain, repeated alternating hot and cold conditions (electric locomotives moving through changeable weather), and environmental erosion (Yang. et al., 2017, Xin, et al., 2020, Wu, et al. 2018, and Wei, et al., 2019). These effects lead to macroscopic (Wu, et al., 2018, and Wei, et al., 2019) (large cracks, bulk loss of material, etc.) and microscopic (wear debris, exfoliation, ploughing groove, cracks, flake layer, particles, massive melts, thermal stress cracking, and pitting) damage to the carbon strips.

To prevent undesired in-transit breakdown, frequent carbon strip inspections are performed on high-speed rail locomotives (Xin, et al., 2020, Yang, et al., 2018, Becker, et al., 1996, Jarek, 2019, and Judek and Jarzebowicz, 2016). In particular, railway overhaul rules in China require the pantograph slide to be inspected periodically (Yang, et al., 2018). This is performed normally with manpower (Yang, et al., 2018). But, this traditional technique can only inspect for abnormal wear on the macroscopic level. More inspections or possibly in-transit inspections are possible with a camera in two and three dimension (Yang, et al., 2018, Jarek, 2019, Judek and Jarzebowicz, 2016, and Bruni, et al., 2018). In general, these pantograph inspection techniques can only detect major faults (Xin, et al. 2020).

It is difficult to gain visual access to the contact area of the catenary wire and strip as a whole for inspection. It is more difficult to inspect dynamic degradation in-transit. As the vehicles burn through their carbon strips, the material of the pantograph rubs on the overhead line potentially resulting in-transit delays, pole wire derailments, and downed overhead lines (Becker, et al., 1996, Jarek, 2019). Presently, there are no "real time" techniques that can detect wear on carbon strips (Jarek, 2019). Currently, JR Dynamics (Jarek, 2019) is developing a low cost carbon strip with wear monitoring system to address this need. Others are examining 3D machine vision sensors for evaluating the carbon strip (Yang, et al., 2018). As indicated above, researchers and developers have taken great pains to monitor and study the characteristics of the carbon strip. A unique approach to this problem in real time was studied in this effort.

This unique effort examines the degradation effects of the carbon strip by monitoring the locomotive's input current line. It was hypothesized that the transient distribution of the current flow around the locomotive's input power line provides a unique signature that can be mapped back to the transient real time degradation properties of the carbon strip of the pantograph allowing one to predict the lifetime of the strip in real time. It was proposed that the current distribution be monitored with existing and next generation UNLV patented electromagnetic dots (EM dots). The

EM dot (Agry, and Schill, 2014a), (Agry, and Schill, 2014b) is a UNLV patented sensor that measures the electric and magnetic fields at a single macroscopically small point in space simultaneously in time. The existing device configuration is based on a seamless, semi-rigid, coaxial configuration. The foundation of the work presented in this document hinges on these hypotheses.

UNLV PULSED-POWER CARBON STRIP DEGRADATION TEST STAND

A pulsed-power carbon strip degradation test stand was developed based on pulsed-power technology. Refer to Figure 1 for an overview of the test stand. Individual features of the test stand can be observed in Figures 2a-g which includes the EM dot sensor, carbon strip, and an in-line current verification monitor. Pulsed power technology was employed so that the test stand can drive average and/or peak currents in locomotive loads in a university lab setting. Further, undesired transient signatures such as switching noise from mechanical relays can be generated and studied. The test stand can operate in a single pulse mode and a multiple pulse mode. An approximate 10 μs pulse (pulse duration) is generated every 135 ms (approximate repetition period). A 1 kΩ limiting $CuSO_4$ liquid resistor was used in the charging process and as a means to isolate the DC power supply from the discharge operation.

For locomotive loads, R_L (pulsed-power, solid-state resistors), of 1.1 Ω, 2.2 Ω, 5 Ω, and 7.2 Ω, the test stand can supply peak (obtained from experiment) and average (calculation) currents of, respectively, 1160 A and 737 A, 1020 A and 645 A, 429 A and 271 A, and 310 A and 196 A. The peak current is $I_o = I_{Paschen} = V_{Paschen}/(R_c + R_L + R_w + R_M) \approx V_{Paschen}/(R_L)$ where the approximation is valid if $R_L >> R_c + R_w + R_M$. $V_{Paschen}$ is the Paschen voltage. It is the breakdown voltage in a gaseous medium such as the discharge switching tube at a particular gaseous pressure-distance of separation between the electrodes. The average current is the current averaged over a discharge time constant $\tau_d = C_b(R_c + R_L + R_w + R_M)$ as given by

$$I_{ave} = \frac{\int_0^{\tau_d} I_o e^{-t/\tau_d} dt}{\int_0^{\tau_d} dt} = [1 - e^{-1}]I_o = 0.632 I_o$$

where I_o is the peak current.

A 5 kV DC power supply set at 2.5 kV in constant voltage mode sources the test stand. A set of Ross relays are activated to place the test stand in single pulse mode or multi-pulse mode. In either single pulsed mode or the first couple of pulses in the multi-pulse mode, noise generated by the mechanical Ross relays significantly affects the voltage/current signatures of the test stand. After a few pulses in multi-pulse mode, the signal is relay noise free. The DC power supply charges a 3.3 μF capacitor bank using a 1 kΩ current limiting resistor.

The 1 kΩ current limiting resistor is composed of two coaxial copper rings immersed in a diluted copper sulfate ($CuSO_4$) solution. As the 3.3 μF capacitor is being charged, the voltage drop across the discharge tube electrodes increases. The internal discharge tube environment is about 6 Torr of Nevada air in the lab. At about 2 kV (more exact 1.9 kV), Paschen effects ensue. The tube environment between the electrodes breaks down discharging the capacitor bank. A high current 10 μs pulse is generated. The capacitor is discharged through a copper sulfate liquid coaxial resistor or appropriate pulsed-power solid-state resistors modeling the locomotive's resistive electrical load, R_L (1 to 10 Ω for liquid resistor; 1.1 to 7.2 Ω for pulse-power solid-state resistors). The carbon strip resistor, R_c (3 mΩ<R_c<28 mΩ), the in-line ammeter resistor (manganin rod resistor or solid-state resistors) R_M = 10 mΩ, the resistive effects of the circuit wires and connectors in the test stand R_W (~0.14 Ω), and the discharge tube are connected in series with the resistor representing the locomotive's load resistance R_L. Let $\tilde{R}_L = R_L + R_w + R_m \approx R_L$ be defined as the equivalent (effective) lumped train load resistance. The equivalent lumped train load resistance $\tilde{R}_L$ is approximately equal to the train load resistance R_L. The manganin rod is used as an in-circuit current monitor. Manganin is a precision resistance alloy with long term resistance stability. An LCR meter is used to measure all resistances especially those less than a couple of Ohms. Further, the LCR meter is used to measure the inductance of the overall circuit with discharge tube shorted. The resistance of the diluted $CuSO_4$ resistors are difficult to measure due to the polar property of the water molecule. In the pulsed-power carbon strip degradation test stand, both the resistance of the $CuSO_4$ resistor, R_L, and the resistance of the manganin rod R_M are unstable for accuracy and verification purposes. The $CuSO_4$ resistors were designed to dissipate the potentially large Joule heating loads generated by the test stand and to provide timely guidance in the experimental study. When using the manganin rod as an in-line means to directly measure the carbon strip current, the currents experimentally measured were approximately 3 to 5 times larger than predicted. These two unstable elements provided unrealistic results when compared to theoretical and computer modeling studies and estimates. Consequently, these elements were replaced with select, pulsed-power, UNLV tested, solid-state resistors. Typically pulsed-power has a large parameter space that may or may not be conducive to the experimental study. Consequently, all tests and characterizations on the resistors were performed in the pulsed-power carbon strip degradation test stand. In multi-pulsed mode, the pulsed-power solid-state resistors tend to become significantly hot on the order of seconds to tens of seconds. This affects the stability of the resistance relative to the room temperature resistance. Between room temperature and about 200° F, the pulsed-power resistors deviate by about 11% (worst case) from their original room temperature value. Internal temperatures as high as 500 °F and in one case over a 1000 °F have been measured at the expense of the stability of the resistor's resistance. As the resistors cool to room temperature, the resistor's resistance approaches the orginal room temperature resistance value prior to the heated state. More will be presented on these resistor tests in Section V: Experimental Results. Consequently, all experiments are performed while the resistors' temperatures are below 200 °F. It will be shown that experiments performed in the pulsed-power carbon strip degradation test stand within five second intervals satisfy the 200° F upper limit constraint in temperature. This assumes that the initial temperature of the resistor is room temperature.

TABLE 1 Groove Labels, Dimensions, and Average Resistances

Refer to Figure 2g for a picture of the commercial carbon strip attached to a metal holder. Refer to Figure 3 for the geometry of a typical groove.

Groove	**Measured Width (mm)**	**Measured Groove Depth H (mm)**	**D Carbon Thickness = Shortest Distance of Separation between electrodes, (Refer to Figure 3) (mm)**	**Measured Resistance (mΩ)**
Short	N/A	21	0	3
Extra Deep	11.63	8.02	12.98	4.5
Very Deep	9.63	6.34	14.66	6.8
Just Deep	8.31	4.44	16.56	11.3
Shallow	2.04	1.29	19.71	16.5
Normal (New)	N/A	0	21	28 (average)

The dot housing containing two EM dot sensors is arbitrarily positioned on the ground line. Only one dot is required for measurements. The second dot is available in case the first one becomes damaged. Further, it finds application in EM dot data verification studies. Ideally, the magnetic field circulates around a straight current carrying wire in a circular geometry with wire on axis. The dots are significantly activated when the field lines pass normal through the sensor head area bounded by the center wire of the dot. Two oscilloscopes are used to record data. One oscilloscope (Tektronix Oscilloscope TDS 6604B [6 GHz Bandwidth, 20 G Samples/s]) records the EM dot data. Two channels per sensor are required. Two channels are used on the second scope (Tektronix Oscilloscope TDS 784C [1 GHz Bandwidth, 4G Samples/s]). The first channel on the second scope is used intermittently to monitor the discharge voltage of the plasma discharge tube using a high voltage probe. Because the high voltage probe seems to undesirable load down the test stand it is not used when taking final sets of data. Here, the discharge tube voltage at discharge is the Paschen voltage. For all experiments the Paschen voltage, $V_{Paschen}$~ 2 kV. The second channel is used to monitor the current passing through the in-line ammeter resistor.

Figure 1 displays a commercial carbon strip compliments of Schunk Inc. type: SK1514-SK85 copper clamp. A 0.25” diameter rod is placed firmly and uniformly on the carbon strip oriented 90^o with respect to the length of the strip. As shown in Figure 2d, 30 pounds of force is applied to the rod in order that good contact is made between the carbon strip and rod. Less weight force on the rod resulted in poor electrical contact. More weight force did not appear to significantly influence static contact. The carbon strip is divided up into six sections each containing a groove of various depth. Table 1 lists the groove description, the groove geometry, and the average groove resistance. Figure 2g provides a top view and side view of the carbon strip with labels. All surface blemishes excluding the groove blemishes without significant changes in thickness seem to generate the same signal signature as the original carbon strip void all blemishes. As a result, significant studies on these blemishes were not pursued.

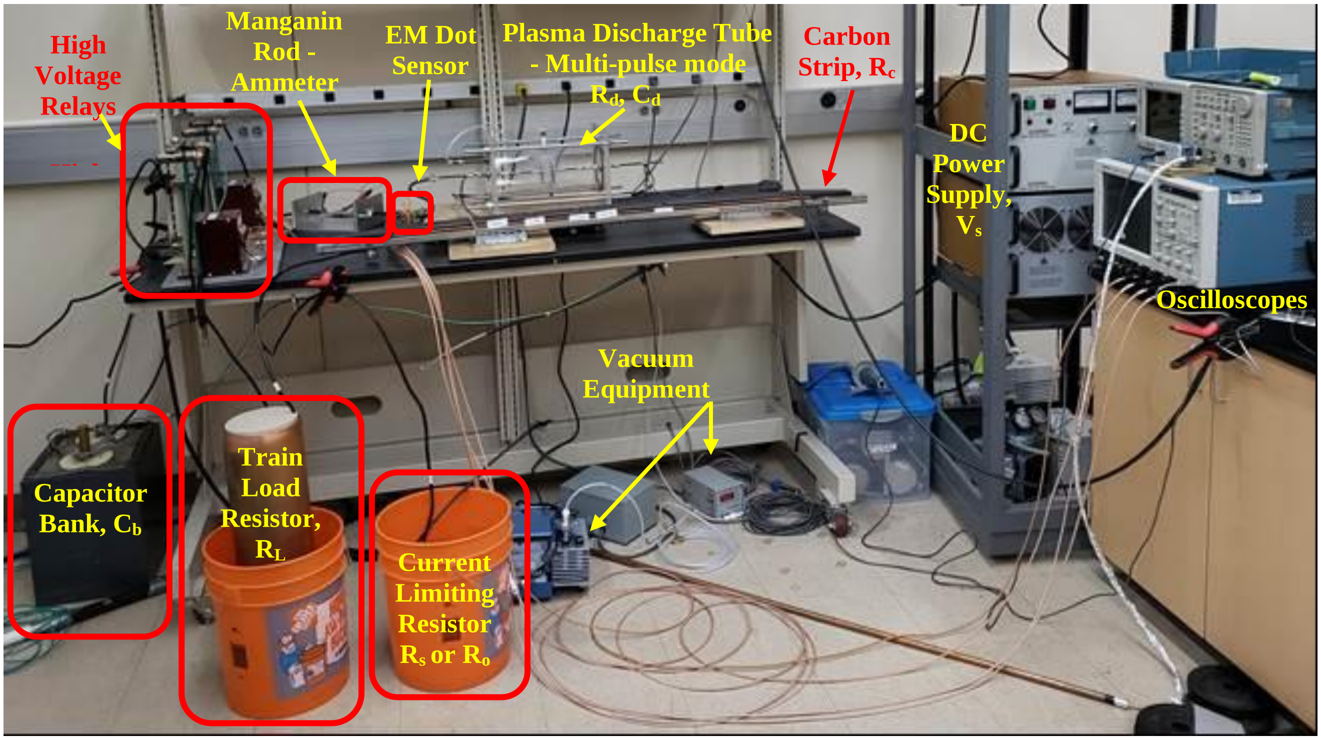
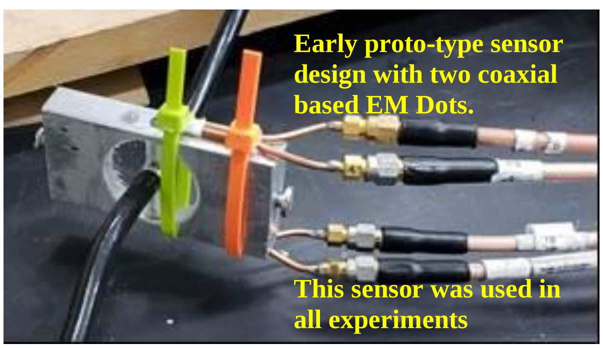
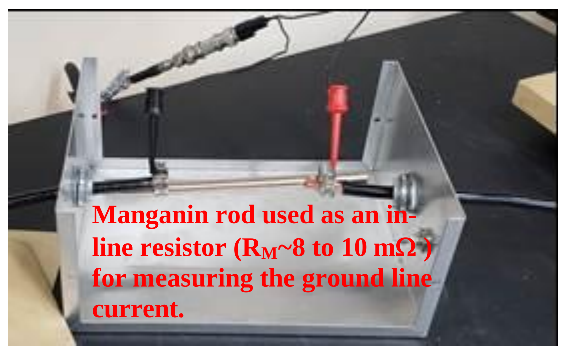


FIGURE 1 An overview of the high-voltage, high current, single and multi-pulsed power carbon strip degradation test stand with static faults in the carbon strip.
Isolated parts of the test stand are provided in Figures 2a-g.

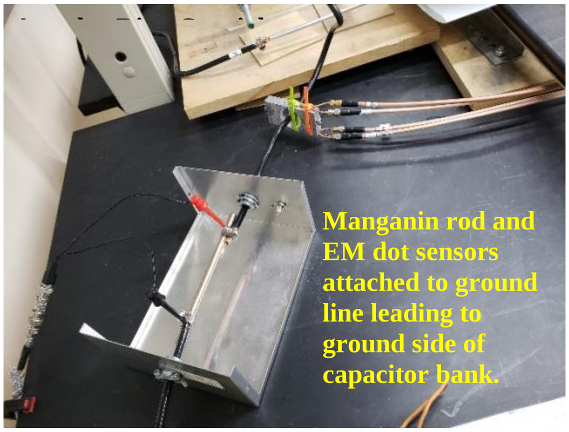


(a)

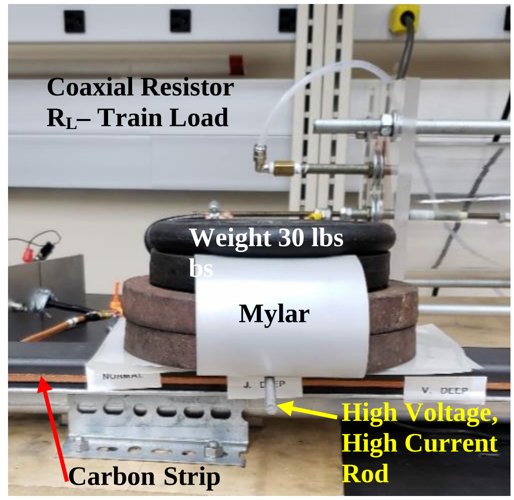


(b)

Annular Ring Coaxial Resistor – Limiting current resistor



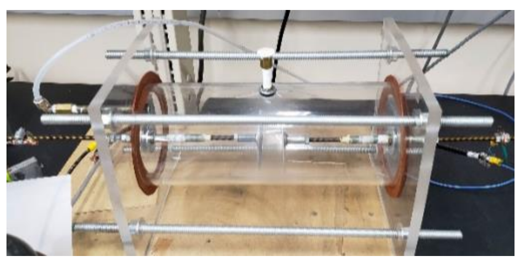
(c)



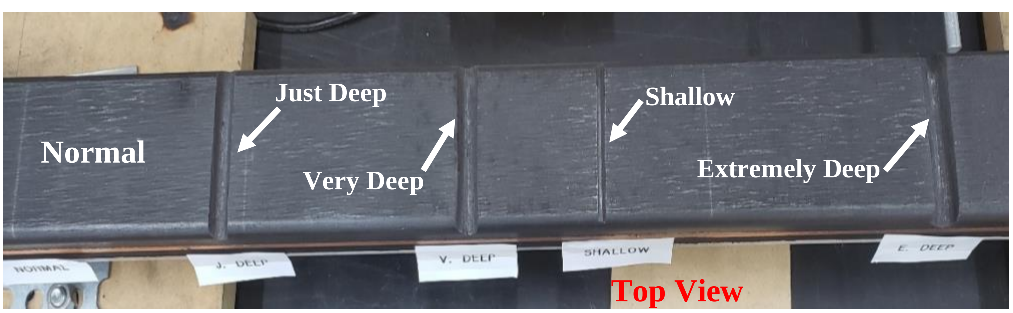
(d)

(e)

Self-Firing Plasma Discharge Tube – Multi-pulse mode



(f)



(g)

FIGURE 2 Isolated Sections of the Pulsed-Power Carbon Strip Degradation Test Stand. (a) Two coaxial based EM dots are mounted in a metal/plastic mount on the ground line of the test stand. The dots are mounted in such a manner to optimize measuring the magnetic field generated by the line current. The ground line is the black coated wire passing through the dot holder. (b) A manganin rod (early design; current design uses pulsed-power resistors not shown) is used as a very low resistor to measure the in-line current. This diagnostic is used as a comparison to the measurement made with the EM dot. (c) The manganin rod (pulsed power resistors not shown) and EM dots are connected in series on the ground line leading to the ground side of the capacitor. (d) The high voltage, high current rod in the test stand is weighted down with 30 lb weights needed for stable measurements. Mylar sheets are used to insulate the iron weights from the rod and carbon strip. (e) A top view of the copper sulfate resistors is shown. The resistors can support the heat loads of the test stand and attain the low resistance values between 0.1 and 15 Ω. The $CuSO_4$ liquid coaxial resistors are not very stable. When accuracy and verification studies are conducted, the manganin rod and the liquid coaxial cable resistor representing the train load are replaced with the pulsed-power, solid-state resistors (1.1 Ω to 7.2 Ω). Refer to Tables 4, 5a, and 6b in Section V. (f) The high voltage/high current multi-pulsing property of the test stand is achieved by a self-firing plasma discharge tube operating about the Paschen voltage based on the tube's internal pressure. (g) An overview of the carbon strip's shallow and deep gouges. Refer to Table 1 for the resistance of each groove studied. Refer to Figure 3 for a typical groove geometry. For select train loads, small but significant changes in the carbon strip resistance were detected.

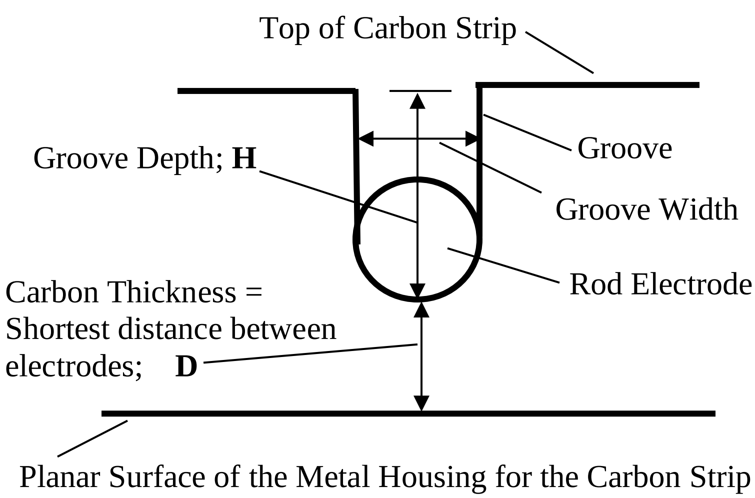


FIGURE 3 Simple Sketch of Rod Electrode, Metal Housing of Carbon Strip, Groove, and Their Relative Locations

SIMPLE ELECTRICAL TRAIN MODEL AND LIMITATION

To determine what limitations exist with monitoring the current flow through the carbon strip, the electrical load of the train is modeled as a simple resistor, R_L. The measured inductance L_m and measured resistance R_W is a consequence of the wires and wire connectors in the pulsed-power carbon strip degradation test stand. The measured resistance of the in-line ammeter is R_M. The loading effect of the carbon strip is characterized by its carbon strip resistance R_c. Electrically, each of these loading effects are in a series configuration with each other. The overhead electrical source line was modeled in the test stand with a 0.25" diameter metallic rod weighted with a 30 lb weight (Refer to Figure 2d). The carbon strip resistance is measured across the carbon strip defect with weighted rod and the metallic housing of the strip. Unstable resistive measurements were observed if the pound force of the rod was less than 30 lbs. This is consistent with findings in literature.
A 5 kV DC voltage source set at v_s= 2.5 kV DC is used to energize the capacitor bank C_b and the discharge tube capacitance. A self-discharging pulsed power discharge tube (Figures 1 and 2f) supplies 10 microsecond in duration pulses of current to the electrical circuit. The carbon strip degradation test stand with monitoring dots are illustrated in Figures 1 and 2a-g.

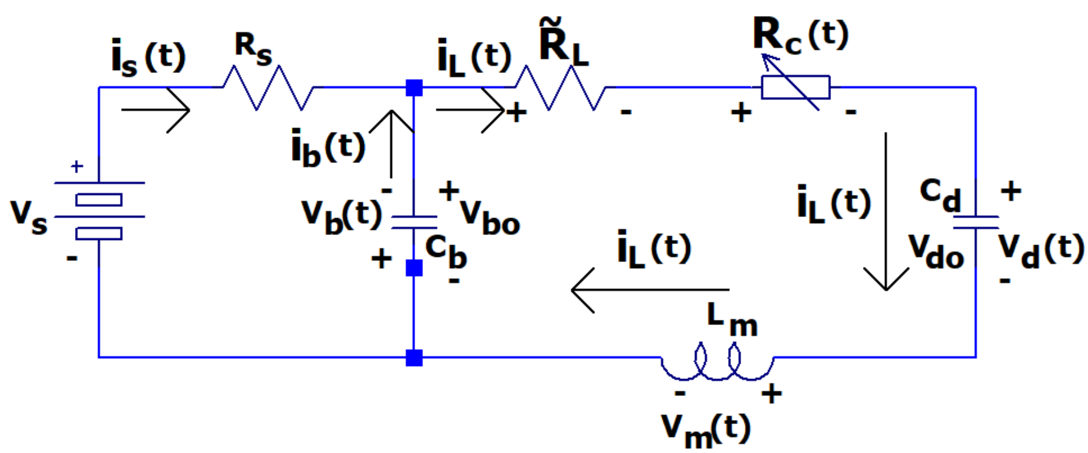


FIGURE 4 The Circuit Model for Charging Mode.
The discharge tube is treated as a capacitor C_d charging up to the Paschen voltage. This circuit model is only valid in the charging mode. The inductance L_m is measured with an LCR meter with all components in the circuit excluding the discharge tube.

Theory

The theory can be separated in terms of a Charging Mode State followed by a Discharging Mode State. Using the final conditions of one state for the initial conditions of the consecutive state, one can build a time history of the charging and discharging effects. A pulsed power theoretical and experimental model was chosen to approximately represent the case of a DC and an AC train configuration simultaneously as well as delivering near typical fast rail currents. Note for simplicity the equivalent (effective) lumped train load resistance $\tilde{R}_L$ is used without loss of generality $\tilde{R}_L = R_L + R_W + R_M$ For clarityR_L, R_W, and R_M are respectively the train load resistance, the resistance of the wires and wire connectors, and in-line resistance of the current measuring resistor (in-line ammeter monitor). The electrical parameters with typical values are listed in Table 2 below.

TABLE 2 Electric Circuit Components with Values for the Carbon Strip Degradation Test Stand

Component	Circuit Label	Value or Range of Values
C_b	Capacitor bank capacitance	3.3 μF measured with LCR meter
V_s	External 5 kV DC voltage source	2.5 kV – Paschen breakdown value 2 kV (more like 1.9 kV) exp.
R_s	Limiting resistor	1 kΩ measured
R_L	Load resistance of locomotive using pulsed-power solid-state resistors	1.1 Ω < R_L < 7.2 Ω meas. with LCR meter
R_W	Wire and connector resistances	R_W=0.14 Ω measured with LCR meter at the terminals disconnected from the capacitor bank with relays in discharge mode and discharge tube shorted.
R_c	Carbon strip resistance	0.003 Ω <R_c < 0.028 Ω meas.
R_M	In-line ammeter resistor sensor	R_M = 10 mΩ meas.
* C_d	Capacitance of the discharge tube based on the parallel plate electrode disks	3.16 pF meas. with LCR meter CALCULATION: Plate radius -- 1.38" plate separation 1" free space - - - 1.34 pF
L_m	Wire inductance in the discharge circuit	Typical 3.32 μH; Measured LCR meter at 200 kHz
R_d	Discharge resistance in the discharge tube to represent internal tube losses	Based on discharge tube volt (~2 kV) and current (3.75 kA) measurements; ratio - 0.53 Ω
V_{bo}	Initial condition for the capacitor bank	Depends on the charging circuit, typical value is near the Paschen breakdown voltage
V_{do}	Initial condition for the electrode capacitance in the discharge tube	Typically at the Paschen voltage
$\tilde{R}_L = R_L + R_W + R_M$	Equivalent lumped train load resistance (Effective train load resistance)	Used as a simplification in the theoretical study

* The LTSpice and MATLAB voltage and current simulations are insensistive to values of C_d between 3.16 pF and 31.6 nF.

Charging Mode

A simple circuit model of a fast rail with a pulsed-power laboratory source is provided in Figure 4. This circuit is valid when in charging mode. The orientation of the branch voltages and currents are provided in the circuit. Kirchhoff's voltage law is applied to paths 'a' and 'b' in the circuit. Although the experiment makes use of static carbon strip loads, a theory is developed allowing the carbon strip resistance to change with time. A brief theory will be presented assuming a DC source voltage, V_s. Based on the branch currents and voltages in the charging circuit (Refer to Figure 4), the governing loop relations are

$$V_s = i_s(t)R_s - \frac{1}{C_b}\int_{t_{oc}^+}^{t} i_b(t)dt + V_{bo}(t_{oc}^+) \tag{1a}$$

$$\frac{1}{C_b}\int_{t_{oc}^+}^{t} i_b(t)dt - V_{bo}(t_{oc}^+) + \left(\tilde{R}_L + R_c(t)\right) i_L(t) + \frac{1}{C_d}\int_{t_{oc}^+}^{t} i_L(t)dt + V_{do}(t_{oc}^+) + L_m\frac{di_L(t)}{dt} = 0 \tag{1b}$$

where

$$\tilde{R}_L = R_L + R_W + R_M. \tag{2}$$

The branch currents are related by the following nodal equation

$$i_b(t) = i_L(t) - i_s(t) \tag{3}$$

Upon differentiating Equation (1a) and Equation (1b) with respect to time and manipulating the relations, the following expressions are obtained in terms of the source and load currents

$$\frac{di_s(t)}{dt} + \frac{1}{\tau_{bs}} i_s(t) = \frac{1}{\tau_{bs}} i_L(t) \tag{4}$$

$$\frac{d^2 i_L(t)}{dt^2} + \frac{1}{\tau_{mLc}(t)}\frac{di_L(t)}{dt} + [\omega_{mbd}^2 + \omega_{mc}^2(t)]i_L(t) = \omega_{mb}^2 i_s(t) \tag{5}$$

where

$$C_{bd} = \frac{C_b C_d}{C_b + C_d},\ \tau_{bs} = R_s C_b,\ \ \omega_{mbd}^2 = \frac{1}{L_m C_{bd}},\ \omega_{mc}^2 = \frac{1}{L_m C_b},\ \tau_{mLc} = \frac{L_m}{\tilde{R}_L + R_c(t)},\ \omega_{mb}^2 = \frac{1}{L_m}\frac{dR_c(t)}{dt}$$

Decoupling Equations (4) and (5) in terms of the carbon strip current $i_L(t)$ yields

$$\frac{d^3 i_L(t)}{dt^3} + A_1(t)\frac{d^2 i_L(t)}{dt^2} + A_2(t)\frac{di_L(t)}{dt} + A_3(t)i_L(t) = 0 \tag{6}$$

where

$$A_1(t) = \frac{1}{\tau_{mLc}(t)} + \frac{1}{\tau_{bs}} \tag{7a}$$

$$A_2(t) = \frac{d}{dt}\frac{1}{\tau_{mLc}(t)} + \omega_{mbd}^2 + \omega_{mc}^2(t) + \frac{1}{\tau_{bs}\tau_{mLc}(t)} \tag{7b}$$

$$A_3(t) = \frac{d\omega_{mc}^2}{dt} + \frac{\omega_{mbd}^2 + \omega_{mc}^2(t) - \omega_{mb}^2}{\tau_{bs}} \tag{7c}$$

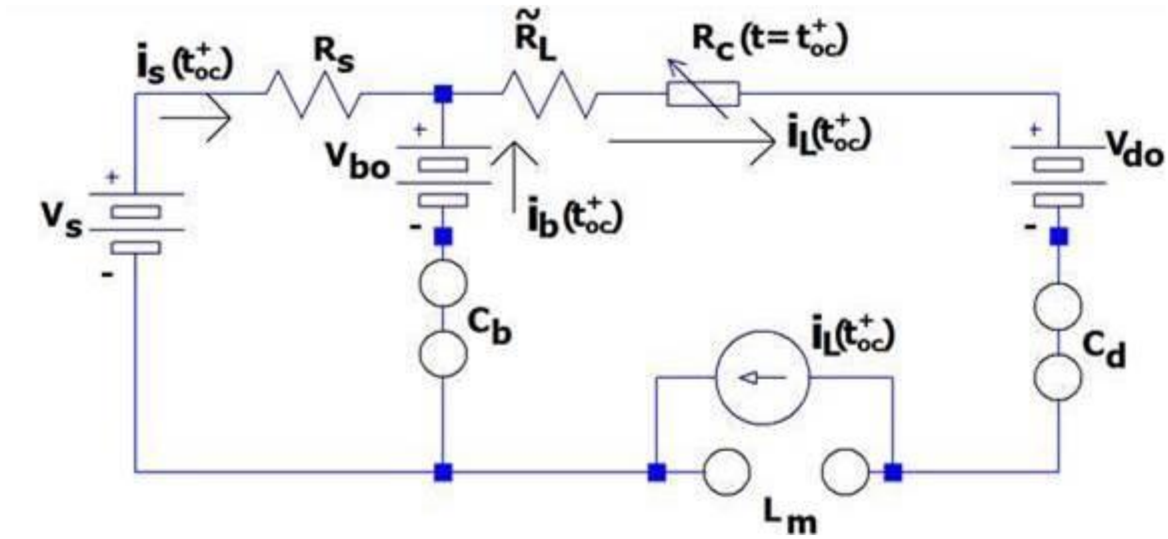


FIGURE 5 The Circuit Represents the Initial Conditions of the Charging Circuit.
Inductive currents and capacitive voltages are continuous in time. At time $t = t_{oc}^+$ the first pulse is being initiated in the circuit.

To complete the charging set of relations, three initial conditions on $i_L(t)$ are required at $t = t_{oc}^+$. Note, subscript 'c' on t_{oc}^+ implies charging. Referring to Equations (1-3) and the initial condition circuit model in Figure 5, the initial condition relations are

$$i_L(t_{oc}^+) = \frac{V_{bo}(t_{oc}^+) - V_{do}(t_{oc}^+)}{R_L + R_c(t_{oc}^+)} \quad (8a)$$

$$\frac{i_L(t_{oc}^+)}{dt} = \frac{[V_{bo}(t_{oc}^+) - V_{do}(t_{oc}^+)] - [\tilde{R}_L + R_C(t_{oc}^+)]}{L_m} \quad (8b)$$

$$\frac{d^2 i_L(t_{oc}^+)}{dt^2} = -\frac{1}{\tau_{mLc}(t_{oc}^+)}\frac{di_L(t_{oc}^+)}{dt} - [\omega_{mbd}^2 + \omega_{mc}^2(t_{oc}^+)]i_L(t_{oc}^+) + \omega_{mb}^2 i_s(t_{oc}^+) \quad (8c)$$

$$i_s(t_{oc}^+) = \frac{V_s(t_{oc}^+) - V_{bo}(t_{oc}^+)}{R_s} \quad (9)$$

Equation (9) comes directly from Equation (1a) and is needed in Equation (8c). For clarity, the voltage across the capacitor cannot change instantaneously, implying $V_{bo}(t_{oc}^+) = V_{bo}(t_{oc}^-)$ and $V_{do}(t_{oc}^+) = V_{do}(t_{oc}^-)$. Similarly at the inductor, $i_L(t_{oc}^+) = i_L(t_{oc}^-)$. For the special case when the circuit is not energized, the following relations are forced $V_{bo}(t_{oc}^-) = 0$, $V_{do}(t_{oc}^-) = 0$, and $i_L(t_{oc}^-) = 0$. Under these conditions, Equations 8a-c with the aid of Equation (9) simplify to: $i_L(t_{oc}^+) = 0$, $\frac{di_L(t_{oc}^+)}{dt} = 0$, and $d^2 i_L(t_{oc}^+)/dt^2 = \omega_{mb}^2 i_s(t_{oc}^+)/R_s$.

As the voltage drop across the discharge plate electrodes approaches the Paschen voltage, the gas in the tube breaks down and allows current to flow in the discharge network. Under this condition, the discharge electrodes are modeled as a discharge capacitor, C_d, in parallel with a discharge resistance, R_d.

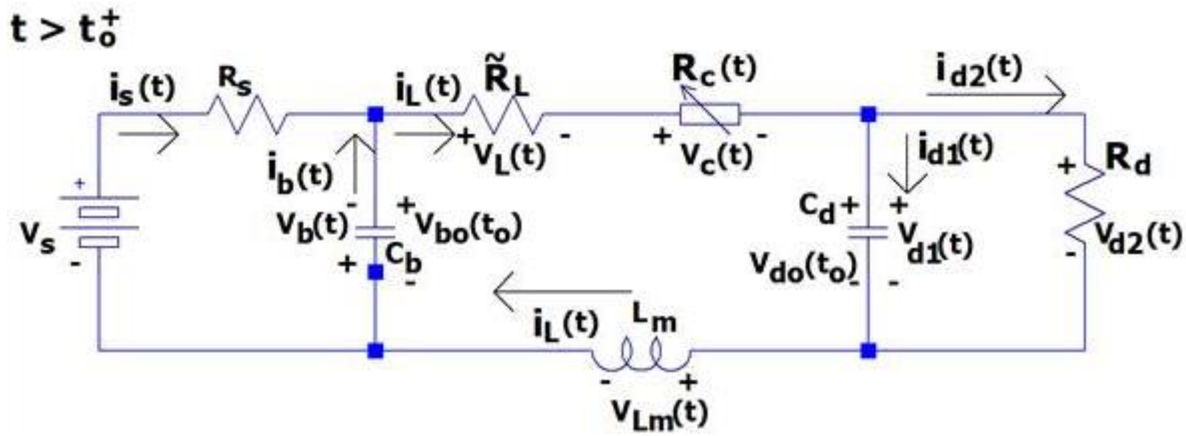


FIGURE 6 The Electrical Circuit Characterizing the Discharge Mode.

The discharge tube is modeled as a discharge capacitor C_d with small capacitance in parallel with a discharge resistor R_d. The discharge circuit is designed to fire when the voltage V_{d1} across the discharge capacitor C_d is the Paschen voltage observed in experiment. Two types of currents pass through R_d. The current released by the modeled gas discharge capacitor C_d and the current released by the capacitor bank C_b. The resistor R_s is large enough to neglect its effects on the time scale of the charging and discharging network. The current released by the modeled gas discharge capacitor C_d is contained in the C_d-R_d portion of the network and released quickly at $t = t_o^+$. Because C_d is small and R_d is small, this discharge does not significantly affect the current $i_L(t)$ released by the capacitor bank C_b. Further, it does not contribute to the current $i_L(t)$. Only the current $i_L(t)$ is measured by the UNLV patented EM dots.

Discharging Mode

The circuit in Figure 6 is valid in this mode. The branch currents and branch voltages are properly referenced in the figure. Writing Kirchhoff's voltage law for each window yields the following three governing equations:

$$V_s = i_s(t)R_s - \frac{1}{C_b}\int_{t_o^+}^{t} i_b(t)dt + V_{bo}(t_o^+) \tag{10}$$

$$\frac{1}{C_b}\int_{t_o^+}^{t} i_b(t)dt - V_{bo}(t_o^+) + \tilde{R}_L + R_c(t))i_L(t) \tag{11}$$

$$+\frac{1}{C}\int_{t_o^+}^{t} i_{d1}(t)dt + V_{do}(t_o^+) + L_m\frac{di_L(t)}{dt} = 0$$

$$i_{d2}(t)R_d = \frac{1}{C_b}\int_{t_o^+}^{t} i_{d1}(t)dt + V_{do}(t_o^+) \tag{12}$$

where the initial time in the *discharge* mode is at t_o^+. Note that t_{oc}^+ and t_o^+ are different in general. The two node equations characterizing the current flow are

$$i_b(t) = i_L(t) - i_s(t) \tag{13}$$

$$i_{d1}(t) = i_L(t) - i_{d2}(t) \tag{14}$$

Differentiating Equations (10)-(12) with time and rearranging yields

$$\frac{di_s(t)}{dt} + \frac{1}{\tau_{bs}} i_s(t) = \frac{1}{\tau_{bs}} i_L(t) \tag{15}$$

$$\frac{d^2 i_L(t)}{dt^2} + \frac{1}{\tau_{mLc}(t)}\frac{di_L(t)}{dt} + \frac{1}{\tau_{bs}}[\omega_{mbd}^2 + \omega_{mc}^2(t)]i_L(t) = \omega_{mb}^2 i_s(t) + \omega_{md}^2 i_{d2}(t) \tag{16}$$

$$\frac{di_{d2}(t)}{dt} + \frac{1}{\tau_{bd}} i_{d2}(t) = \frac{1}{\tau_{dd}} i_L(t) \tag{17}$$

where

$$C_{bd} = \frac{C_b C_d}{C_b + C_d},\ \tau_{bs} = R_s C_b,\ \omega_{mbd}^2 = \frac{1}{L_m C_{bd}},\ \omega_{mb}^2 = \frac{1}{L_m C_b},\ \tau_{mLc} = \frac{L_m}{\tilde{R}_L + R_c(t)} \tag{18a-h}$$

$$\omega_{md}^2 = \frac{1}{L_m C_d},\ \omega_{mc}^2 = \frac{1}{L_m}\frac{dR_c(t)}{dt},\ \tau_{dd} = R_d C_d$$

Decoupling Equations (13)-(17) in terms of the current passing through the carbon strip yields

$$\frac{d^4 i_L(t)}{dt^4} + B_1 \frac{d^3 i_L(t)}{dt^3} + B_2(t)\frac{d^2 i_L(t)}{dt^2} + B_3(t)\frac{di_L(t)}{dt} + B_4(t) i_L(t) = 0 \tag{19}$$

where

$$B_1(t) = \frac{1}{\tau_{mLc}(t)} + \frac{1}{\tau_{bs}} + \frac{1}{\tau_{dd}} \tag{20a}$$

$$B_1(t) = 2\frac{d}{dt}\left(\frac{1}{\tau_{mLc}(t)}\right) + \omega_{mbd}^2 + \omega_{mc}^2(t) + (\frac{1}{\tau_{bs}} + \frac{1}{\tau_{dd}})\frac{1}{\tau_{mLc}(t)} + \frac{1}{\tau_{bs}\tau_{dd}} \tag{20b}$$

$$B_3(t) = \frac{d^2}{dt^2}\left(\frac{1}{\tau_{mLc}(t)}\right) + 2\frac{d\omega_{mc}^2(t)}{dt} + \left(\frac{1}{\tau_{bs}} + \frac{1}{\tau_{dd}}\right)\left[\frac{d}{dt}\left(\frac{1}{\tau_{mLc}(t)}\right) + \omega_{mbd}^2 + \omega_{mc}^2\right] + \frac{1}{\tau_{bs}\tau_{dd}\tau_{mLc}(t)} - \frac{\omega_{mb}^2}{\tau_{bs}} - \frac{\omega_{md}^2}{\tau_{dd}} \tag{20c}$$

$$B_4(t) = \frac{d^2\omega_{mc}^2(t)}{dt^2} + \left(\frac{1}{\tau_{bs}} + \frac{1}{\tau_{dd}}\right)\frac{d\omega_{mc}^2}{dt} + \frac{\omega_{mbd}^2 + \omega_{mc}^2(t)}{\tau_{bs}\tau_{dd}} - \frac{\omega_{mb}^2 + \omega_{md}^2}{\tau_{bs}\tau_{dd}} \tag{20d}$$

Equation (19) will require four initial conditions $i_L(t_o^+)$, $\frac{di_L(t_o^+)}{dt}$, $\frac{d^2 i_L(t_o^+)}{dt^2}$, $\frac{d^3 i_L(t_o^+)}{dt^3}$ in order to obtain a unique solution. Directly, Equations (10) and (12) at $t = t_o^+$ yield, respectively,

$$i_s(t_o^+) = \frac{V_s(t_o^+) - V_{bo}(t_o^+)}{R_s} \tag{21a}$$

$$i_{d2}(t_o^+) = \frac{V_{d0}(t_o^+)}{R_d} \tag{21b}$$

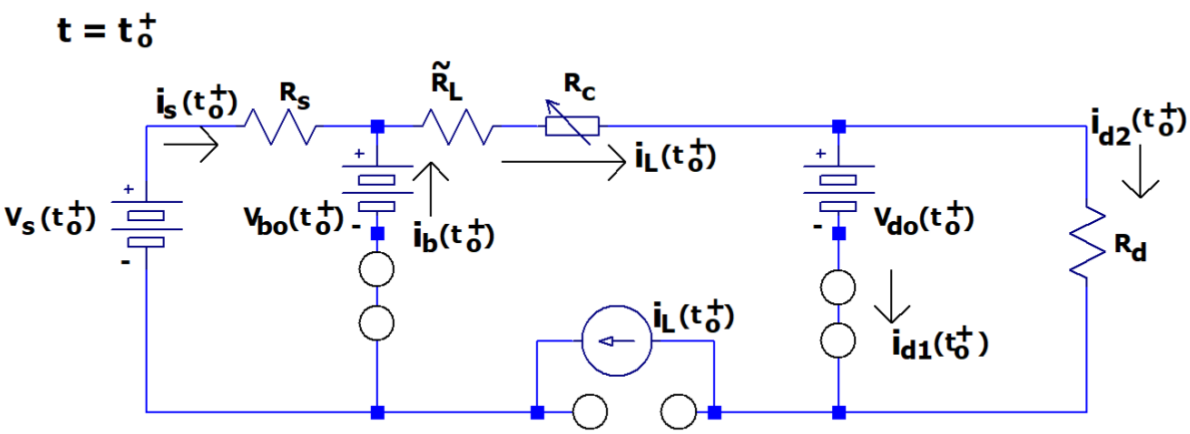


FIGURE 7 The Electrical Representation of the Discharge Circuit at Time $t = t_o^+$.
At this time, the discharge voltage $V_{do}^+(t_o^+)$has reached the Paschen voltage in the charging cycle. The Paschen voltage is the threshold voltage for the gas in the discharge tube to breakdown releasing an electrical pulse.

The circuit in Figure 7 directly yields the first initial condition

$$i_L(t_o^+) = \frac{V_{bo}(t_o^+) - V_{do}(t_o^+)}{\tilde{R}_L + R_c(t_o^+)} \tag{22a}$$

Equation (11) at $t = t_o^+$ to yield the second initial condition

$$\frac{di_L(t_o^+)}{dt} = \frac{[V_{bo}(t_o^+) - V_{do}(t_o^+)] - [\tilde{R}_L + R_c(t_o^+)]i_L(t_o^+)}{L_m} \tag{22b}$$

The third initial condition comes from Equation (16) making use of Equations (21a,b) giving

$$\frac{d^2 i_L(t_o^+)}{dt^2} = -\frac{1}{\tau_{mLc}(t)}\frac{di_L(t_o^+)}{dt} - [\omega_{mbd}^2 + \omega_{mc}^2(t)]i_L(t_o^+) + \omega_{mb}^2\frac{V_s(t_o^+) - V_{bo}(t_o^+)}{R_s} + \omega_{md}^2\frac{V_{do}(t_o^+)}{R_d} \tag{22c}$$

The final initial condition at $t = t_o^+$ is attained by operating on Equation (16) using $[\frac{d}{dt} + \frac{1}{\tau_{dd}}]$. Simplifying using Equations (21a,b) results in

$$\frac{d^3 i_L(t_o^+)}{dt^3} = -M_1(t_o^+)\frac{d^2 i_L(t_o^+)}{dt^2} - M_2(t_o^+)\frac{di_L(t_o^+)}{dt} - M_3(t_o^+)i_L(t_o^+) + [\frac{1}{\tau_{dd}} + \frac{1}{\tau_{bs}}]\omega_{mb}^2 i_s(t_o^+) \tag{22d}$$

where

$$M_1(t_o^+) = \frac{1}{\tau_{mLc}(t_o^+)} + \frac{1}{\tau_{dd}} \tag{23a}$$

$$M_2(t_o^+) = \frac{d}{dt}\frac{1}{\tau_{mLc}(t_o^+)} + \omega_{mbd}^2 + \omega_{mc}^2 + \frac{1}{\tau_{dd}\tau_{mLc}(t_o^+)} \tag{23b}$$

$$M_3(t_o^+) = \frac{d\omega_{mc}^2(t_o^+)}{dt} + \frac{[\omega_{mbd}^2 + \omega_{mc}^2(t_o^+)]}{\tau_{dd}} - [\frac{\omega_{mc}^2}{\tau_{bs}} + \frac{\omega_{md}^2}{\tau_{dd}}] \tag{23c}$$

From Figure 6 as the discharge resistance approaches infinity, the discharging network becomes the charging network. One can show that the relations for the charging network state with initial conditions are recovered from the discharging network state. Here, the carbon strip resistance, $R_c(t)$, is a function of time. The time varying carbon strip resistance is solely responsible for the time varying coefficients of relations Equations (6) and (19). Allowing the carbon strip resistance to be constant in time, one independently recovers the third and fourth order circuit solutions with constant coefficients.

In order to apply the initial conditions of the discharge mode, the final conditions of the charging state are sought. Let the final charging time in the charging mode be $t = t_f^- = t_o^-$ whereas the initial charging time in the discharge state is $t = t_o^+$. At the capacitors, C_b and C_d, the voltage is required to be continuous when transitioning between states implying

$$v_b(t = t_f^- = t_o^-) = v_b(t = t_o^+) = V_b(t_o^+) \tag{24a}$$

$$v_d(t = t_f^- = t_o^-) = v_d(t = t_o^+) = V_d(t_o^+) \tag{24b}$$

Similarly, the current at an inductor must be continuous

$$i_L\left(t = t_f^- = t_o^-\right) = i_L(t = t_o^+) = I_{Lo}(t_o^+) \tag{24c}$$

The initial conditions in the discharging mode Equations (22a-23c) are dependent on $V_{bo}(t_0^+)$, $V_{do}(t_0^+)$, and $I_L(t_0^+)$.

Based on Paschen effects, the Paschen voltage, $V_{Paschen}$, is determined directly from monitoring the voltage drop across the discharge tube. At the time the discharge tube fires, the Paschen voltage has been reached, $v_d\left(t = t_f^- = t_o^-\right) = V_{Paschen}$. In the charging circuit, the current charging the discharge tube is $i_L(t)$ characterized by Equation (6). The final charging time which is the initial discharging time is determined from

$$v_d\left(t = t_f^- = t_o^-\right) = v_d\left(t = t_f^+ = t_o^+\right) = \frac{1}{C_d}\int_{t_o^+}^{t=t_o^+} i_d(t)dt + V_{do}(t_o^+) = V_{Paschen} \tag{25}$$

At $t = t_f^-$, Equations (22a, b) is used to determine both $i_L\left(t = t_f^- = t_o^-\right)$ and $\frac{di_L(t_f^- = t_o^+)}{dt}$. Substituting in Equation (1b) yields

$$V_b(t_0^-) = -[\left(\tilde{R}_L + R_c(t_l^-)\right) i_L\left(t_f^-\right) + v_d\left(t_f^-\right) + L_m \frac{di_L\left(t_f^-\right)}{dt}] \tag{26}$$

Theory Implemented with MATLAB

MATLAB is used for calculating the load current based on the theory for both the constant (Appendix A) and the time-varying (Appendix B) carbon resistance. For the constant carbon case, the code solves the charging network by solving a third-order ordinary differential equation (ODE) using the substitution method to reduce higher-order ODE to three first order differential equations. For better representation, a symbolic notation is used to represent these ODEs. Using "matlabFunction", the reduced symbolic expression is converted to a Matlab function handle "RRcurent" to generate the solution to the third-order ODE using the ODE45 command, taking into consideration the appropriate initial conditions. The same process is followed to solve the fourth-order ODE of the discharging network with the initial conditions based on the final conditions from the charging network.

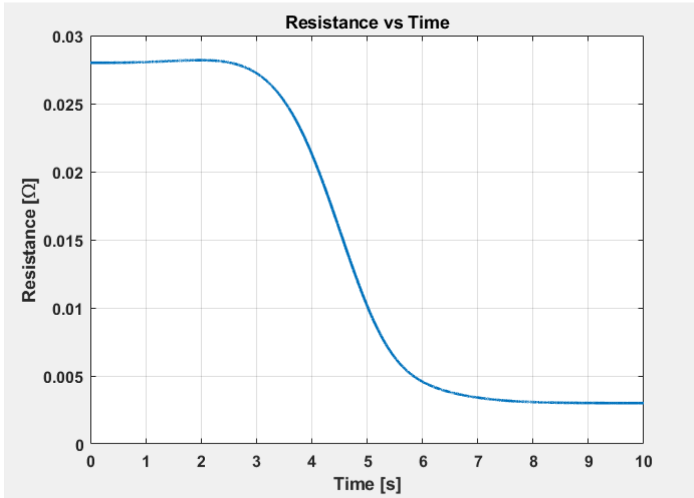


FIGURE 8 The Time Varying Carbon Strip Resistance as a Function of Time; $R_c(t)$. The time varying nature simulates a modeled wear effect due to friction between the power lines and the carbon strip in- transit.

The time-dependent carbon resistance in Figure 8 is characterized by

$$R_c(t) = 0.03 + 0.0125\,\exp[-(0.001t^5 + 0.015t^3)] + 0.0125\exp\,(-0.01t^3) \qquad (27)$$

The carbon resistance is modeled based on physics but limited in scope as shown in Figure 8. It is assumed that the resistance will stay constant over a time range before degradation and degrades quickly until it reaches its minimum value, after which it remains constant.

To solve for the load current, a third-order ODE for the charging network and a fourth-order ODE has to be solved for the discharging network, with time-dependent coefficients in both networks. The program is written in a fashion that the charging network and the discharging network work concertedly. The final conditions from the charging network act as the initial conditions for the discharging network and they are time-dependent in the discharging stage. The initial conditions are converted from the symbolic value to double-precision value using the “double” command.

Computer Aided Electric/Electronic Design and Modeling Using LTSpice

Figure 9 displays a multi-pulse LTSpice model of the periodic charging and self-discharging of a capacitor bank C_b using a discharge tube as a self-breaking switch. Refer to Appendix C. The discharge tube is modeled by a discharge capacitor C_d in parallel with the series combination of a discharge resistor R_d and a voltage-controlled switch SW1. Refer to Figure 9. The voltage controlled switch forces the condition that the transient discharge state overlaps the transient charging state to emulate the self-charging/self-discharging nature of the discharge tube. A stable equilibrium is never reached. Multiple nearly identical pulses are generated. When the switch voltage V_{c2} exceeds an upper threshold voltage, SW1 is activated. The discharge resistor R_d is connected in parallel to C_d. Both the capacitor bank C_b and the discharge capacitor C_d discharge. The bank capacitor discharges through the train load and carbon strip resistors. The experimentally measured Paschen voltage in the discharge tube is ~ 2 kV. When the switch voltage decreases below a lower voltage threshold, the discharge process of the tube has come to completion. SW1 opens breaking the electrical connection between C_d and R_d electrically floating R_d. The discharge capacitor C_d (and C_b) resumes charging.

Both the charging and the discharging networks contain the electrical train loading effect R_L, the carbon strip resistance R_c, the in-line current sensing resistor R_2, the overall wire and connector resistances of the test stand R_W, the overall inductance of the network L_m, and the discharge tube network. The resistor R_s is used to decrease [increase] the number of pulses per unit time by decreasing [increasing] the flow of current from the voltage source V_s to the capacitor bank C_b in combination with the discharge tube capacitor C_d. The overall inductance is the measured inductance of the test stand in the absence of the capacitor bank and the discharge tube. The discharge tube is replaced with a wire about the length of the tube. Typical values for each element of the network is given in Table 2.

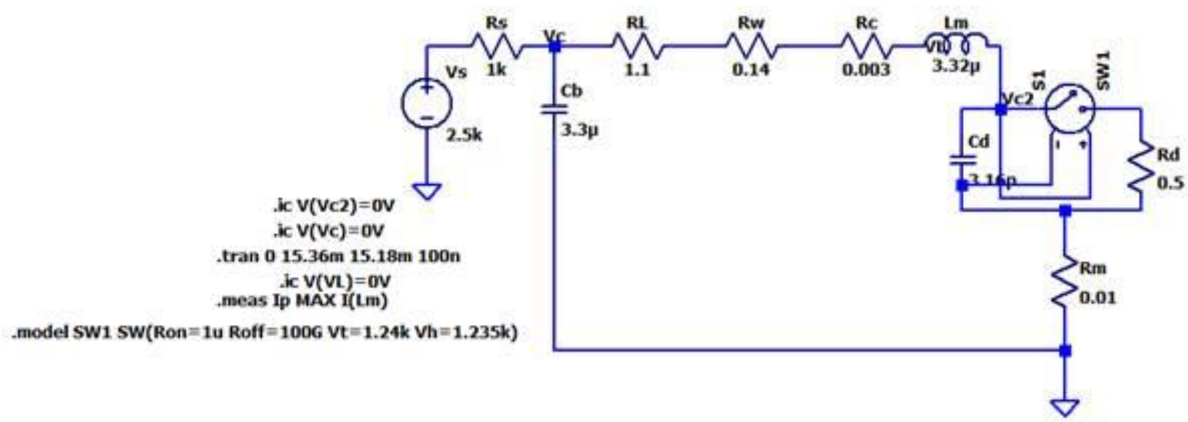
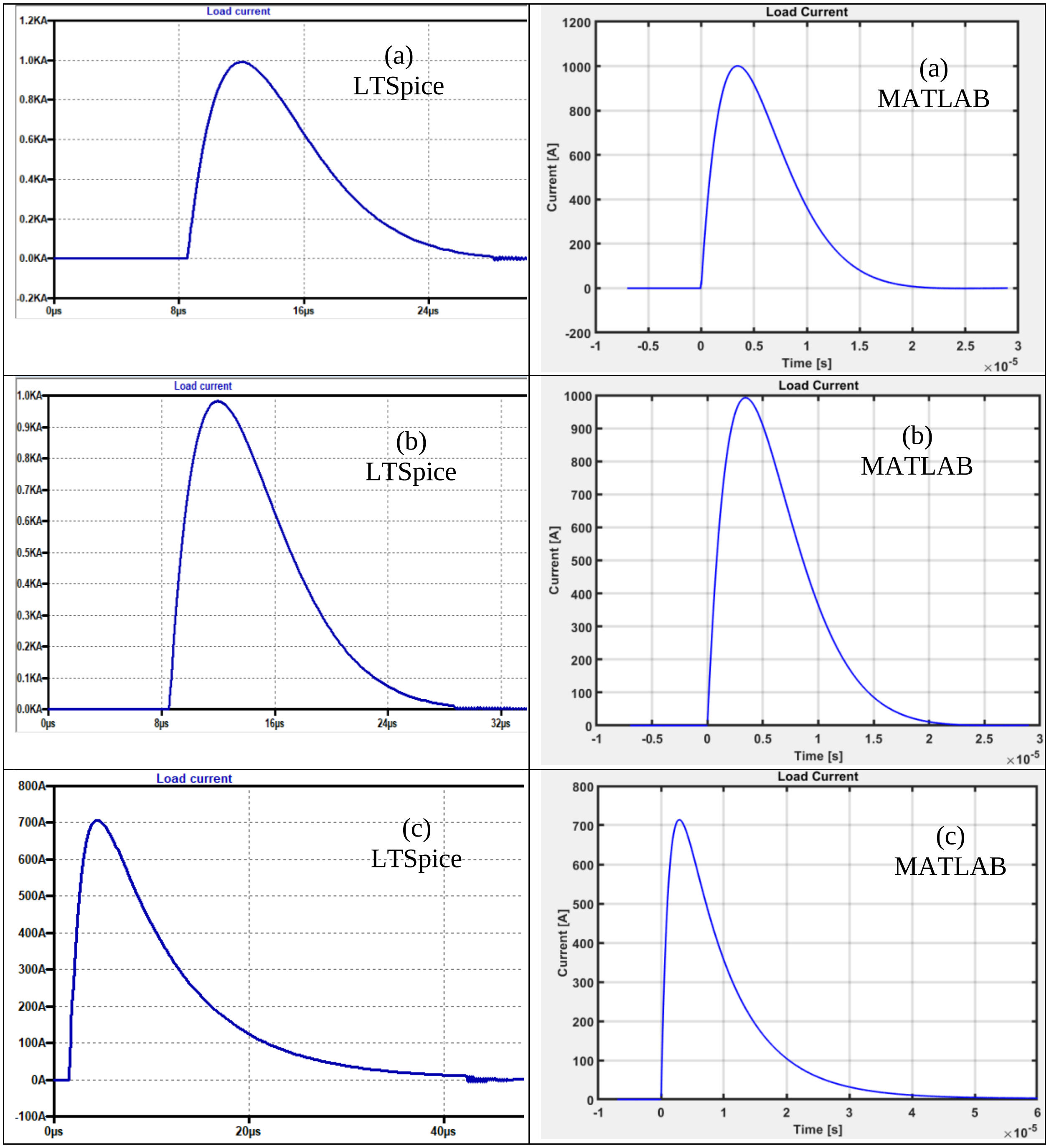


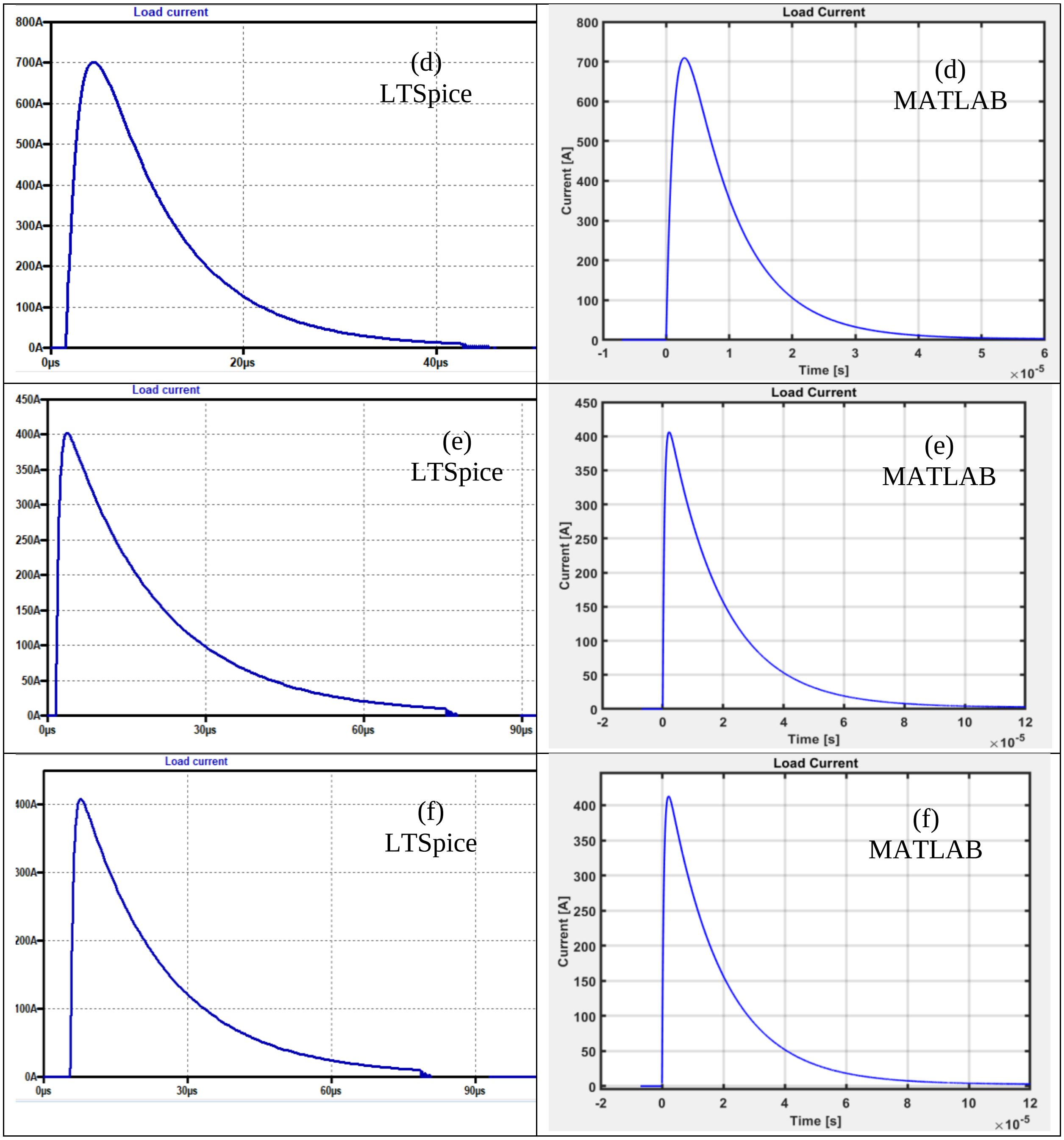
FIGURE 9 LTSpice Circuit Emulating the Pulsed-Power Carbon Strip Degradation Test Stand Used in Experiment.

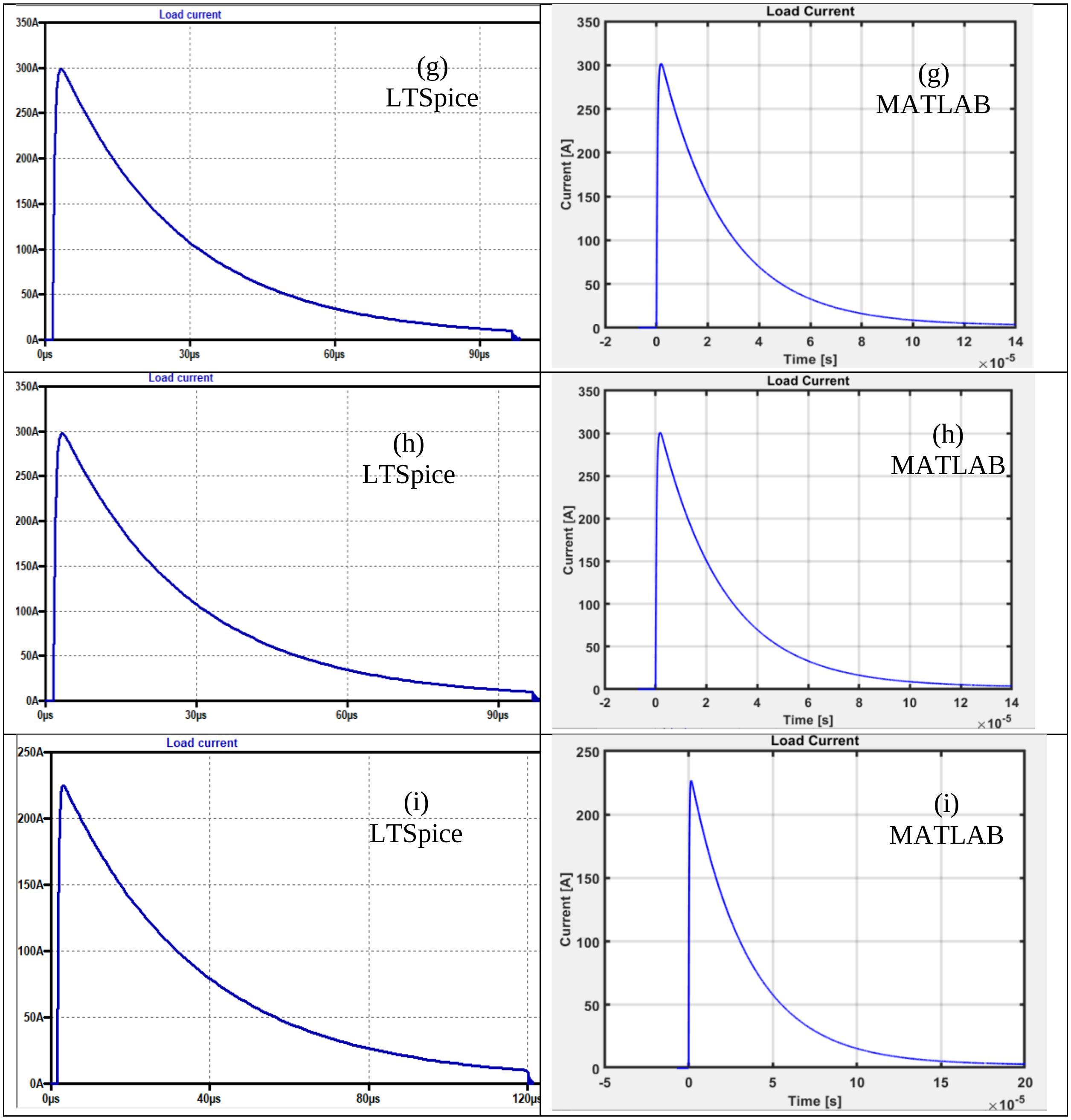
Comparison Between Theory Implemented with MATLAB Software and Computer Aided Electrical/Electronic Design, Analysis, and Modeling Software Using LTSpice

Figures 10 a-j provide a comparison between the theory implemented with MATLAB and the LTSpice computer aided design and modeling software. From Table 3, the modeling software and the theory show exceptional agreement. The percent deviations of the current peaks in the theory and the modeling software is less than 1.5% as shown in Table 3. The difference in pulse widths in theory (MATLAB) and simulation (LTSpice) at 10% of the peak amplitude is less than 1 µs. Since the pulse widths in both theory and simulation are on the order of tens of microseconds, the maximum worst case error is 1µs/10µs=0.1 consequently 10%. It is observed that as the train load resistance R_L increases, the difference of the peak currents at R_{cmin} and at R_{cmax} over the peak current associated with R_{cmin} decreases. The ability to be able to distinguish differences in peak current over the range of carbon strip resistances decreases rapidly.

Good agreement in theory and simulation verifies that the models studied are similar. Further, simulation and theory offers guidance and verification to experimental results assuming that the components and the model is good enough to corroborate the physics of the problem.




Load current
(d)
LTSpice
Load Current
(d)
MATLAB
Current [A]
Time [s]
Load current
(e)
LTSpice
Load Current
(e)
MATLAB
Current [A]
Time [s]
Load current
(f)
LTSpice
Load Current
(f)
MATLAB
Current [A]
Time [s]


Load current
(g)
LTSpice
Load Current
(g)
MATLAB
Current [A]
Time [s]
Load current
(h)
LTSpice
Load Current
(h)
MATLAB
Current [A]
Time [s]
Load current
(i)
LTSpice
Load Current
(i)
MATLAB
Current [A]
Time [s]

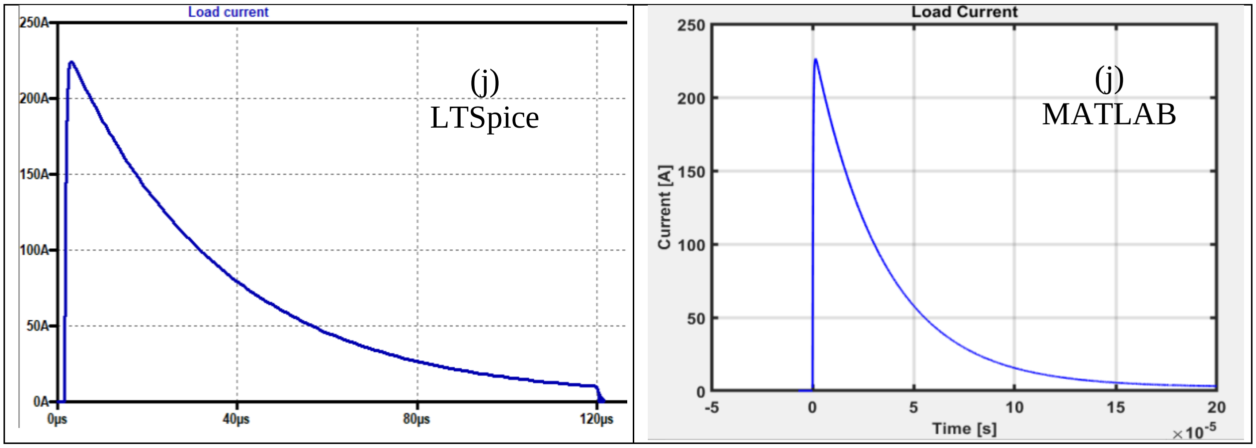


FIGURE 10 Theoretical Results Implemented with MATLAB is Compared to the LTSpice Computer Aided Electric and Electronic Design and Model Simulation.

The LTSpice circuit simulated is displayed in Figure 9. The identifying features of each figure may be found in Table 3. Overall, notice the close similarity between the MATLAB figures and the LTSpice figures. [Experimental parameters: R_W=0.14Ω, L_m=3.32 µH, R_M=10 mΩ, R_s=1kΩ, C_b=3.3 µF, R_d=0.5 Ω and C_d=3.16 pF. Refer to Table 2.]

TABLE 3. Summary of Theory Implemented with MATLAB and Simulation Using LTSpice

Fig. 10 a-j	R_L [Ω]	R_C [mΩ]	I_L peak (MATLAB) [A]	I_L peak (LTSpice) [A]	% Diff. Peaks [1] [%]	Pulse width 10%/90% (MATLAB) [µs]	Pulse width 10%/90% (LTSpice) [µs]	Pulse width Diff. 10% [µs]	[2]Peak Range [% Peak Range] (A) (%)	[3]Estimated peak (R circuit only) [A]
a	1.1	3	1001	990.53	1.05	14.21/3.05	14.3/3.04	0.09	8.7 [0.87]	1668
b	1.1	28	992.3	981.73	1.07	14.43/3.04	14.5/3.01	0.07		1635
c	2.2	3	713.5	706.4	1.00	23.05/3.13	22.08/3.2	0.97	4.5 [0.63%]	898.07
d	2.2	28	709	701.7	1.03	23.26/3.16	23.18/3.14	0.08		888.1
e	5	3	405.5	402	0.86	44.9/3.55	45/3.55	0.1	0.9 [0.22%]	397.9
f	5	28	404.6	400.5	0.98	45.4/3.55	45.2/3.45	0.2		395.9
g	7.2	3	301.8	298.8	0.99	61.8/4.1	62.38/4.1	0.58	0.9 [0.3%]	304.4
h	7.2	28	300.9	297.9	1.00	62.1/4.15	62.58/4.0	0.48		303.4
i	10	3	226.9	224.6	1.01	85.46/4.74	84.8/4.87	0.66	0.5 [0.22%]	219.5
j	10	28	226.4	224.1	1.02	85.5/4.75	85.06/4.88	0.44		218.9

NOTE:

[1] %Diff Peaks= 100*[I_L peak {MATLAB}- I_L peak {LTSpice}]/ I_L peak {MATLAB}

[2] The difference between the current peaks for R_c=28 mΩ and R_c=3 mΩ based on MATLAB results. The % peak range is the range compared to the maximum peak current in MATLAB.

[3] Estimated peak assuming the modeled circuit is purely resistive with capacitor source is at its maximum voltage prior to discharging (Paschen voltage).

EXPERIMENTAL RESULTS

One electromagnetic dot sensor is used to monitor the line current passing from the overhead power line through the carbon strip to the electric, fast rail, locomotive. Various grooves of different depths were built into the carbon strip material. The resistance of each groove blemish was measured with the LCR meter across the weighted rod and the metal support housing the carbon strip. Refer to Table 1 and Figures 2d,g. The measured resistance of the weighted metal rod on the housing support (short condition) was 3 mΩ. The average resistance between the weighted rod and metal housing supporting an unblemished carbon strip is 28 mΩ.

Train Load (Typical)

It is worthy to note that a typical train load is dynamic in nature. Literature suggests that the input electrical train loads, R_L, range between 0.75 and 5 Ω. One author adopted a train load of 60 mΩ per km length of train. (Alnuman et al 2018) The estimated value is based on their Figure 12 and the value given for the electrical rail resistance in Table 1 of their paper (Alnuman et al., 2018). A train load of approximately 3 Ω was adopted by some authors. (Chymera et al 2010, Ciccarelli et al 2011). This was observed by taking the ratio of Figure 18 to Figure 17 in their paper (Ciccarelli et al 2011). Given the locomotive power using the active power, voltage, and resistance relationship ($R=V^2/P$) suggests a high-speed railroad resistance between 0.75 and 1.25 Ω for a 3kV DC power supply line (Mersen, 2021). A train load between 30 and 90 mΩ/km has been used as well. (Razmjou, and Younesi, 2021)

Weighted Rod

Literature findings (Seo et al 2006), (Koyama et al 2014) and (Usuda, 2008) verify that a 30 lb force between the carbon strip housing and the overhead electrical power line is needed to attain good electrical connection with the carbon strip. All experiments were performed with a 30 lb weighted rod.

Special Considerations for Using Pulsed-Power Resistors

Pulsed-power solid-state resistors were tested and studied in the pulsed-power carbon strip degradation test stand. Let the device under test (DUT) be the solid-state resistors. In all cases, the in-line current measuring resistor (e.g., manganin rod) is replaced by a wire short. Further, the carbon strip was in short circuit mode (R_c=3 mΩ). The device under test was inserted in the test stand in place of the train load resistor. Outside of these three implementations, the test stand configuration is unchanged. The test stand is operating in multi-pulse mode.

Figures 11a-c displays typical current pulse generated in the test stand for a 10 mΩ, 1.1 Ω, and 5 Ω resistor under test. The carbon strip degradation test stand contributes an added 0.14 Ω wire and connector loading effect. Note each current pulse has a pulse width of 10 μs and a repetition period of 135 ms as determined directly from the periodic discharge voltage as given in Figure 11d. Note that the peak current decreases as the resistance of the DUT increases. This only occurs when the resistance of the DUT dominates the test stand loading effect.

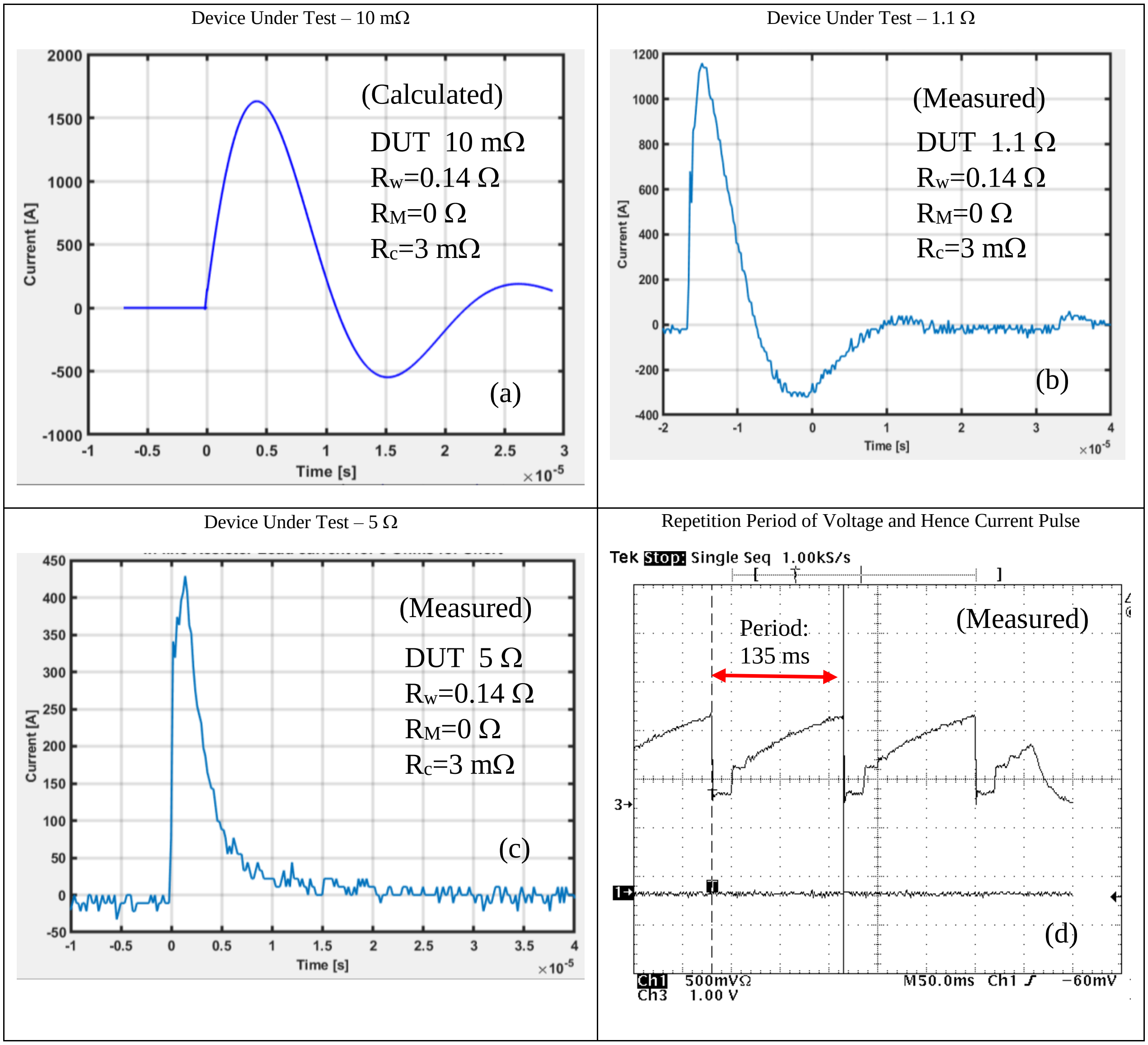


FIGURE 11. Depicts Typical Current and Voltage Discharge Pulses Generated in the UNLV Pulsed-Power Carbon Strip Degradation Test Stand.

Current pulses for a) 10 mΩ resistor (current calculated; peak 1.63 kA), b) 1.1 Ω resistor (current measured; peak 1.16 kA), and c) 5 Ω resistor (current measured; peak 429 A). The test stand's overall wire inductance is measured as L_m=3.32 μH. Adding the resistive load of the test stand wires and connectors (R_W=0.14 Ω), yields an effective resistance of a) 0.15 Ω, b) 1.25 Ω, and c) 5.14 Ω. In all three cases, the pulse width is ~10 μs. The measured period of repetition of the discharge voltage and hence the train load current is d) 0.135 s. As the temperature of the pulsed power resistor changes, the resistance of the pulsed-power resistor changes as indicated in Figure 12. Performing experiments over short intervals of time will maintain the integrity of the resistor and its resistance resulting in minimal error in measurement. Refer to Table 2 for other values.

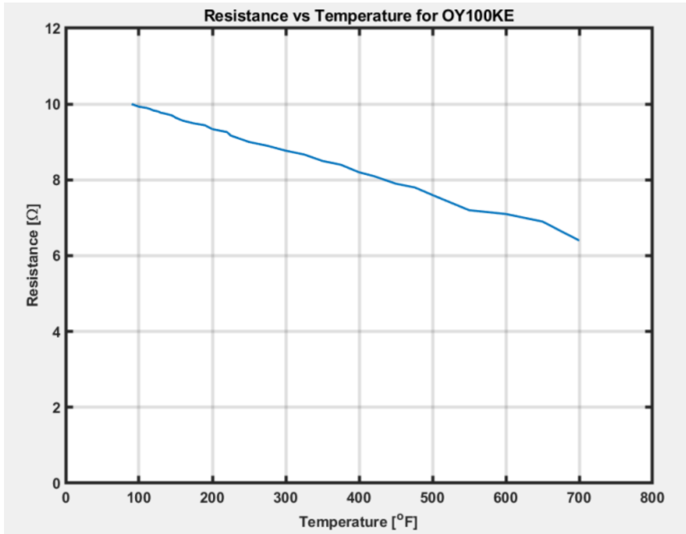


FIGURE 12 **UNLV Laboratory Measured Temperature Characteristics of the 10 □ Ohmite Rated Resistor (Model: OY100KE).**
It is observed that the measured resistance decreases linearly with an increase in temperature. At about 200 °F, the measured resistance decreases to about 9.2 Ω

Table 4 provides achieveable upper limits in temperatures for the discrete solid-state resistors. The time intervals to attain these temperature levels is less than a minute. Further, the resistance after cooling to room temperature is approximately the original room temperature resistance prior to heating. Figure 12 is a resistance versus temperature plot for the 10 Ω Ohmite resistor model number OY100KE. The resistor was Joule heated in the test stand. A craftsman multimeter with heat sensor (thermocouple) was used to measure the temperature. An LCR meter was used to measure the resistance. At 200 °F the rated room temperature 10 Ω resistor decreased to about 9.2 Ω. Note that the resistance decreases linearly with increase in temperature.

The resistance of the coaxial copper sulfate liquid resistor and the manganin rod resistor are unstable. When measured with an LCR meter, the liquid resistor exhibited significant and continuous drift (typically over a range of 10 □ plus). Specific resistances initially sought were between 1 and 15 □. When isolated, resistance measurements of the manganin rod were stable. Used as a resistor, the measured currents were three to five times larger than the theoretical current implying that the resistance of the manganin rod changed significantly. Consequently, select pulsed-power solid-state resistors were examined in the UNLV pulsed-power carbon strip degradation test stand. Both the liquid and manganin resistors have been replaced with solid-state resistors for accuracy, for verification studies, and for final test studies. Some of their properties pertinent to the pulsed-power carbon strip degradation test stand have been studied. Refer to Figures 11a-d for typical pulse configuration experienced by the solid-state resistors under test. Specifications and UNLV laboratory tested studies are provided in the following table. Tests were performed without a fan or external cooling mechanism outside of natural air flow in a laboratory setting. The test stand's overall wire inductance is measured as L_m=3.32 ΩH.

TABLE 4 Initial Tests on the Pulsed-Power Solid-State Resistors Showing Operational Limitations

Rated Resistance (Specified) [Ω]	Resistance (Measured) [Ω]	Company	Model No.	Ratings [W]	Multi-pulse Test	*Time Duration [s]	Monitored Final Temperature [^{O}F]	Resistance (after cooling) [Ω]
1.5	1.6	TE Conn	ER741R5JT	3	✓	14	228	1.6
2.2	2.24	TE Conn	ER742R2JT	3	✓	15	500	2.19
3.3	3.08	Ohmite	OY33GKE	2	✓	50	1050	3.13
4.7	4.5	Ohmite	OY47GKE	2	✓	43	875	4.45
10	10.11	Ohmite	OY100KE	2	✓	36	826	9.82
30 m	31 m	TE Conn	ER74R03KT	3	✓	22	220	31
10 m	11 m	TE Conn	ER74R01KT	3	✓	31	152	11

*The resistors are characterized in the UNLV pulsed-power carbon strip degradation test stand operating in multi-pulsed mode. The time duration is the time it takes for the resistor starting at room temperature to heat to the monitored final temperature.

None of the Ohmite resistors were destroyed even after Joule heating to temperatures peaking well over 1000 ^{O}F. Only smoke was produced as the temperature rose above 400 ^{O}F. Heat tape was used to adhere the heat sensor to the resistor. It is possible that the smoke was a consequence of heat tape use. The resistance of the cooling resistors approached their initial rated value at room temperature. The milli-Ohm resistors (ER74R03KT and ER74R01KT) are exposed to the maximum possible current that would be generated by the test stand. The test stand's resistive load of R_W=0.14 Ω significantly dominates the milli-Ohm resistor's resistance. Consequently, the largest theoretical peak current experienced by the milli-Ohm resistors is ~$V_{Paschen}/R_W$= ~14.3 kA.

Table 5a focuses on the individual resistance of the resistor about the 150 ^{o}F and 200 ^{o}F tempertures using manufactured supplied temperature coefficient of resistance (TCR). At 200 ^{o}F the Ohmite resistors will decrease between 6.7% and 11%. The TE Conn resistors have a positive TCR. It appears that a 1% increase in resistance will result at 200 ^{o}F. Typically, it takes about 10 seconds for the Ohmite resistors to heat to 200 F. To avoid possible 11 % deviations in the Ohmite resistors, experiments are to be performed within a five second period. Test results for a 5 s time duration in the UNLV pulsed-power carbon strip degradation test stand are provided in Table 5b. Here, the resistance range is determined not only for the single resistor but also for the resistor combinations used in experiment. Two tests were conducted for each resistor combination specified in Table 5b. The 7.2 Ω resistor is unique to the extent that the resistor combination is composed of one or more resistors each from two different companies, an equivalent 5 Ω resistor from TE Conn and a 2.2 Ω from Ohmite. Consequently, the extrema of both resistor degradation range are added yielding an extrema resistance range at their respective end temperatures. Refer to Table 6 for calculation details.

In summary, as long as experiments are operating in a time duration less than five seconds and there is sufficient down time for cooling, the resistors will be stable enough to make conclusions regarding pulsed-power carbon strip degradation effects.

TABLE 5a. Temperature Profile Based on Experiment in the UNLV Pulsed-Power Carbon Strip Degradation Test Stand in Multiple Pulsed Mode - A

The temperature baseline for all test studies is the rated room temperature resistance. All tests were conducted in the pulsed-power carbon strip degradation test stand operating in multi-pulse mode where the carbon resistance is the 3 mΩ short. The in-line current monitor resistor is shorted, R_M=0. The wire resistance and inductanceof the test stand are accounted for, R_W=0.14 Ω and L_m=3.32 μH The train load resistance is replaced by the DUT resistor.

Rated Resistance / Power Rating (Specified) [Ω] / [W]	Company / Model No.	[4]TCR [ppm/ºC]	[2]Time Duration [s]	[3]Monitored Final Temperature [ºF]	[1]Resistance Range at 150ºF [Ω]	[1]Resistance Range at 200ºF [Ω]	[5]Min./Max. Error from Rated Res. 200 ºF [Min%] / [Max%]
1.5 / 3	TE Conn / ER741R5JT	0 to 60	14	228	1.5 – 1.5	1.5 – 1.51	0% / 0.67%
2.2 / 3	TE Conn / ER742R2JT	0 to 60	9	~200	2.2 – 2.21	2.2 – 2.21	0% / 0.45%
3.3 / 2	Ohmite / OY33GKE	-1600 to -1000	10	~200	3.09 – 3.17	2.95 – 3.08	6.7% / 11%
4.7 / 2	Ohmite / OY47GKE	-1600 to -1000	16	~200	4.40 – 4.51	4.20 – 4.39	6.6% / 11%
10 / 2	Ohmite / OY100KE	-1600 to -1000	10	~200	9.36 – 9.60	8.93 – 9.33	6.7% / 11%
30 m / 3	TE Conn / ER74R03KT	0 to 60	15	150	30 m – 30 m	DNA	DNA
			23	150			DNA
			30	200	*DNA	30 m – 30 m	0% / 0%
			34	200			0% / 0%
10 m / 3	TE Conn / ER74R01KT	0 to 60	23	145	10 m – 10 m	DNA	DNA
			23	148			DNA
			50	197	DNA	10 m – 10 m	0% / 0%
			55	201			0% / 0%

[1] Calculated: $R_{new}(T°C) = R \pm (\frac{TCR*\Delta T°C*R}{le6})$ where ΔT is the change in (operating temperature – initial temperature) temperature of the resistor [ºC]; R is the value of the resistance of the resistor; and TCR is as given in the table. Examples:

Resistors are at the room temperature (26 ºC or 79 ºF) initially.

At 200 ºF (93 ºC), for OY100KE (10 Ω)

$R_{new} = 10 - \left(\frac{1600*(93-26)*10}{1e6}\right) = 8.93\ \Omega$ or $10 - \left(\frac{1000*(149-26)*10}{1e6}\right) = 9.33\ \Omega.$

At 200 ºF (93 ºC), for ER74R01 (10 mΩ)

$R_{new} = 10e-3 + \left(\frac{200*(93-26)*10e-3}{1e6}\right) = 9.9999\ m\Omega$ or $10e-3 + \left(\frac{-200*(93-26)*10e-3}{1e6}\right) =$ $10.0001\ m\Omega$

[2] The time it takes for the temperature of the resistor in the pulsed-power carbon strip degradation test stand to rise from room temperature to the monitored final temperature (150 ºF or 200 ºF). The resistors larger than 1 Ω took about 10 secs to reach the 200 °F. At 200 ºF, the resistors greater than 1 Ω in the table have less than a 11% change in its room temperature value.

[3] The final temperature monitored typically ~150 ºF or ~200 ºF.

[4] Manufacturer's TCR (Temperature Coefficient of Resistance)

[5] Highlighted numbers in the "Resistance range at 200ºF" column are used in the worst case percent calculation.

* DNA – Does Not Apply

TABLE 5b. Temperature Profile Based on Experiment in the UNLV Pulsed-Power Carbon Strip Degradation Test Stand in Multiple Pulsed Mode - B

The temperature baseline for all test studies is the rated room temperature resistance. All tests were conducted in the pulsed-power carbon strip degradation test stand operating in multi-pulse mode where the carbon resistance is the 3 mΩ short. *The time duration of each experiment is 5 s.* Two tests were conducted for each resistor specified. One set of tests is not highlighted the second in gray highlighted. The in-line current monitor resistor is shorted, R_M=0. The wire resistance and inductance of the test stand are accounted for, R_W=0.14 Ω and L_m=3.32 μH The train load resistance is removed and replaced by the DUT resistor(s). Resistor combinations are included. The following combinations have been examined: two 2.2 Ω resistors in parallel yield an equivalent 1.1 Ω resistor, two 10 Ω resistors in parallel yield an equivalent 5 Ω resistor, and two 10 Ω resistors in parallel with a series 2.2 Ω resistor yield an equivalent 7.2 Ω resistor. The maximum possible resistance deviation based on manufacture's TCR extremes is less than 10% (worst case) maximum deviation and less than 6% (worst case) minimum deviation in the rated room temperature resistance.

Rated Resistance (Specified) [Ω] (Combinations)	Company / Model No.	TCR [ppm/°C]	Time Duration/Particular Resistor Monitored [s] / [Ω]	Monitored Final Temperature [°F]	Resistance Range at Final Temperature [Ω]	[5]Min./Max. Error from Rated Res. Final Temp. [Min%] / [Max%]
1.1 (Two 2.2 Ω resistors in parallel)	TE Conn / ER742R2JT Both Res.	0 to 60	5 / 2.2	180	1.1 – 1.1037	0% / 0.34%
			5 / 2.2	175	1.1 – 1.1035	0% / 0.32%
1.5 (Single)	TE Conn / ER741R5JT	0 to 60	5 / 1.5	238	1.5 – 1.5079	0% / 0.52%
			5 / 1.5	283	1.5 – 1.5102	0% / 0.68%
2.2 (Single)	TE Conn / ER742R2JT	0 to 60	5 / 2.2	420	2.2 – 2.2250	0% / 1.1%
			5 / 2.2	358	2.2 – 2.2205	0% / 0.93%
3.3 (Single)	Ohmite / OY33GKE	-1600 to -1000	5 / 3.3	182	2.9979 – 3.111	5.7% / 9.2%
			5 / 3.3	184	2.9920 – 3.1075	5.8% / 9.3%
4.7 (Single)	Ohmite / OY47GKE	-1600 to -1000	5 / 4.7	182	4.2697 – 4.431	5.7% / 9.2%
			5 / 4.7	163	4.3491 – 4.4791	4.7% / 7.5%
5.0 (Two 10 Ω resistors in parallel)	Ohmite / OY100KE Both Res.	-1600 to -1000	5 / 10	128	4.7822 – 4.8639	2.7% / 4.4%
			5 / 10	127	4.7867 – 4.8667	2.7% / 4.2%
7.2 (Two 10 Ω resistors in parallel, combination in series with 2.2 Ω resistor)	10 Ω Ohmite / OY100KE	-1600 to -1000	5 / 10	125	4.7956 – 4.8722	2.6% / 4.1%
			5 / 2.2	156	2.2000 – 2.1944	0% / 0.26%
	2.2 Ω TE Conn / ER742R2JT		5 / 10	128	4.7822 – 4.8639	2.7% / 4.4%
		0 to 60	5 / 2.2	162	2.2000 – 2.1939	0% / 0.28%
10 (Single)	Ohmite / OY100KE	-1600 to -1000	5 / 10	164	9.2444 – 9.5278	4.7% / 7.6%
			5 / 10	171	9.1822 – 9.4889	5.1% / 8.2%

TABLE 6 Evaluation of the 7.2 □ Rated Resistor Base on a Series Combination of Resistors Manufactured by Two Different Companies

Note that in five seconds the temperature of the resistors in series is not the same. The gray highlighted text is one test and the unhighlighted text is the second test.

Time Duration	Monitored	Individual or Parallel	Resistance Range at	Min./Max. Error from	Combina-tions of	Maximum 7.2 Ω	Min./Max. Error from

[s]	(Measured) Final Temp. [ºF]	Combination Rated Resistance [Ω]	Monitored Final Temp. [Ω]	Rated Res. Final Temp. [Min%] / [Max%]	Extrema Range for 7.2 [Ω]	Resistance Range [Ω]	Rated 7.2 Ω Resistor [Min%] / [Max%]
5	125	5	4.7956 – 4.8722	2.6% / 4.1%	6.996, 6.990, 7.072, 7.067	6.990 – 7.072	1.78% / 2.92%
5	156	2.2	2.2000 – 2.1944	0% / 0.26%			
5	128	5	4.7822 – 4.8639	2.7% / 4.4%	6.982, 6.976, 7.064, 7.058	6.976 – 7.064	1.89% / 3.1%
5	162	2.2	2.2000 – 2.1939	0% / 0.28%			

Parameter Window Estimate

To quantify an estimate on the limitation of the parameter window, the train load and carbon strip load are only modeled by their resistive effects $\tilde{R}_L$and R_c respectively. All resistive load contributions in the test stand (except for R_c) are added to the train load. The ratio of the carbon strip voltage (current) with blemish to the carbon strip voltage (current) without blemish is

$$\bar{V}_n = \frac{R_c}{R_{cn}}\bar{I}_n = \frac{v_c(t)}{v_{cn(t)}} = \frac{R_c}{R_{cn}}\frac{i_c(t)}{i_{cn(t)}} = \frac{R_c}{R_{cn}}\frac{\tilde{R}_L + R_{cn}}{\tilde{R}_L + R_c} \tag{28}$$

where $R_{cn} = 28m\Omega$ and $0.003\Omega \leq R_c \leq 0.028\Omega$. Here, $\bar{V}_n$ and $\bar{I}_n$ are the normalized (unitless) voltage and current based on the carbon thickness of a new carbon strip. When R_c=0.003Ω, $R_c/R_{cn} = 0.11$; when R_c=0.028Ω, $R_c/R_{cn} = 1$ and $\bar{V}_n = 1$ and $\bar{I}_n = 1$. The latter conditions imply that the carbon strip is in its ideal, new, or unblemished state. Consider the range of $\bar{V}_n$ and $\bar{I}_n$, at the extremes in train loads over the range of resistances of the carbon strip. When $\tilde{R}_n = 1\Omega$, (short) $0.1098 \leq \bar{V}_n \leq 1$ and $1 \leq \bar{I}_n \leq 1.025$ (short). For $\tilde{R}_L = 10\Omega$, (short) $0.1074 \leq \bar{V}_n \leq 1$ and $1 \leq \bar{I}_n \leq 1.0025$ (short). Considering the current. For $\tilde{R}_L = 1\Omega$, an approximate 2.5 % increase in current relative to the unblemished strip in its shorted state is observed. For $\tilde{R}_L = 10\Omega$, an approximate 0.25% increase in current relative to the unblemished strip in its shorted state is observed. At this point, explicit experimentation is required in order to determine if the percent changes are high enough to monitor carbon strip degradation in a noisy signal environment.

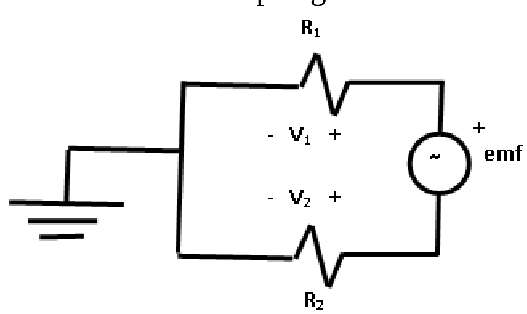


FIGURE 13 A simple model of the EM dot in B-dot mode (Agry and Schill, 2014a)

EM Dot

The theory of the EM dot hinges on the change in the magnetic flux linked to the dot generating an electromagnetic force (emf) in the dot lines. The emf is shared between the two oscilloscope channels as indicated in Figure 13 and Equations (29a,b) in B dot mode (Agry and Schill, 2014a, Agry and Schill, 2014b) The current in the line generating the time varying magnetic flux is proportionally linked to the emf summed up (integrated) over time as suggested by Equations (29a,b). (Agry, and Schill, 2014a)

$$emf = v_1(t) - v_2(t) = A_d \frac{\partial B_n(t)}{\partial} \tag{29a}$$

$$i_M(t) = \frac{1}{M} \int_0^t [v_1(t) - v_2(t)]dt + i_M(0) \tag{29b}$$

As the experiment is performed, voltage data from a single dot is recorded on two channels of an oscilloscope (Tektronix Oscilloscope TDS 6604B [6 GHz Bandwidth, 20G samples/s]). Point by point in time, the channel signals are subtracted as indicated in Equation (29a). The signal is then integrated over time. This yields the uncalibrated magnetic field driving the dot in the test stand. The time varying uncalibrated magnetic field data is then transformed into the frequency domain. The calibration numbers of the dot are used to calibrate the uncalibrated time varying magnetic field signal in the frequency domain. A data acquisition and processing code was in part updated and advanced to perform these tasks. Refer to Appendices D-M. To minimize computation efforts, it was observed that the spectral signal is significantly rich in frequencies below 500 MHz. Consequently, the dot signal was corrected over the frequency range between 1 MHz and 500 MHz. The inverse Fourier transformed signal yields the corrected magnetic field signal in the test stand. Knowing the magnetic field and the dot location mounted on the current carrying wire under test, the current flowing through the carbon strip may be computed with Ampere's law assuming the wire is straight and of infinite-in-extent (or approximately long). Refer to Equation (29b).

Experimental Procedure

The test stand consists of a coaxial liquid ring resistor, a bank capacitor, two relays (SW 1 and 2) [SW – switch], two separate pulsed-power solid-state resistors (train load and an in-line current sensing resistor), carbon strip, discharge tube, and data monitoring instruments. The voltage across the discharge tube (Paschen voltage) is monitored using a high voltage capacitive probe. A 3.3 µF capacitor bank is connected to SW 1 and charged up to an initial 2.5 kV through a 1 kΩ coaxial liquid ring resistor by a 5 kV DC voltage source. This voltage is released with the aid of SW 2 through the pulsed-power resistor connected to the carbon strip and a connecting rod linking the carbon strip to the discharge tube. At a pressure of 6 Torr, the air gas between the discharge tube electrodes breaks down at about ~2 kV (or more exact 1.9 kV). This voltage is the Paschen voltage, $V_{Paschen}$. As the voltage increases beyond the Paschen voltage, the discharge tube generates a discharge current that varies with the value of the pulsed-power solid-state resistor (load). A 10 mΩ pulsed-power solid-state current sensing resistor acts like an ammeter and is used to monitor the current generated by the discharge tube. With the discharge tube acting as a self-closing switch, a non-contact EM dot (in B dot mode) is used to monitor the discharge (pulse) current. The EM dot is placed in a secured, shielded housing to avoid undesired movement of the sensor during the experiment and to shield the sensor end of the dot from detecting external electromagnetic noise. The results obtained from the EM dot and the in-line current sensing resistor are compared for consistency. The experiment was performed for different conditions of the carbon strip (normal, short, different groove depths). The experiment was set up in a manner to monitor the current flowing through the carbon strip with different load values.

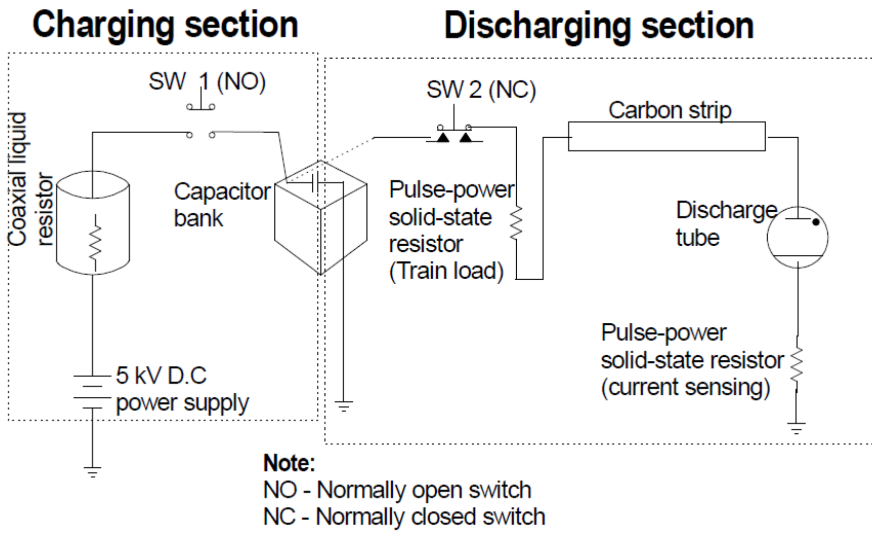


FIGURE 14 Simple Circuit Model to Visualize the Operation Sequence.

To achieve this, the following sequence of procedures is performed: (Refer to Figure 14.)

Note: SW on → Power on; SW off → Power off

Note: SW 1 on → short circuit, off → open circuit; SW 2 on → open circuit, off → short circuit

1. Both switches off
2. Power supply on (voltage set at 2.5 kV)
3. SW 2 on (open circuit – isolates the capacitor from the discharge load)
4. SW 1 on (short circuit – charges the capacitor bank)

For single pulse mode

a. SW 1 off (open circuit – disconnects the capacitor from the voltage source for a single pulse mode)
b. SW 2 off (short circuit – releases the capacitor energy to the load and breaks down the air gap in the discharge tube)

For multi-pulse mode

i. SW 2 off (short circuit – continually releases the capacitor energy to the load)
ii. SW 2 on (open circuit – isolates the capacitor from the load)

5. Power supply off

6. SW 1 and 2 off (if in on position)
7. Ground the residual energy with the shorting stick

Experimental Results of Detecting Carbon Strip Degradation with EM dots and an In-line Ammeter Monitor (Current Measuring 10 mΩ Resistor)

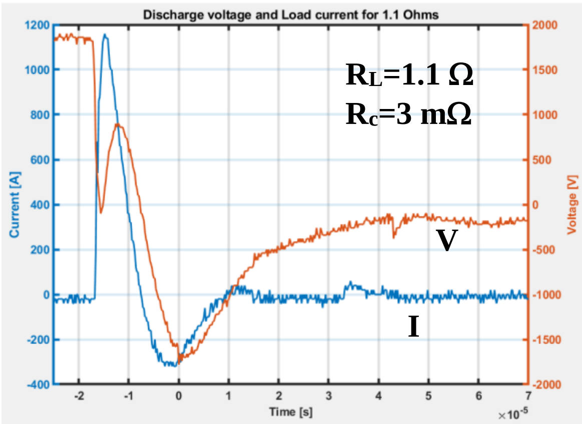
The purpose of this setion is to demonstrate that the EM dot sensor can distinguish the differences in the static carbon strip thickness. The issue regarding the size of the train load and the ability to distinguish the carbon strip thicknesses at that size with the EM dots is addressed. The EM dot in B-dot mode monitors the magnetic field generated by the current drawn by the locomotive. As the carbon strip wears, its thickness decreases. In principle, the current increases generating a stronger magnetic field. However, the carbon strip resistance range is small 3 m Ω < R_c< 28 m Ω. The EM-dot must be sensitive to changes in the magnetic field within the range of R_c. The effective train load is $\tilde{R}_L = R_L + R_W + R_M$ or $\tilde{R}_L = R_L + 0.15\Omega$.

Consequently, $R_L + 0.153\Omega < \tilde{R}_L + R_c < R_L + 0.178\Omega$. As the train load R_L exceeds 0.178 Ω, the train load R_L begins to dominate all remaining resistive effects of the locomotive and carbon strip (in the UNLV pulsed-power carbon strip degradation test stand). The peak current generated will decrease inversely proportional to increases in the train load R_L. Recall that the initial source voltage is fixed by Paschen effects and initially the discharge current $I_L \approx V_{Paschen}/(\tilde{R}_L + R_c)$ (Refer to Figures 6 and 7) if inductive effects can be neglected or assumed small. The change in the current sourced to the train over the range of R_c, will be difficult to resolve.

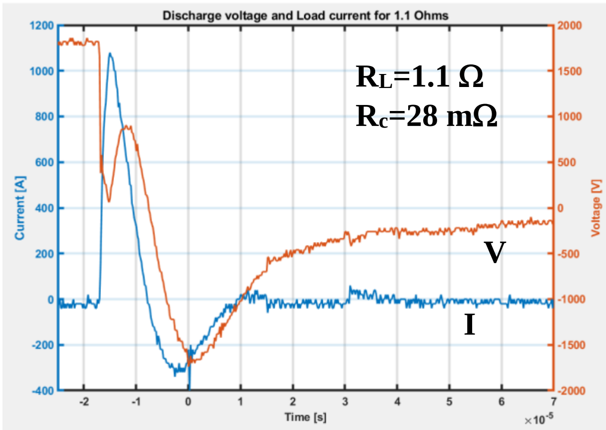
Figures 15a-h provide the discharge voltage and the input train current drawn for the train load conditions 1.1 Ω, 2.2 Ω, 5.0 Ω, and 7.2 Ω. At the lower train loads, 1.1 Ω and 2.2 Ω, it is easily obvserved that the peak currents for the shorted case (R_c=3 m Ω) is larger than for the case when the carbon strip is new without blemish (R_c=28 mΩ). At the higher train loads, the same effect is observed but the differences are much smaller. This also tends to suggest that increasing the Paschen voltage will increase the difference between the R_c=3 mΩ peak current and the R_c=28 mΩ peak current allowing one to resolve carbon strip thicknesses at larger train loads R_c beyond what has been examined here.

Figure 16a-h along with Table 7 explicitly demonstrates that the EM dot can resolve the various carbon strip thicknesses as displayed in Figure 2g and Table 1 for the train load cases 1.1 Ω (Figure 16a) and 2.2 Ω (Figure 16c). For these two cases, the current measured monotonically increases as one approaches the short condition. Further, if neighboring groove geometries are lumped together, the train loads between 2.2 Ω and 5.0 Ω may be used to distinquish a coarser carbon strip degradation. For example, when the 5 Ω train load transition from the Shallow case (411 A) to the Just Deep case (418 A) one may want to visually inspect the carbon strip since a number of other depths also measured 418 A. Refer to Table 7. The resolution for the train load case 7.2 Ω is probably too small to take advantage of. Hence, train load effects greater than 5.0 Ω will be difficult to resolve. Both the EM dot data (not highlighted; Figures 16a,c,e,g) and the in-line current sensor data (gray highlighted text; Figures 16b,d,f,h) are shown in Table 7. In general, the EM dot data and the in-line current sensor data show similar tendencies and, in a rough sense, similar current amplitudes. The data acquisition and processing code (Appendices D-M) corrects the raw data recorded from the dots and determines the magnetic field at the location of the dot. The source current driving the field can then be determined.

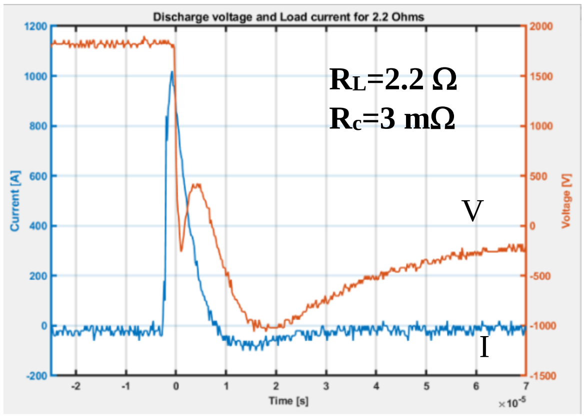
In principle it is possible to distinguish small changes from large values by evaluating the difference between the large values. Noise issues may be a problem. Consequently, the EM dot loses its ability to distinguish the differences in the carbon strip thickness for large R_L. Another possible option is to increase the train input voltage. In a pulsed power setting this is possible by increasing the Paschen voltage. In practice, this may not be as easy to implement in a railroad setting. This is discussed in greater depth in the conclusion.



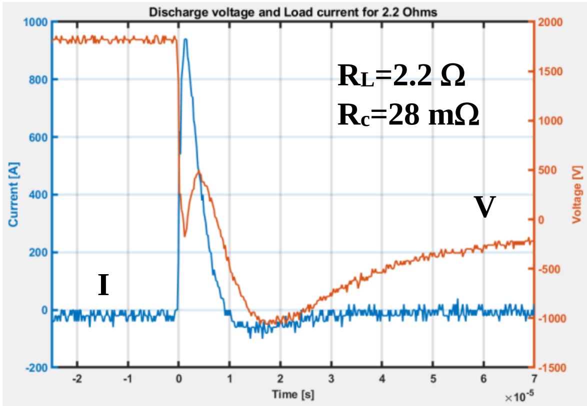
(a)



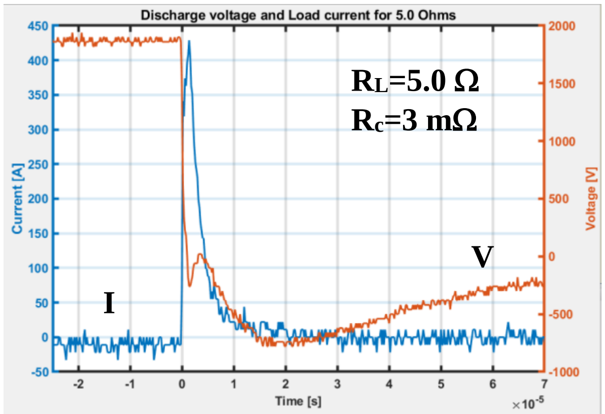
(b)

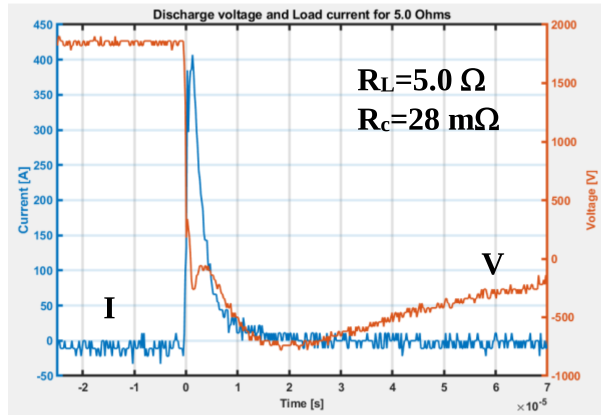


(c)

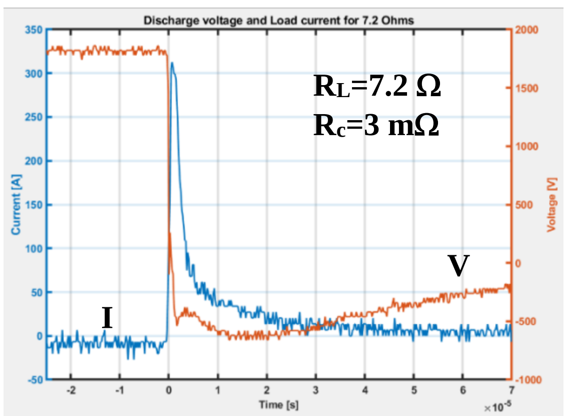


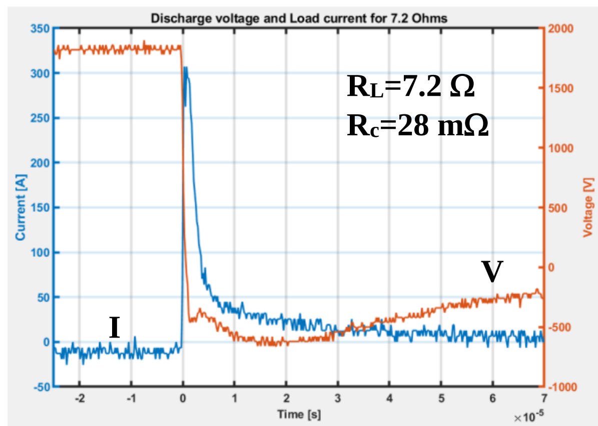
(d)





(e) (f)

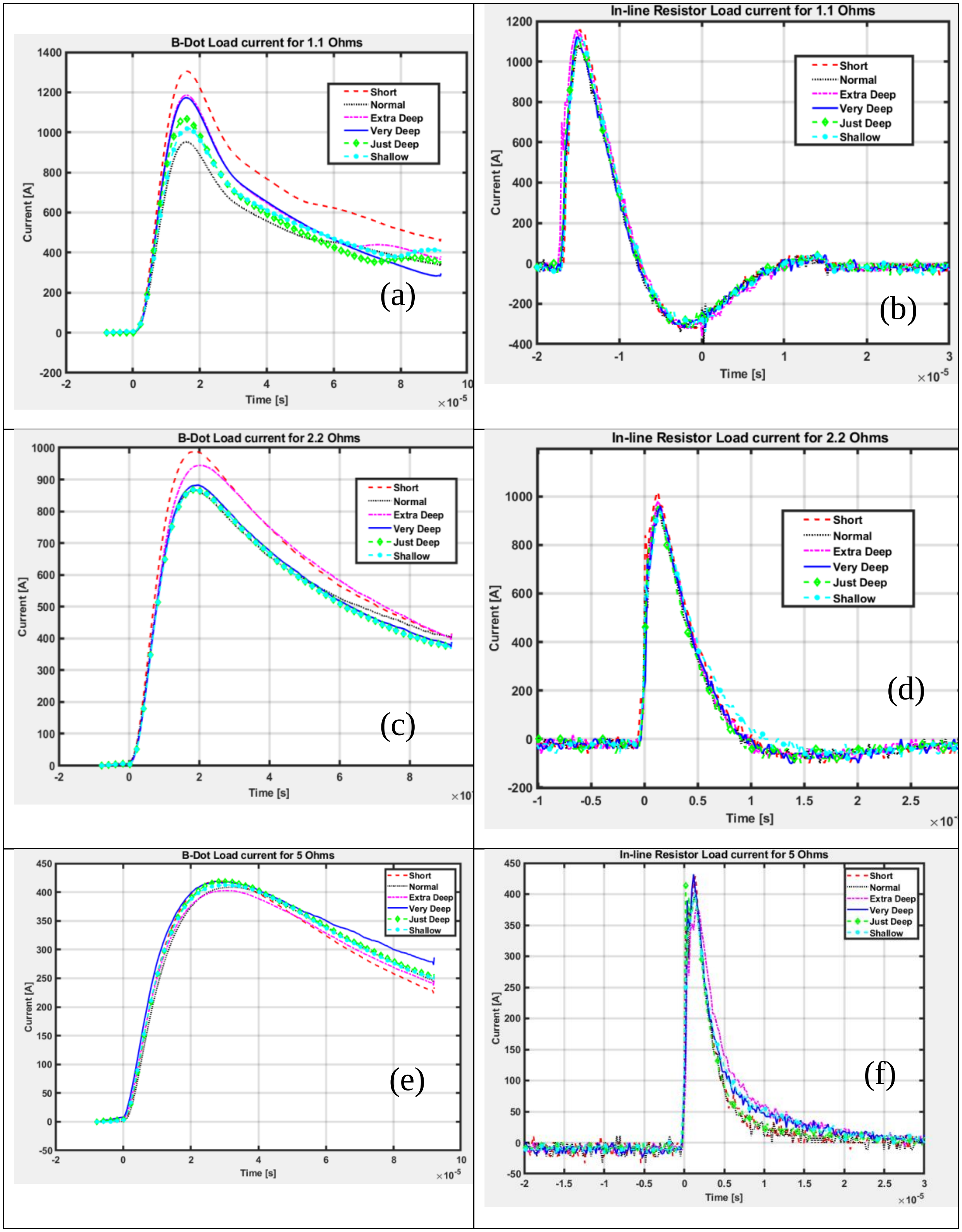




(g) (h)

FIGURE 15 The Discharge Voltage across the Discharge Tube Electrodes and the Input Current Passing through the Carbon Strip are Displayed.

The discharge voltage and input currents are displayed for a) R_L=1.1 Ω and R_c =3 mΩ, b) R_L=1.1 Ω and R_c =28 mΩ, c) R_L=2.2 Ω and R_c =3 mΩ, d) R_L=2.2 Ω and R_c =28 mΩ, e) R_L=5.0 Ω and R_c =3 mΩ, f) R_L=5.0 Ω and R_c =28 mΩ, g) R_L=7.2 Ω and R_c =3 mΩ, and h) R_L=7.2 Ω and R_c =28 mΩ. In each case, the initial discharge voltage denoted as the Paschen voltage is roughly 2 kV (more accurately 1.9 kV). The load current in the figures above are the currents measured by the in-line current measuring resistor (in-line current monitor or ammeter). These in-line currents are the same as the magnified in-line currents displayed in Figures 16b,d,f,h. for the short case (3 mΩ) and the normal unblemished case (28 mΩ). In all train load cases, the voltage changes sign in the discharge process on the time scales shown. Further for the 1.1 and 2.2 train loads, the current changes sign. For the remaining train loads investigated (5.0 and 7.2) the discharge current is single signed. The current pulse width is roughly 10 µs. Not shown, the pulse repetition period is 135 ms as shown in Figure 11d. In these figures, the train load resistance is not the equivalent train load resistance. It is the actual train load resistance. [Experimental parameters: R_W=0.14 Ω, L_m=3.32 µH, R_M=10 mΩ, R_s=1kΩ, C_b=3.3 µF, R_d=0.5 Ω, and C_d=3.16 pF. Refer to Table 2.]


B-Dot Load current for 1.1 Ohms
Short
Normal
Extra Deep
Very Deep
Just Deep
Shallow
Current [A]
Time [s]
×10^-5
(a)
In-line Resistor Load current for 1.1 Ohms
Current [A]
Time [s]
×10^-5
(b)
B-Dot Load current for 2.2 Ohms
Current [A]
Time [s]
(c)
In-line Resistor Load current for 2.2 Ohms
Current [A]
Time [s]
(d)
B-Dot Load current for 5 Ohms
Current [A]
Time [s]
×10^-5
(e)
In-line Resistor Load current for 5 Ohms
Current [A]
Time [s]
×10^-5
(f)

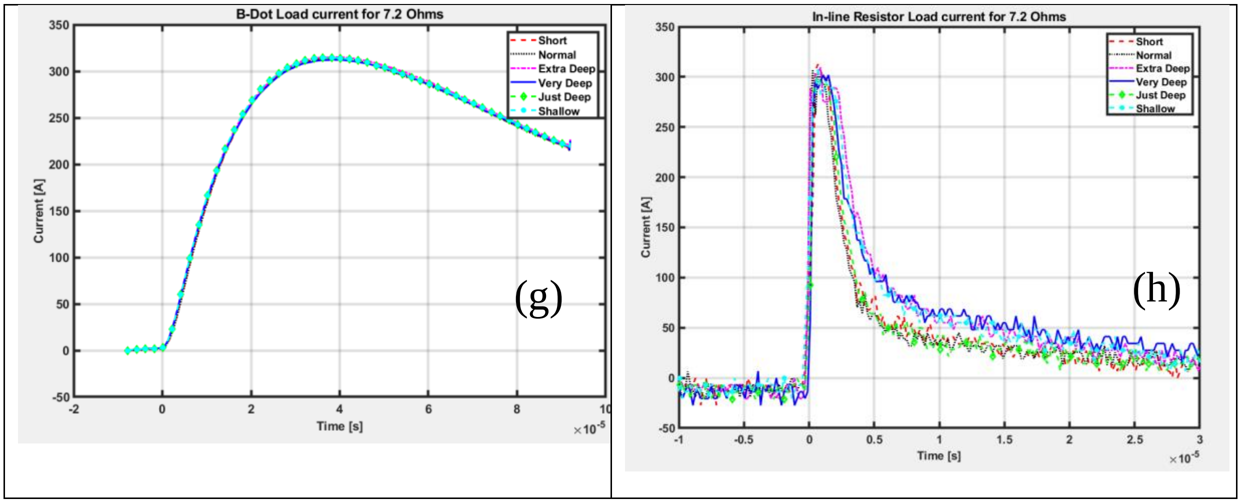


FIGURE 16 Experimental Studies of the Line Current Passing through the Carbon Strip Resistor for Various Static Groove Depths.
Refer to Figure 2g and Figure 3 based on the groove geometry given in Table 1. Refer to Table 7 for specifics. It is observed that the EM dot can distinguish the differences in the carbon strip resistances in the five grooves of different depth for the train loads between 1.1 Ω (possibly lower; not tested) and 2.2 Ω. If neighboring groove-geometries evaluated are lumped together then train loads between 2.2 Ω and 5 Ω can be used to distinguish a coarser carbon strip degradation. For a train load of 7.2 Ω the EM dot could no longer distinguish the resistance differences in the carbon strip grooves. The current detected with the in-line current monitoring resistor was noisier than that measured with the dot. [Experimental parameters: R_W=0.14 Ω, L_m=3.32 μH, R_M=10 mΩ, R_s=1kΩ, C_b=3.3 μF, R_d=0.5 Ω, and C_d=3.16 pF. Refer to Table 2.]

TABLE 7. Provides the Peak Current as Detected by the EM Dot and an In-line Current Monitor in the Pulsed-Power Carbon Strip Degradation Test Stand
The latter detector is a 10 mΩ pulsed-power solid-state resistor. For train loads between of 1.1 Ω (possibly lower: not tested) and 2.2 Ω the train load current monotonically decreases from the short state to the normal state. One can distinguish the differences in the static carbon strip thickness. For the 5.0 Ω train load, the current does not change monotonically. Further, it is difficult to resolve the different thicknesses of the carbon strip. Since the current between the short and normal case is 11 A, some gross lumped differences can be distinguished. Consequently, train loads between 2.2 Ω and 5 Ω can be used to monitor the gross carbon strip degradation. The extra deep case seems to have exceeded the range between the short and normal currents. For a train load of 7.2 Ω, there is only a 1 A difference between the short and normal currents. It is unclear if one can resolve useful information from the data in this case. The data presented below is associated with the plots in Figures 16 a-h. [Experimental parameters: R_W=0.14 Ω, L_m=3.32 μH, R_M=10 mΩ, R_s=1kΩ, C_b=3.3 μF, R_d=0.5 Ω, and C_d=3.16 pF. Refer to Table 2.]

Fig. 16	Sensor Type EM Dot (EM) or In-line Current Sensor (ICS)	Train Load [Ω]	Groove Geometry: Refer to Figs. 2g and 3 and Table 1.0 Line Current Measured with Sensor for Particular Train Load					
			Short (3mΩ) [A]	Extra Deep [A]	Very Deep [A]	Just Deep [A]	Shallow [A]	Normal (28mΩ) [A]
a	EM	1.1	1305.48	1185.48	1172.67	1066.60	1019.71	952.61
b	ICS	1.1	1160	1160	1120	1120	1120	1080
c	EM	2.2	987.52	944.68	882.63	868	868	865

d	ICS	2.2	1020	980	960	960	940	938
e	EM	5	418	402	418	418	411	407
f	ICS	5	429	432	309	413	402	402
g	EM	7.2	314	314	312	313	313	313
h	ICS	7.2	310	309	309	308	309	308

MONITORING AND END-OF-LIFE PREDICTION

Statements made in this section have not been experimentally investigated in a university laboratory or on a train in the field. The prototype field monitoring device houses two passive EM dot sensors. Only one working dot sensor is necessary for the monitoring device to perform properly. Two sensors were provided in case one dot is damaged. As the strip wears, the line current increases. As the line current changes, the magnetic field generated by the power line changes proportionately. In principle, changes in the magnetic field generated by the changes in the line current may be detected by the EM dots. From these measurements, carbon thickness, wear rate, and state curves may be constructed in-transit. Both curve generation techniques and usage will be presented in this section. Two techniques are presented on how to constuct the characteristic curves of the carbon strip. The first technique uses the fast rail in-transit for a particular set of train parameters. The second technique is based off of a continuous carbon thickness versus resistance curve formed from discrete measurements with interpolation conducted in a laboratory setting. It is assumed that the characteristic curves are monotonically continuous with varying carbon strip wear rates (varying slopes) allowed. Finally, statistical deviations from baseline characteristic curves (monitoring) and end-of-life predictions will be addressed.

Characteristic Curve Generation – Technique 1

In principle, it is anticipated that one can generate carbon thickness, wear rate, and state curves from empirical measurements on a train in-transit. There is no need for a pre-existing, discrete, static resistance versus thickness curve. In-transit it is assumed that one cannot directly measure the thickness, wear rate, and state of the carbon strip. One can monitor these parameters indirectly by measuring the current flowing in the power inlet lines. Non-destructively and non-intrusively, these measurements may be made by monitoring the change in the magnetic field generated by the current in the power line with an electromagnetic (EM) dot sensor. The initial (new state) and minimum (short state) carbon strip resistance need to be measured.

The construction of the carbon thickness, wear rate (slope of the carbon thickness versus time curves), and the state curves will be considered. Here, the state curves are generated due to discrete changes in area and conductivity of the carbon strip resistance. Other states such as temperature, moisture, pressure, train load, speed, track grade, etc. may be modeled into the state parameters $A\sigma$ or in another physical parameter.

The wear rate, W_R, is the rate of increase in the carbon thickness as given by

$$W_R(t) = -\frac{dD(t)}{dt} = -\lim_{\Delta t \to 0} \frac{D(t+\Delta t) - D(t)}{\Delta t} \tag{30}$$

where D(t) represents the distance of separation between the high voltage line and the metallic carbon strip holder. This distance is referred to as the carbon thickness. Refer to Figure 3. The limiting form of the wear rate guides one in computing the wear rate with time. When the carbon strip is new, the distance of separation is largest. At the end-of-life, the distance of separation is zero (short) or a specified operating minimum threshold value. The minus sign in Equation (30), is needed since D(t) decreases with time whereas mathematically $dD(t)/dt$ is the rate of increase in the distance of separation. Physically, D(t) is continuous and monotonically decreases with time. Assuming the wear rate is piecewise constant (Refer to Figure 17), the carbon thickness may be written as

$$D(t) = -W_{R(j+1)}(t - t_j) + D_{cj} \quad D_{c(j+1)} \leq D(t) \leq D_{cj} \tag{31}$$

where j=0,1,2, … and D_{cj} represents the thickness at time t_j corresponding to the jth state. During a particular state j where $D_{c(j+1)} \leq D(t) \leq D_{cj}$ the state parameters are a constant. At each D_{cj} both the wear rate and/or state parameters are allowed to change. In general, the thickness D(t) is continuous over all time and may change in slope at the state transitions D_{cj}. Refer to Figure 18.

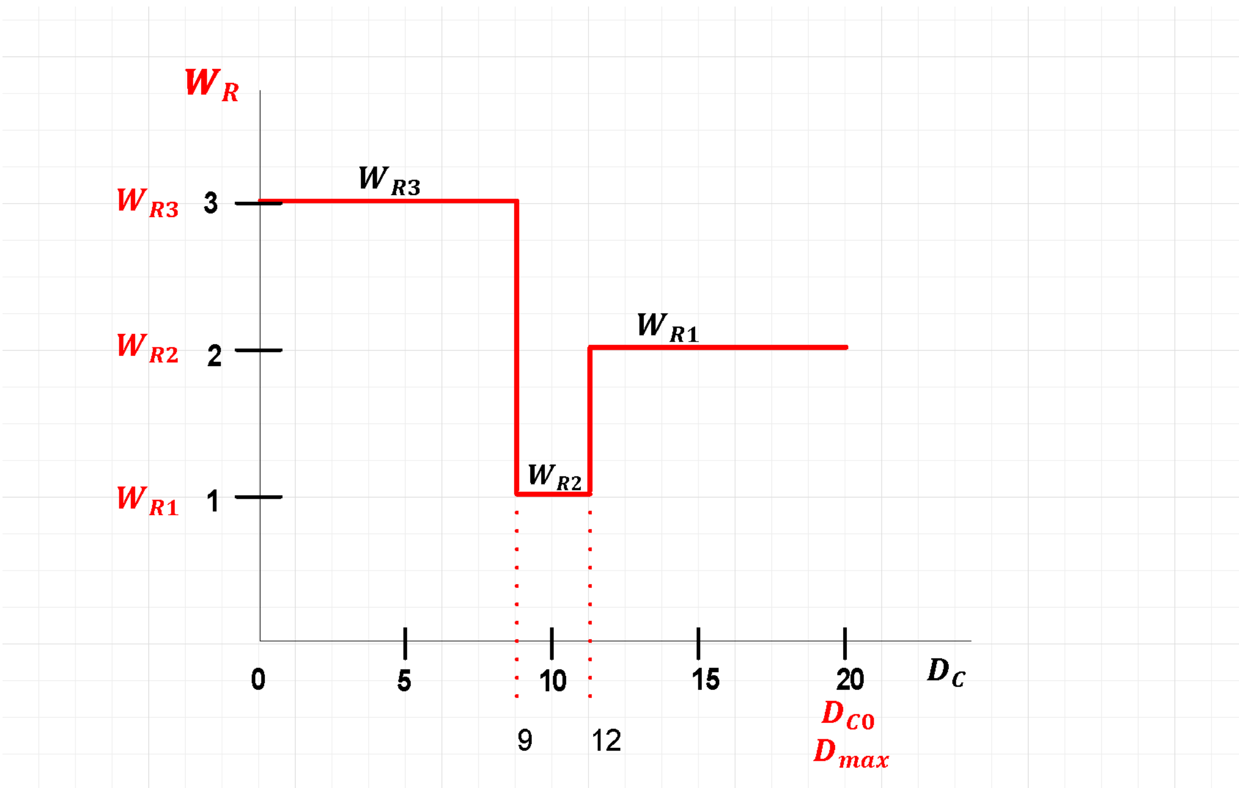


FIGURE 17 Wear Rate for a Carbon Strip Containing Three States.
The wear rate is plotted against the carbon strip thickness.

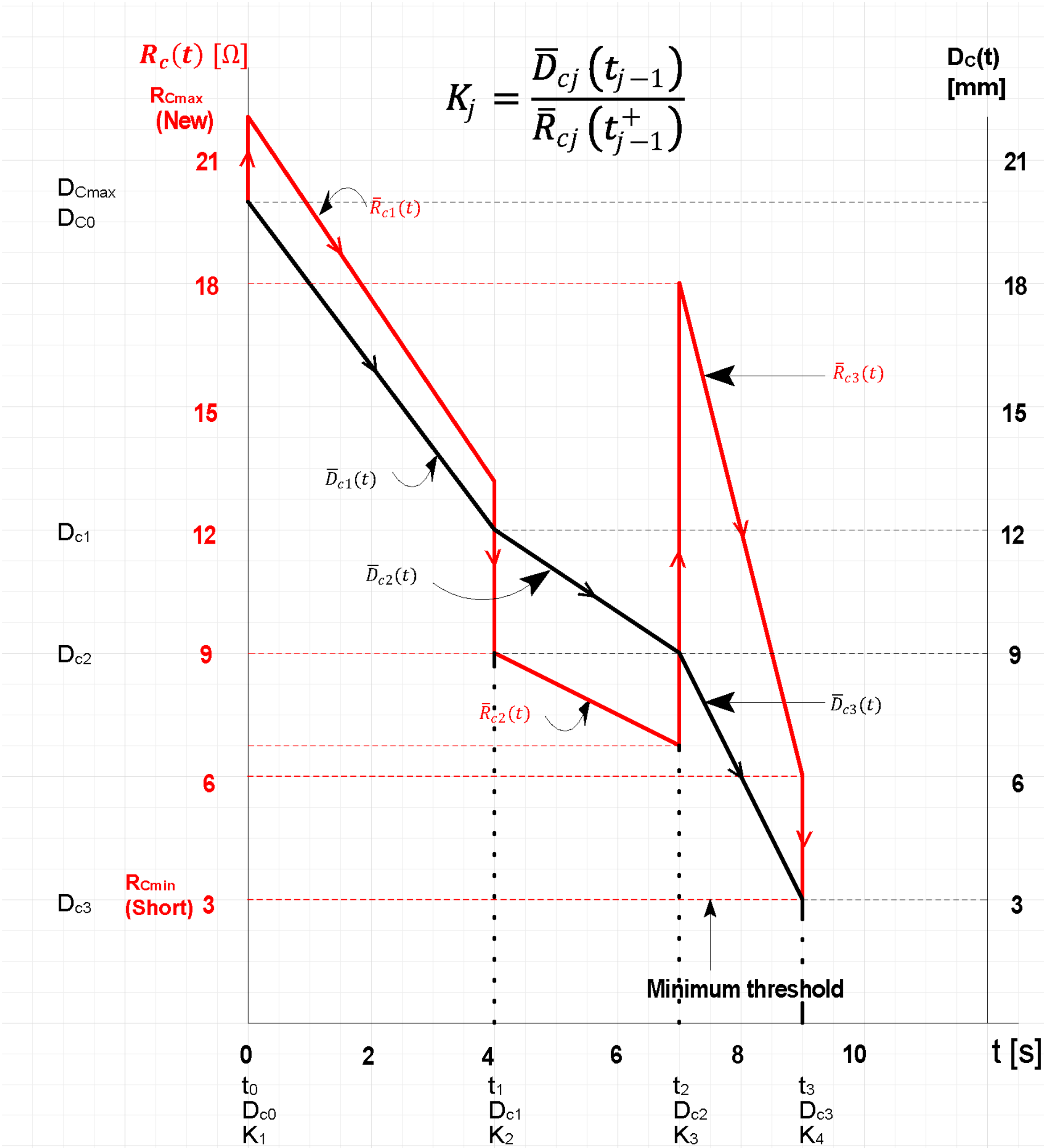


FIGURE 18 The Carbon Strip Resistance and Thickness versus Time Curves. Note that the resistance versus time is discontinuous at the state transitions. Further, just beyond the state transitions a change in the slope of the resistance occurs as well. Here, a minimum threshold is defined to be the short condition. (Drawn to scale.)

As in Figure 17 consider that during the lifetime of the carbon strip, there are three changes of state. The wear rate equations characterizing the thickness evolution from new (D_{co}) to end-of-life (D_{c3} [short]) is given as

$$D(t) = \begin{cases} \bar{D}_{c1}(t) = -W_{R1}(t - t_0) + D_{co} & D_{c1} \le D(t) \le D_{co} \quad t_o \le t \le t_1 \quad 0 \le (t - t_o) \le T_1 = t_1 - t_o \\ \bar{D}_{c2}(t) = -W_{R2}(t - t_1) + D_{c1} & D_{c2} \le D(t) \le D_{c1} \quad t_1 \le t \le t_2 \quad 0 \le (t - t_1) \le T_2 = t_2 - t_1 \\ \bar{D}_{c3}(t) = -W_{R3}(t - t_2) + D_{c2} & D_{c3} \le D(t) \le D_{c2} \quad t_2 \le t \le t_3 \quad 0 \le (t - t_2) \le T_3 = t_3 - t_2 \end{cases} \quad (32a)$$

where

$$T_1 = \frac{D_{co} - D_{c1}}{W_{R1}}, \quad T_2 = \frac{D_{c1} - D_{c2}}{W_{R2}}, \quad T_3 = \frac{D_{c2} - D_{c3}}{W_{R3}} \quad (32b)$$

The superscript bar on D represents the segmental curve of the carbon thickness curve. These relations correspond to the, for example, thickness curve given by Figure 18. Each continuous change in slope means a change in the carbon strip wear rate. Note, the carbon thickness curve is continuous and the wear rate [slope] is piecewise continuous.

At this point, the thickness information is not known. Even so, one can monitor the line current by measuring the magnetic field generated by the line current. A simplified model for the EM dot is given in Figure 13. From this figure and Faraday's law,

$$B_n(t) = \frac{1}{A_d} \int_0^t emf\,dt = \frac{1}{A_d} \int_0^t [v_1(t) - v_2(t)]dt + B_n(0) \quad (33a)$$

B_n is the magnetic field passing normal through the dot sensor area, A_d is the planar dot sensor area bounded by the sensor wires, and $v_1(t)$ and $v_2(t)$ are the voltage drops across the two output terminals of the dot. The current flowing in a straight wire can be approximated as

$$i_s(t) = \frac{2\pi a}{\mu_o A_d} \int_0^t [v_1(t) - v_2(t)]dt + i_s(0) \quad (33b)$$

Here, μ_o is the permeability of free space, and 'a' is the circular radius of the magnetic field lines around the current carrying wire threading normal through the planar dot area A_d bounded by the sensor wire in the plane. Further, $i_s(t)$ is the power line current supplied to train with carbon strip and $i_s(0)$ is the supplied current at time t=0.

Knowing the approximately constant train load resistance, R_L, the power line voltage, V_L, and the measured line current $i_s(t)$ supplied to the train, one can determine the carbon strip resistance.

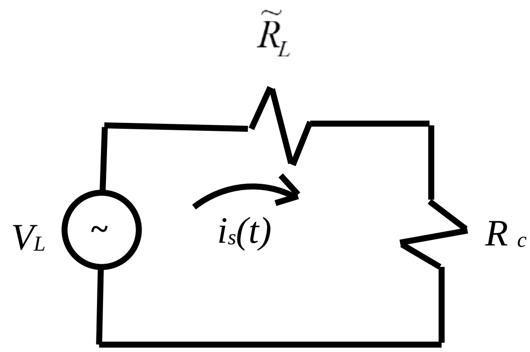


FIGURE 19 A Simple Electrical Circuit Model Characterizing the Train Load with Carbon Strip.

For the circuit above (Figure 19), the carbon strip resistance is given by

$$R_c = \frac{V_L(t)}{i_s(t)} - R_L \quad \text{or} \quad R_c = \frac{V_L(t) - R_L i_s(t)}{i_s(t)} \qquad (34a)$$

where the low frequency definition for the carbon strip resistance is

$$R_c(t) = {}^{D_c(t)}/_{\sigma A} \qquad (34b)$$

where σ is the conductivity of the resistor and A is the cross sectional area of the resistor in which the current passes normal through. Note that σ and A or the combination σA are the state parameters. Consider that there are three discrete states in this example. Therefore, the following equations

$$R_c(t) = \begin{cases} \bar{R}_{c1}(t) = -\frac{W_{R1}}{A_1\sigma_1}(t - t_0) + \frac{D_{co}}{A_1\sigma_1} & D_{c1} \le D(t) \le D_{co} \quad t_o \le t \le t_1 \quad 0 \le (t - t_o) \le T_1 = t_1 - t_o \\ \bar{R}_{c2}(t) = -\frac{W_{R2}}{A_2\sigma_2}(t - t_1) + \frac{D_{c1}}{A_2\sigma_2} & D_{c2} \le D(t) \le D_{c1} \quad t_1 \le t \le t_2 \quad 0 \le (t - t_1) \le T_2 = t_2 - t_1 \\ R_{c3}(t) = -\frac{W_{R3}}{A_3\sigma_3}(t - t_2) + \frac{D_{c2}}{A_3\sigma_3} & D_{c3} \le D(t) \le D_{c2} \quad t_2 \le t \le t_3 \quad 0 \le (t - t_2) \le T_3 = t_3 - t_2 \end{cases} \qquad (39)$$

describe the lifetime of the carbon strip in terms of its carbon strip resistance value. For clarity, as observed in Equation (34a), the carbon resistance curve(s) is based on the source current as a function of time. The EM dots are instrumental in measuring this parameter Equation (33b). Once the carbon resistance curves are constructed, one can determine the carbon strip thickness curve. The superscript bar on R_c represents the segmental portion of the carbon strip resistance curve. Since D(t) is continuous, the carbon resistance is piecewise continuous since $R_c(t) = D(t)/(\sigma A)_j$ is dependent on the state as illustrated in Figures 17 and 18. Define K_j to represent the state constant at the initial point in time of the jth segment on the thickness and resistance characteristic curves as

$$K_j = \frac{\bar{D}_{cj}(t_{j-1})}{\bar{R}_{cj}(t_{j-1}^{+})} = (\sigma A)_j \tag{36}$$

where j=1, 2… . The bar implies the jth segment on the characteristic curves as displayed in Figure 18. Graphically this implies that the $R_c(t)$ versus time curve may be used to 1) generate the carbon thickness vs time curve, and 2) identify where the changes in state occur relative to the thickness of the carbon strip as identified by D_{cj} (discrete values). Each discontinuous jump implies a transition from one state to another. To recover the thickness curve, adjustments are made at the beginning of each state including the initial state where $R_{c\,max}$ is known. A continuous change in state may also occur. This implies that the slope of the resistance and carbon thickness curve changes or, equivalently, the wear rate changes.

Consider Figure 18 that contains both the thickness characteristic curve and the resistance characteristic curve of the carbon strip. The $R_c(t)$ versus t curve is constructed with the aid of dot measurements in-transit, Equations (29a,29b,33a,33b), and Equations (34a,b). Making use of Equation (36), the thickness curve versus time can be generated. Discontinuous jumps in the carbon resistance curve is a sign of a discrete change in state. Note at t=0 in Figure 18, the maximum carbon resistance and maximum carbon thickness are known and plotted. The state constant, K_1 [m/Ω], can be determined from Equation (36) where

$K_1 = \bar{D}(t = t_o = 0)/\bar{R}_{c1}(t = t_o^{+} = 0^{+}) = D_{c\,\max}/R_{c\,\max}$. Then, point by point for $t_o<t<t_1$, the resistance curve $\bar{R}_{c1}(t)$ is adjusted using $\bar{D}_{c1}(t) = K_1\bar{R}_{c1}(t)$. The adjusted resistance curve is now the carbon thickness curve for $t_o<t<t_1$. The resistance curve $\bar{R}_{c2}(t)$ for $t_1<t<t_2$ is adjusted in the same manner. First the new state constant is determined at the transition time $t=t_1$ where $K_2 = \bar{D}_{c2}(t = t_1)/\bar{R}_{c2}(t = t_1^{+})$. With K_2 known, then $\bar{D}_{c2}(t) = K_2\bar{R}_{c2}(t)$. The adjusted resistance curve yields the thickness curve for $t_1<t<t_2$. For completeness sake, $K_3 = \bar{D}_{c3}(t = t_2)/\bar{R}_{c3}(t = t_2^{+})$. The thickness curve is $\bar{D}_{c3}(t) = K_3\bar{R}_{c3}(t)$ for $t_2<t<t_3$. At $t = t_3^{-}$ the adjusted resistance point R_{cmin} yields D_{cmin}. The carbon thickness lifetime curve for a set of train parameters has been constructed from the carbon resistance curve based on a DC model of the carbon resistance. Table 8 shows the equations and key parameters for Figure 18.

With the carbon thickness curve known, the wear rate in each state can be determined with Equation (30). In particular, the wear rate is the slope of each line segment on the characteristic curve for carbon thickness. It is anticipated that wear rates may be uniform if the speed of the train is held constant, environmental conditions are the same, the grade of the tacks are uniform, the overhead power line and carbon strip are in ideal condition, the applied pantograph force is uniform, and the train's electrical and mechanical load (weight, brake usage, etc.) are constant and the same. The wear rates under most conditions will differ and can be monitored with the EM dot sensor in-transit.

TABLE 8 Parameters and Equations Needed for Figure 18

This figure is used to generalize how to construct the carbon thickness and resistance curves while in-transit. The data does not represent experimental data generated from either the UNLV degradation test stand or an electric train.

j	Time Range (s)	Wear Rate W_{Rj} (mm/s)	Carbon Thickness Transition Points $D_{c(j-1)}$ (mm)	State Constant K_j (mm/Ω)	Carbon Resistance or Thickness Equations (32a,b) and (35) D_c ---- (mm) R_c ---- (Ω)
1	$0 < t < 4$	2	20		$D_c(t) = -2(t-0) + 20 = \bar{D}_{c1}(t)$
2	$4 < t < 7$	1	12		$D_c(t) = -1(t-4) + 12 = \bar{D}_{c2}(t)$
3	$7 < t < 9$	3	9		$D_c(t) = -3(t-7) + 9 = \bar{D}_{c3}(t)$
1	$0 < t < 4$	2	20	0.9091	$R_c(t) = D_c(t)/K_1 = -2.2(t-0) + 22 = \bar{R}_{c1}(t)$
2	$4 < t < 7$	1	12	1.333	$R_c(t) = D_c(t)/K_2 = -0.75(t-4) + 9 = \bar{R}_{c2}(t)$
3	$7 < t < 9$	3	9	0.5	$R_c(t) = D_c(t)/K_3 = -6(t-7) + 18 = \bar{R}_{c3}(t)$

Characteristic Curve Generation: Technique 2

In the second technique, it is assumed that there are no discontinuous state changes which should be the case for a conventional carbon strip under ideal conditions. Continuous state changes (changes in $\bar{R}_{cj}$ and $\bar{D}_{cj}$ segment slopes) are allowed. Consequently, both the carbon thickness and the carbon resistance curves will be continuous and monotonically decreasing. In this technique, the carbon thickness curve is based on select, discrete, static resistance measurements versus thickness performed in a laboratory setting.

In-transit, the EM dot will be used to measure the $emf = v_1(t) - v_2(t)$. Equations (29a,29b,33a,33b) link the electromotive force to the source line current powering the locomotive. This is the current passing through the carbon strip. Formulated from Figure 19 and Equation (34a), the carbon resistance can be determined given the train load R_L and the power line voltage. The carbon strip resistance can be related to a carbon strip thickness based on the static resistance and corresponding thickness measurements performed in a laboratory setting. With the aid of interpolation, laboratory measurements may be compiled in the form of a continuous curve in a plot or, equivalently, numbers in a look-up table.

For a locomotive in-transit, the carbon strip will degrade with time. Recording the carbon strip thickness with time one can determine the wear rate as given by Equations (30, 31) for a particular set of train parameters or state.

Monitoring and Predictions

At this point, the characteristic carbon resistance, carbon thickness, and carbon wear rate curves are known. In this section, the usage of these curves to determine when the carbon strip needs to be replaced will be addressed. In-transit monitoring will allow for corrections and iterations in predictions as the end of life of a carbon strip approaches especially when the wear rate is abnormal due to unexpected faults.

The characteristic curves are assumed to be generated for the practical carbon strip under typical locomotive operating parameters in a typical environment. That is, the practical strip is the average strip with its blemishes when released for public and private use. A typical environment includes: track grade, temperature changes, etc. When generating the characteristic curves one must consider an average set of typical locomotive operating parameters. Statistical techniques should also consider that the carbon strip product may be used on different lines yielding entirely different environmental conditions.

To use the carbon strip for monitoring and prediction purposes, either a complete time operating history of the carbon strip ($D_c(t)$, $R_c(t)$, t) or select points of the time history ($D_c(t=t_f)$, $R(t=t_f)$, t_f) AND some of the carbon strip characteristics (R_{cmin}, R_{cmax}, D_{cmin}, D_{cmax}) are required. The time t_f is the total time duration of use up to the present time, $D_c(t=t_f)$ and $R_c(t=t_f)$ are respectively the carbon thickness and the carbon resistance at $t=t_f$. The time history $t = t_f$ will pinpoint which state the carbon thickness operating point is in. The minimum and maximum characteristics are needed to make minor adjustments or recalibrations to the existing characteristic curves. Common working thresholds need to be identified on the characteristic curves as displayed in Figure 18.

It is assumed that the baseline generated characteristic curve is based on a statistically averaged set of empirical curves considering a range of train parameters and strip make-up. If the monitored data nearly fits the baseline curves, then the carbon strip is degrading in an acceptable manner. The end-of-life time and distance for the carbon strip may be computed based on the characteristic curve directly. Respectively, let T_P and T_F represent the present carbon strip usage time and the overall carbon strip usage time where $t_{n-1} < T_p < t_n$ for n=1, 2, …, N. N is the total number of states of the carbon strip. The remaining train usage time is $T_F - T_P$. The remaining end-of-life carbon strip thickness is $D_{EoL} = \left(t_n - T_p\right)W_{Rn} + \sum_{j=n+1}^{N} T_j W_{Rj}$ where T_j is defined in Equation (32a). Given the average speed of the locomotive, one can estimate the number of train miles the strip will last at some average speed: $L_{train} = v_{ave}(T_F - T_P)$.

Consider the monitored data deviates significantly from the baseline characteristic curves. If the monitored carbon strip resistance lies within the predefined safety threshold limits, an approximate end-of-life time duration and remaining operating distance can be determined given an average wear rate. In this situation one assumes that there is only one state. Based on Equation (30), real time data as measured by the EM-dot can be used to determine an average wear rate. The remaining train usage time T_F-T_P is known based on the strips past usage history to present. The remaining end-of-life estimate of the carbon strip thickness is $D_{EoL} \approx (T_F - T_P)W_{ave}$. Knowing the train velocity, the railroad engineer can calculate the approximate distance the train can travel prior to the carbon strip's end-of-life when maintenance is needed. The average wear rate in one interval of the time may differ in a different interval of the time. An estimated carbon strip history needs to be kept and updated for this particular strip. This will allow the train engineer to make informed decisions on how to use the remainder of the carbon strip. Or, to optimize the usage of the strip to reach maintenance locations or have maintenance engineers come to the train at a predetermined location.

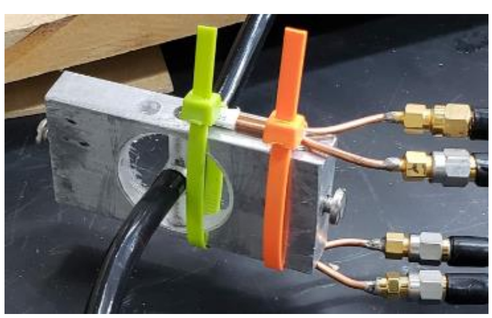
If the monitored carbon strip resistance extends beyond predefined safety thresholds, the train should be stopped and the carbon strip replaced. If the carbon strip has a serious fault, the carbon strip wear rate will statistically deviate from some statistical norm. Further if the carbon strip intermittently loses contact with the overhead power line, a ‘scratchy’ (noisy) current surge in the locomotive’s input power line will be observed upon contact. This is a result of electrical arcing occurring in the air region between the power line and the carbon strip. Arcing may be visible at the point and time of contact. If the wear rate’s statistical deviation is outside of some accepted bounds or if the dot monitored input power is abnormally scratchy (noisy) with intermittent off states, then it would be prudent to have the carbon strip examined or changed immediately.

Other applications are also feasible. For example, a carbon strip manufacturer may want to optimize their carbon strip designs for fast or slow rail. The field monitoring device would be a useful tool for such a study.

List and Pictures of Deliverables

The following is a list of hardware deliverables:

1. One early prototype monitor – houses two EM dots based on a coaxial cable configuration. The housing consists of a plastic disk with power inlet wire on axis. The plastic disk is embedded within a metal ring. The dot sensor is not shielded. The dots are not permanently mounted in this holder. This monitor has been used in all experiments and verification studies. Refer to Figure 20a.
2. One, more recent, prototype housing two EM dot sensors permanently mounted in the dot holder. The holder can be easily installed over a variety of cable diameters with or without the use of crushable material to grip the wire to prevent sliding along the cable. Refer to Figure 20b. This unit has been used for data verification purposes of the dots used in experiments.
3. A sample stripline/microstrip EM dot sensor without connectors. Refer to Figure 20c.



(a)

(b)

(c)

FIGURE 20 Picture of Deliverables Provided to the Program Manager:
a) first prototype of coaxial cable based magnetic field monitor using EM dot sensor, b) more recent prototype of coaxial cable based magnetic field monitor using EM dot sensor, and c) EM dot sensor based on stripline/microstrip technology.

The calibration numbers versus frequency for the four labelled dots based on a coaxial configuration mounted in the proto-type holders are displayed in Table 9. The data acquisition and processing code (Appendices D-M) corrects the raw data recorded from the dots and determines the magnetic field at the location of the dot.

TABLE 9 Calibration Numbers for the Dots Supplied to the Program Manager Used in this Research Effort

Frequency (MHz)	Calibration Number			
	RR01a	RR01b	RR3	RR4
1	22170.5533	15004.4984	23946.9	788.5209
10	26363.8059	23518.6867	55139.94	65578.1989
20	25791.5476	20323.7026	50999.07	45578.1989
50	37139.2702	29081.68	65873.3	36897.9
100	34687.732	27915.0252	38488.9	37149.4499
200	109887.485	77355.6154	84748.4	110783.6159
300	50621.3894	45779.9586	76930.4	59309.4562
400	195419.3431	138163.2726	262519.3	15922.3064
500	215449.6005	179303.4245	327671.5	197505.7931
600	178561.3603	152746.2301	202657.5898	190783.6159
700	127953.4529	113323.9547	100996.1464	129684.7577
800	95645.3789	85318.9482	104699.9236	130798.7379
900	76335.4351	61129.7494	80176.7915	82327.607
990	78380.6264	62368.6603	78025.3593	98573.6418

CONCLUSION/SUMMARY/RECOMMENDATIONS

A pulsed-power carbon strip degradation test stand was modeled, designed, built, and thoroughly tested with two (electromagnetic) EM dots (similar sensor corroboration check), an in-line current monitoring pulsed-power resistor (different sensor corroboration check) technique, an independent theoretical verification study implemented with MATLAB, and an independent simulation verification study with a computer aided design and analysis code LTSpice. Agreement was shown among all four checks. Under appropriate conditions, the EM dots have been able to resolve various degrees of carbon strip degradation. Back-of-the-envelope calculations and simpler experimental tendencies have been used to identify an approximate parameter window characterizing the carbon strip from its new state to its terminal or short state. The resolution of the change in carbon thickness decreases with increases in the train load.

The pulsed-power carbon strip degradation test stand employs a self-firing discharge tube to generate both single and multiple pulses with a short pulse width and a large repetition time. The EM dots respond to the change of the magnetic field generated by the source current. As the pulse configuration approaches a DC case, the EM dot no longer responds to the discharge tube's signal signature. The Paschen voltage is responsible for initializing the discharge in the gas medium between the discharge tube's electrodes. The locomotive (train), the test stand, and the carbon strip load down the discharge network drawing a load current. If the combination of train load and test stand load is large compared to the overall range of the carbon strip resistance, it becomes difficult to resolve changes in the load current and the carbon strip thickness. For the pulsed – power carbon strip degradation test stand, a range of train loads have been identified that can discriminate the six discrete carbon thicknesses examined.

It has been demonstrated by monitoring the supply input current, one can distinguish one static carbon thickness from another static carbon thickness for an appropriate range of train loads. Since the electrical properties of the carbon strip degradation test stand is similar to the electrical properties of a moving train, then one concludes that it is possible to measure the degradation of the carbon strip in-transit under appropriate conditions. Armed with wear rates and carbon strip characteristic curves it is possible to estimate the end-of-life of the carbon strip. Given the average speed, one can estimate the number of remaining train miles the strip will last.

Consider the discharge circuit provided in Figure 6. Typical electric trains (electric locomotives) are sourced with a DC or an AC (50 Hz, 60 Hz) high voltage line. Consider replacing the source resistance R_s with a short, the capacitor bank C_b with an open circuit, and the discharge tube C_d – R_d parallel circuit combination with a short. Neglecting inductive effects and combining all resistive effects into the train load resistance $\tilde{R}_L$ yields the simplified circuit model given in Figure 19. The source current drawn from the DC voltage source generates the external magnetic field. Assume the change in the magnetic field is dependent directly on the wearing down of the carbon strip. The time scales in this case are orders of magnitude slower relative to the pulsed-power excitation technique. As a result, it is not clear if the EM dots or other electromotive force (EMF) dependent techniques will be able to detect the signals much less detect changes in the carbon thickness. In this case low frequency magnetic field detectors such as the Hall effect detectors may need to be used. Outside of what voltage source is used, the model in Figure 19 is still valid. The interpretation of the physics regarding the degradation properties when using a DC voltage source and the pulsed power voltage source are the same. In particular when $\tilde{R}_L \gg R_{c\,\max}$, the line (or load) current $I_L \approx V_{Line}/(\tilde{R}_L + R_c) \approx V_{Line}/\tilde{R}_L$. The line (load) current and its generated magnetic field are no longer sensitive to changes in the carbon resistance. Consequently, one cannot measure the change in state of the carbon strip. These same arguments and limitations are valid for trains sourced with AC.

If one chooses to apply the pulsed-power technique in practice, then one would have to couple a large pulsed power signal to the electrical line. This condition could be relaxed if the train itself housed the pulse-power hardware using the overhead power line sourcing the train also source the pulse generator. One can then control the properties of the pulse (pulse duration, pulse period, pulse magnitude, etc.) being generated. One would not have to rely on an incompatible pulse generated at some distant power station. Depending on the conditions of the line and the environment, the pulse generated from the power station could become highly attenuated and highly dispersive in the vicinity of the train.

Since the dot does not respond to DC or, approximately, AC (60 Hz) signals, the dots act as a natural filter to respond only to the coupled high frequency components of the probing signal. On train time scales, microsecond pulses of high voltage interruption for a matter of seconds every half hour should not significantly affect the overall state of the train. The probing current signature will under appropriate conditions yield similar results as in this effort as long as the probing signal is not current limited. Although slower, one can produce a similar effect by lowering and raising the pantograph. Noise issues and arcing from connecting/disconnecting the power line may act as a suitable, time varying signal to interrogate the state (condition) of the carbon strip.

It is recommended that the different stages of this research be tested on an electric train that has train loads similar to and beyond the train loads used in this research. The latter case will demonstrate what limitations exist.

ACKNOWLEDGMENTS

Rocky Gonzalez (MS candidate) is acknowledged for his assistance in building and drafting various parts of the UNLV carbon strip degradation test stand. Rocky has also assisted in the design and calibration of the initial microwave/stripline based EM dots used at the onset of this research endeavor.

The USDOT University Transportation Center on Railroad Infrastrcture at UNLV is acknowledged and appreciated for supporting one PhD candidate, one MS candidate, and one undergraduate student under grant number 69A3551747132.

The research group is in debt to Schunk Inc. for supplying a commercial carbon strip. Type SK1514-SK85 in the year 2019. This helped lay down the foundation of the research effort.

The papers generated from this report will be in partial fulfillment for the Doctor of Philosophy degree for Olufemi Olubode (PhD candidate).

REFERENCES

1. Yang, S., Z. Liu, H. Ouyang, H. Wang, and X. Lu. Sliding Mode Control with PD Sliding Surface for High-Speed Railway Pantograph-Catenary Contact Force Under Strong Stochastic Wind Field. *Shock and Vibration*, Vol. 2017, 2017, pp. 1-16.
2. Xin, T., C. Roberts, P. Weston, and E. Stewart. Condition Monitoring of Railway Pantographs To Achieve Fault Detection And Fault Diagnosis. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, Vol. 234, No. 3, 2020, pp. 289-300.

3. Wu, G., J. Wu, W. Wei, Y. Zhou, Z. Yang, and G. Gao. Characteristics of the Sliding Electric Contact of Pantograph/Contact Wire Systems in Electric Railways. *Energies*.; Vol. 11, No. 1: 17. 2018
4. Wei, W., Y. Song, Z. Yang, G. Gao, P. Xu, M. Lu, C. Tu, M. Chen, and G. Wu. Investigation of the Impacts of Thermal Shock on Carbon Composite Materials. *Materials*. 2019; 12, No. 3: 435.
5. Yang, K., J. Peng, C. Jiang, X. Jiang, L. Xiao, B. Wang, X. Gao, L. Xie, and H. Peng. Design of the Fall-Block Sensing of the Railway Line Pantograph Based on 3D Machine Vision Sensors. *Sensors*. Vol. 18, No. 7: 2305, 2018.
6. Becker, K., A. Rukwied, W. Zweig, and U. Resell. Simulation of the Wear Behaviour of High-Speed Over-Head Current Collection Systems. *Transactions on the Built Environment*, Vol. 19, 1996, pp. 281-290.
7. Jarek, R. Pantograph Carbon Strip Wear Detection System. UK Research & Innovation, Abstract, Funded Period: June 18 2018-May 19, 2019. https://gtr.ukri.org/projects?ref=133502
8. Judek, S., and L. Jarzebowicz. Analysis of Measurement Errors in Rail Vehicles' Pantograph Inspection System. *Elektronika Ir elektrotechnika*, Vol. 22, No. 3, 2016, pp. 20–23.
9. Bruni, S., G. Bucca, M. Carnevale, A. Collina, and A. Facchinetti. Pantograph–Catenary Interaction: Recent Achievements and Future Research Challenges. *International Journal of Rail Transportation*, Vol. 6, Issue 2, 2018, pp. 57-82.
10. Agry, A. A., and R. A. Schill, Jr., Calibration of Electromagnetic Dot Sensor – Part 1: B-Dot Mode. *IEEE Sensors Journal*, Vol. 14, No. 9, 2014, pp. 3101-3110. (2014a)
11. Agry, A. A., and R. A. Schill, Jr., Calibration of Electromagnetic Dot Sensor – Part 2: D-Dot Mode. *IEEE Sensors Journal*, Vol. 14, No. 9, 2014, pp. 3111-3118. (2014b)
12. Alnuman, H., D. Gladwin, and M. Foster. Electrical Modelling of a DC Railway System with Multiple Trains. *Energies*, Vol. 11, Issue 11, 2018, pp. 1-20.
13. Chymera, M. Z., A. C. Renfrew, M. Barnes, and J. Holden. Modeling Electrified Transit Systems. *IEEE Transactions on Vehicular Technology*, Vol. 59, 2010, No. 6, pp. 2748-2756.
14. Ciccarelli, F., D. Iannuzzi, and D. Lauria. Supercapacitors-based Energy Storage for Urban Mass Transit Systems. *Proceedings of the 2011 14th European Conference on Power Electronics and Applications*, 2011, pp. 1-10.
15. Mersen USA Bn Corp., *Current Collection Technical Guide.* https://www.mersen.com/uploads/tx_mersen/17-current-collection-technical-guide-mersen_03.pdf pp. 5. Accessed April 23, 2021.
16. Razmjou, S., and A. Younesi. A Comprehensive DC Railway Traction System Simulator Based on MATLAB: Tabriz Line 2 Metro Project Case Study. *Journal of Operation and Automation in Power Engineering*, Vol. 9, Issue 2, 2021, pp.144 – 159.
17. Seo, S., Y. H. Cho, J. Y. Mok. and C.-S. Park, A Study on the Measurement of Contact Force of Pantograph on High Speed Train. *J Mech Sci Technol,* Vol. 20, 2006, pp. 1548–1556.
18. Koyama T., M. Ikeda, K. Nakamura, S. Tabayashi, and M. Niwakawa, Measurement of the Contact Force of the Pantograph by Image Processing Technology. *Quarterly report of RTRI*, Vol. 55, No. 2, 2014, pp. 73–78.
19. Usuda, T. The Pantograph Contact Force Measurement Method in Overhead Catenary System. Presented at World Congress on Railway Research (WCRR), Seoul, 2008.

APPENDICES

Appendix A: (MATLAB) Carbon Strip Degradation Test Stand Verification Theory 1 (Carbon Resistance Independent of Time)

```
clc,close all,clear all
tic
% circuit parameters
Vs = 2.50e3; Rs = 1000; Cb = 3.3e-6; RL = 10; Rc = 28e-3; Lm = 3.32e-6;
Cd = 3.16e-12; Rd = 0.5; % discharge tube elements
Cbd = Cb*Cd/(Cb+Cd); Tbs = Rs*Cb; T_mlc = Lm/(RL+Rc); Tdd = Rd*Cd;
% w_mbd, w_md and w_mb squared have been written without the square
w_mbd = 1/(Lm*Cbd); w_mb = 1/(Cb*Lm); w_md = 1/(Cd*Lm);
A1 = 1/T_mlc + 1/Tbs; A2 = w_mbd + 1/(Tbs*T_mlc); A3 = (w_mbd-w_mb)/Tbs;
E1 = 1/T_mlc + 1/Tbs + 1/Tdd;
E2 = w_mbd + 1/(Tbs*Tdd)+(1/T_mlc)*(1/Tbs+1/Tdd);
E3 = w_mbd*(1/Tbs+1/Tdd)+1/(T_mlc*Tbs*Tdd)-w_mb/Tbs-w_md/Tdd;
E4 = 1/(Tbs*Tdd)*(w_mbd-w_mb-w_md);

% p = [1 A1 A2 A3]; % coefficients of s^3 + A1*s^2 + A2*s + A3 = 0
% s = roots(p);
% s1 = s(1,1); s2 = s(2,1); s3 = s(3,1); % corresponding to s1, s2, s3
% q = [1 E1 E2 E3 E4]; % coefficients of s^4 + E1*s^3 + E2*s^2 + E3*s + E4=0
% v = roots(q);
% v1 = v(1,1); v2 = v(2,1); v3 = v(3,1); v4 = v(4,1);  % corresponding to
s1_bar, s2_bar, s3_bar, s4_bar

% Q = (3*A2-A1^2)/9; R=(9*A1*A2-27*A3-2*A1^3)/54;
% S =(R+sqrt(Q^3+R^2))^(1/3); T=(R-sqrt(Q^3+R^2))^(1/3);

%% Third order ODE to be solved for charging
time = [-0.7e-5 0]; % time range
Ito = w_mb*(Vs/Rs); % initial current
syms y(x) x Y
D1y = diff(y,x);
D2y = diff(y,x,2);
D3y = diff(y,x,3);
Eqn3 = D3y + A1*D2y + A2*D1y + A3*y == 0;
[VF3,Subs3] = odeToVectorField(Eqn3);
IC = [0;0;Ito]; % initial conditions
RRcurent3 = matlabFunction(VF3, 'Vars',{x,Y});
[x,y] = ode45(RRcurent3,time,IC);
% plot(x,y),grid
% legend(string(Subs3))

%% Fourth order ODE to be solved for discharging
time1 = [-1e-7 150e-6]; % time range
% Ito1 = 138.89e-3; Ito2 = 0; Ito3 =196.42; Ito4 = 1428753;
% Ito1 = 6.83e-3; Ito2 = -7.8162; Ito3 = -1.51e10; Ito4 = 2.79e21;
Vbo = 2.50e3; Vdo = 2.48e3;
Ito1 = (Vbo-Vdo)/(RL+Rc); Ito2 = (Vbo-Vdo)./Lm-((RL+Rc)/Lm)*Ito1;
Ito3 = -Ito2/T_mlc-(w_mbd*Ito1)+(w_mb/Rs).*(Vs-Vbo)+ w_md*(Vdo/Rd);

Ito4 = -(1/T_mlc+1/Tdd)*Ito3 -(w_mbd+1/(Tdd*T_mlc))*Ito2...
    -(w_mbd/Tdd-w_mb/Tbs-w_md/Tdd).*Ito1 - (1/Tbs-1/Tdd)*(w_mb/Rs).*(Vs-Vbo);

syms z(t) t Y
```

```
D1z = diff(z,t);
D2z = diff(z,t,2);
D3z = diff(z,t,3);
D4z = diff(z,t,4);
Eqn4 = D4z + E1*D3z + E2*D2z + E3*D1z+ E4*z == 0;
% Eqn4 = D4z + 20e3*D3z + 5e3*D2z + 2e5*D1z + 3e2*z == 0;
[VF4,Subs4] = odeToVectorField(Eqn4);
% IC1 = [Ito1-17.73;Ito2;Ito3;Ito4]; % initial conditions
IC1 = [Ito1;Ito2;Ito3;Ito4]; % initial conditions
RRcurent4 = matlabFunction(VF4, 'Vars',{t,Y});
[t,z] = ode45(RRcurent4,time1,IC1);
plot(t,z(:,1)),grid
% plot(t,z(:,1),'b',x,y(:,1),'b'),grid
% legend(string(Subs4))
title('Load Current')
xlabel('Time [s]'), ylabel('Current [A]')
set(get(gca,'children'),'linewidth',2);
 set(gca,'position',[0.12,0.11,0.79,0.84]);
set(gca,'linewidth',3,'fontname','helvetica','fontweight','bold','fontsize',1
6);
set(get(gca,'xlabel'),'fontname','helvetica','fontweight','bold','fontsize',1
6)
set(get(gca,'ylabel'),'fontname','helvetica','fontweight','bold','fontsize',1
6)
set(get(gca,'title'),'fontname','helvetica','fontweight','bold','fontsize',16
)
 toc
%% Matrix method for solving the fourth order ODE
% t0 = 4.831e-3; % initial discharge time
% S = [exp(v1*t0)      exp(v2*t0)       exp(v3*t0)       exp(v4*t0);
%       v1*exp(v1*t0)   v2*exp(v2*t0)    v3*exp(v3*t0)    v4*exp(v4*t0);
%       v1^2*exp(v1*t0) v2^2*exp(v2*t0)  v3^2*exp(v3*t0)  v4^2*exp(v4*t0);
%       v1^3*exp(v1*t0) v2^3*exp(v2*t0)  v3^3*exp(v3*t0)  v4^3*exp(v4*t0)];
% F = inv(S)*IC1;
% current = F(1,1)*exp(v1.*time1)+ F(2,1)*exp(v2.*time1)+...
%             F(3,1)*exp(v3.*time1)+ F(4,1)*exp(v4.*time1);
%     plot(time1,current),grid minor
```

Appendix B: (MATLAB) Carbon Strip Degradation Test Stand Verification Theory 2 (Carbon Resistance a Function of Time)

```
clc,close all,clear all
tic
% circuit parameters

Vs = 2.5e3; Rs = 250; Cb = 3.2e-6; RL = 1.008e-0; Lm = 1.1e-6;
Cd = 3.16e-12; Rd = 1e-5;

w_mb = 1/(Cb*Lm);Tbs = Rs*Cb;  Tdd = Rd*Cd;
w_md = 1/(Cd*Lm); Cbd = Cb*Cd/(Cb+Cd); w_mbd = 1/(Lm*Cbd);


%% Third order ODE to be solved for the charging network
syms Rc(t) Tmlc(t) A1(t) A2(t) A3(t) y(t) t Y
time = [0 10e-3]; % time range
Rc(t) = 0.003+0.015*exp(-0.001*t.^5+0.015*t.^3)+0.015*exp(-0.01*t.^3); %
model of the carbon strip resistance
Tmlc(t) = Lm./(RL+Rc(t));

A1(t) = 1./Tmlc(t) + 1/Tbs;
A2(t) = diff(1./Tmlc(t),t)+w_mbd+(1/Lm).*diff(Rc(t),t)+1/(Tbs.*Tmlc(t));
A3(t) = (1/Lm).*diff(Rc(t),t,t)+(1/(Lm*Tbs)).*diff(Rc(t),t)...
    +(w_mbd-w_mb)/Tbs;

D1y(t) = diff(y(t),t);
D2y(t) = diff(y(t),t,t);
D3y(t) = diff(y(t),t,t,t);
Eqn3 = D3y(t) + A1(t)*D2y(t) + A2(t)*D1y(t) + A3(t)*y(t) == 0;
% [VF3,Subs3] = odeToVectorField(A2,A3,Eqn3);
[VF3,Subs3] = odeToVectorField(Eqn3);
Ito = w_mb*(Vs/Rs); % initial value
IC = [0 0 Ito]; % initial conditions
R = matlabFunction(VF3, 'Vars',{t,Y});
[T,I] = ode45(R,time,IC);
plot(T,I),grid,title('Charging Network Current signature')
legend(string(Subs3))

Rp = Rc(T); % generating values for the carbon strip resistance
V_d = cumtrapz(I(:,1),T)./Cd; % discharge voltage at final charging time
V_b = -((RL+Rp).*I(:,1)+V_d+Lm.*I(:,2)); % bank voltage at final charging
time
figure
plot(T,V_d,'k',T,-V_b,'r'),grid
%% Fourth order ODE to be solved for the discharging network
syms Rc(t) Tmlc(t) B1(t) B2(t) B3(t) B4(t) z(t) t Y
time2 = [0 9e-5]; % time range
Rc(t) = 0.003+0.015*exp(-0.001*t.^5+0.015*t.^3)+0.015*exp(-0.01*t.^3); %
model of the carbon strip resistance
Tmlc(t) = Lm./(RL+Rc(t));

B1(t) = 1./Tmlc(t) + 1/Tbs + 1/Tdd;
B2(t) = 2*diff(1./Tmlc(t),t)+w_mbd+(1/Lm).*diff(Rc(t),t)+...
```

```
    (1/Tbs + 1/Tdd)*1/Tmlc(t) + 1/(Tbs*Tdd);
B3(t) = diff(1./Tmlc(t),t,t)+ (2/Lm).*diff(Rc(t),t,t)+(1/Tbs+1/Tdd).*...
   (diff(1./Tmlc(t),t)+w_mbd+(1/Lm).*diff(Rc(t),t))+1/(Tbs*Tdd*Tmlc(t))-
w_mb/Tbs-w_md/Tdd;
B4(t) = (1/Lm).*diff(Rc(t),t,t,t)+(1/Tbs+1/Tdd)*(1/Lm).*diff(Rc(t),t,t)+...
    (1/(Lm*Tbs*Tdd)).*diff(Rc(t),t)-(w_mb+w_md-w_mbd)/(Tbs*Tdd);

% initial conditions as evaluated (approx.) from the ODEs at time t = t0+
Vbo = 2.0e3; Vdo = 1.98e3; % initial cap. bank & discharge voltage
% Vbo = double(V_b); Vdo = V_d;
DR = diff(Rc); D2R = diff(Rc,2);
DT = diff(1/Tmlc); D2T = diff(1/Tmlc,2);
t0 = 2.6e-3;
Ito1 = (Vbo-Vdo)/(RL+Rc(t0)); Ito2 = (Vbo-Vdo)/Lm-((RL+Rc(t0))/Lm)*Ito1;
Ito3 = -Ito2/Tmlc(t0)-(w_mbd+(1/Lm).*DR(t0))*Ito1 +(w_mb/Rs)*(Vs-Vbo)+...
     + w_md*(Vdo/Rd);
Ito4 = -(1/Tmlc(t0)+1/Tdd)*Ito3 -
(DT(t0)+w_mbd+(1/Lm)+DR(t0)+1/(Tdd*Tmlc(t0)))*Ito2...
    -((1/Lm)*D2R(t0)+(1/Tdd)*(w_mbd+(1/Lm).*DR(t0))-
(w_mb/Tbs+w_md/Tdd))*Ito1...
    + (1/Tdd-1/Tbs)*(w_mb/Rs)*(Vs-Vbo);
It1 = double(Ito1); It2 = double(Ito2); It3 = double(Ito3); It4 =
double(Ito4);

% initial conditions as taken directly from the charging network solution
% at time t = t0+
% Ito1 = 14.4e-3; Ito2 = -16.56; Ito3 = -2.49e10; Ito4 = 1.68e22;

D1z(t) = diff(z(t),t);
D2z(t) = diff(z(t),t,t);
D3z(t) = diff(z(t),t,t,t);
D4z(t) = diff(z(t),t,t,t,t);
Eqn4 = D4z(t) + B1(t)*D3z(t) + B2(t)*D2z(t) + B3(t)*D1z(t) + B4(t)*z(t)== 0;
% [VF4,Subs4] = odeToVectorField(B2,B3,B4,Eqn4);
[VF4,Subs4] = odeToVectorField(Eqn4);
% IC2 = [Ito1 Ito2 Ito3 Ito4]; % initial conditions
% IC2 = [0 0 0 Ito1 Ito2 Ito3 Ito4]; % initial conditions
IC2 = [It1;It2;It3;It4]; % initial conditions
Rdc = matlabFunction(VF4, 'Vars',{t,Y});
[Tdc,Idc] = ode45(Rdc,time2,IC2);
figure
plot(Tdc,Idc(:,1)),grid,title('Discharging Network Current signature')
legend(string(Subs4))
toc
```

Appendix C: LTSpice Circuit Netlist (Carbon Strip Degradation Test Stand Circuit Representation)

```
Vs N001 0 2.5k
Cb Vc 0 3.3µ
RL N002 Vc 1 % resistance for a 1 Ω load
Rm N004 0 0.008
Rs Vc N001 1k
Rc VL N002 0.028  % carbon resistance for “normal” condition
Cd Vc2 N004 3.16p
Rd N003 N004 0.5
S1 Vc2 N003 Vc2 N004 SW1
Lm VL Vc2 3.32µ Rser=1p
.ic V(Vc)=0V
.model SW1 SW(Ron=1u Roff=100G Vt=1.24k Vh=1.235k)
.ic V(VL)=0V
.ic V(Vc2)=0V
.tran 0 15.27m 15.19m 100n
.meas Ip MAX I(Lm) % to measure the peak value of the load current
.backanno
.end
```

Appendix D: EM Dot Data Acquisition and Processing Program Guide

This document provides a guide on how to use the in-house, data acquisition and processing software based on a MATLAB platform to process raw data collected by the EM dot in both B-dot mode and D-dot mode. When the EM dot is used to measure the magnetic field the dot is operating in B-dot mode. When the EM dot is used to measure the electric field the dot is operating in D-dot mode. Refer to the references [10,11] for more information regarding EM dot measurements. Data is displayed in both the spectral domain as well as in the time domain.

The EM dot sensor used in B-dot mode has already been calibrated in the frequency domain over a wide range of low discrete frequencies between 10 MHz and 990 MHz. A calibration table should have accompanied your dot upon initial receipt of the dot. Upon request, the dot can also be calibrated over discrete high frequencies between 2 and 6 GHz. The calibration numbers supplied with the EM dot are used as multiplication factors in converting the raw uncalibrated field data to calibrated field data.

Only the multi-frequency processing option will be presented. This option allows us to process the data captured by the EM dot and multiply the result by a set of the calibration numbers at specified frequencies (spectral calibration function) to generate the calibrated magnitude of the magnetic field.

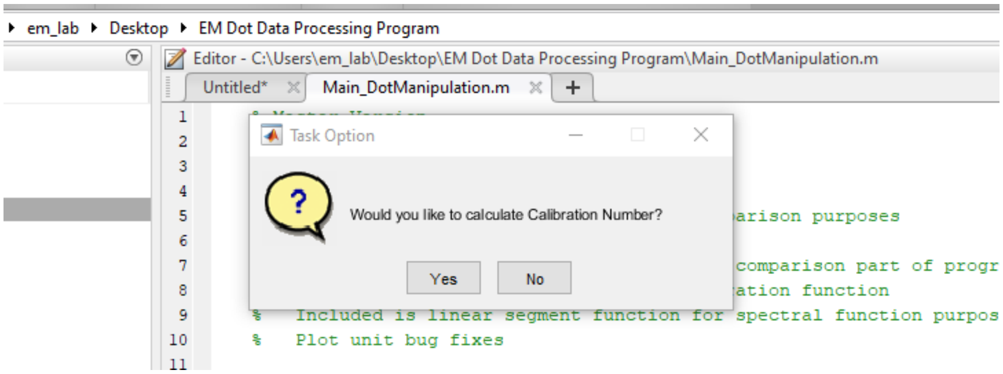
To begin, please ensure all the .m files are saved in a single folder. The required files for this program are listed in appendices E - M, in no particular order. The filenames are:

1. Appendix E - Main_DotManipulation.m (Main file for data acquisition and processing)
2. Appendix F – datainput.m (Data input function file)
3. Appendix G – fftfilter.m (Performing fast Fourier transform)
4. Appendix H - inputdlg2.m (Input dialogue program)
5. Appendix I – plotter.m (Plotter)
6. Appendix J – splotter.m (Spectral plotter)
7. Appendix K – spectralprocessing.m (Spectral processing file)
8. Appendix L – Mfquestdlg.m (Input dialogue box file)
9. Appendix M - struct2csv.m (Structure to .CSV file program)

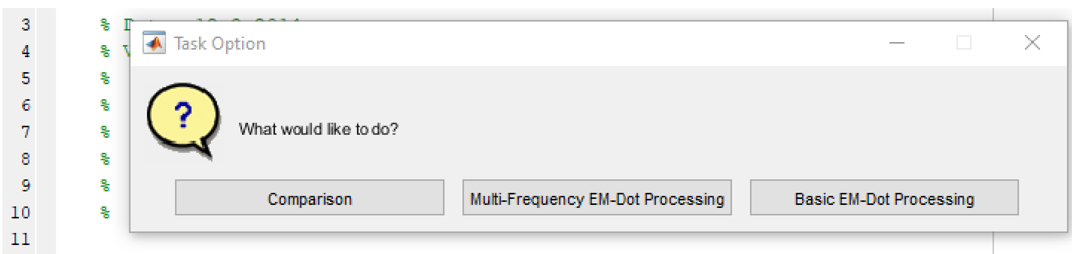
The filenames need to be attached to the programs listed in each appendix. Text in parentheses is a brief modifier subtitle.

Below are the steps to obtain the calibrated data from the program:

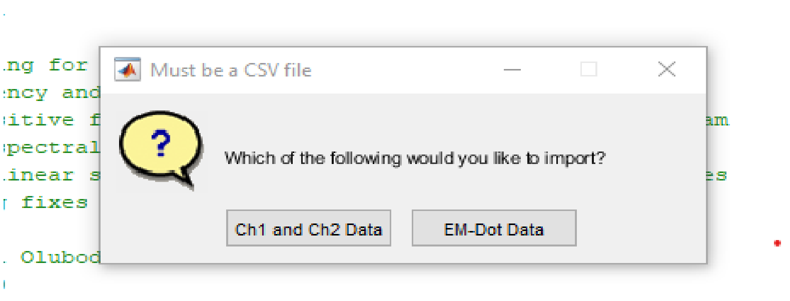
(i) Run the file "Main_DotManipulation.m" and select "No" when asked the question "Would you like to calculate Calibration Number?"



(ii) Please select "Multi-Frequency EM-Dot Processing" when the question "What would you like to do?" is displayed.

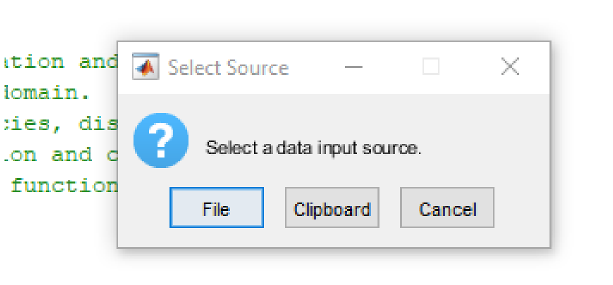


(iii) The next step is to select "Ch1 and Ch2 Data" when the question "Which of the following would you like to import?" is displayed.

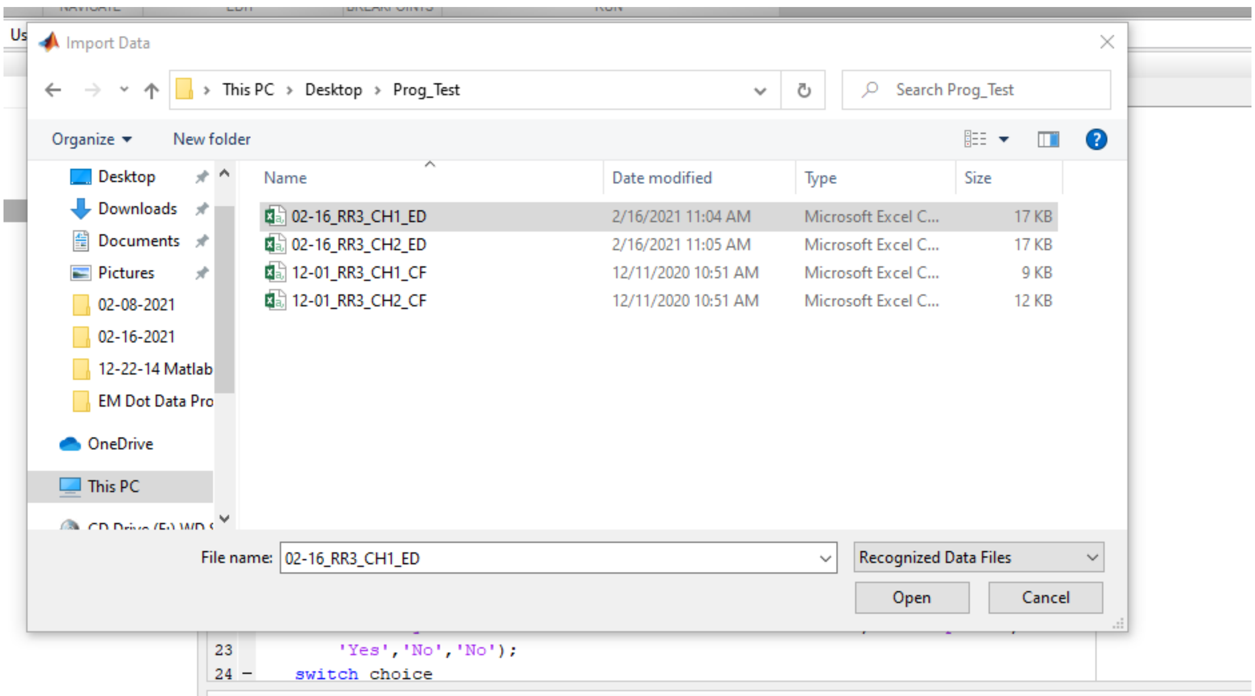


Selecting this option allows the user to upload the .csv files (excel file format) saved directly from the oscilloscope channels. Either of the channels can be uploaded as Ch1 or Ch2 without affecting the outcome of the magnetic field, it only inverts the waveform.

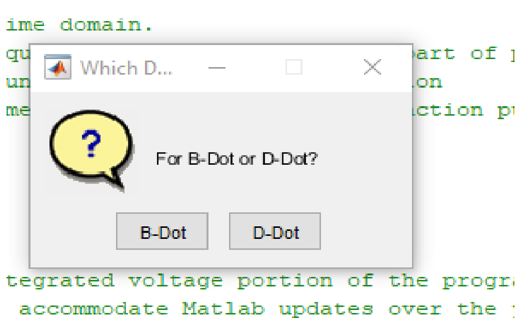
(iv) For the next step, please select the "File" option and upload the Ch1 file by clicking "open→next→finish" and repeat the same process to upload the Ch2 file.



To minimize the amount of time needed to find these files, it is suggested that the files from the oscilloscope be placed in the same folder as these processing files.

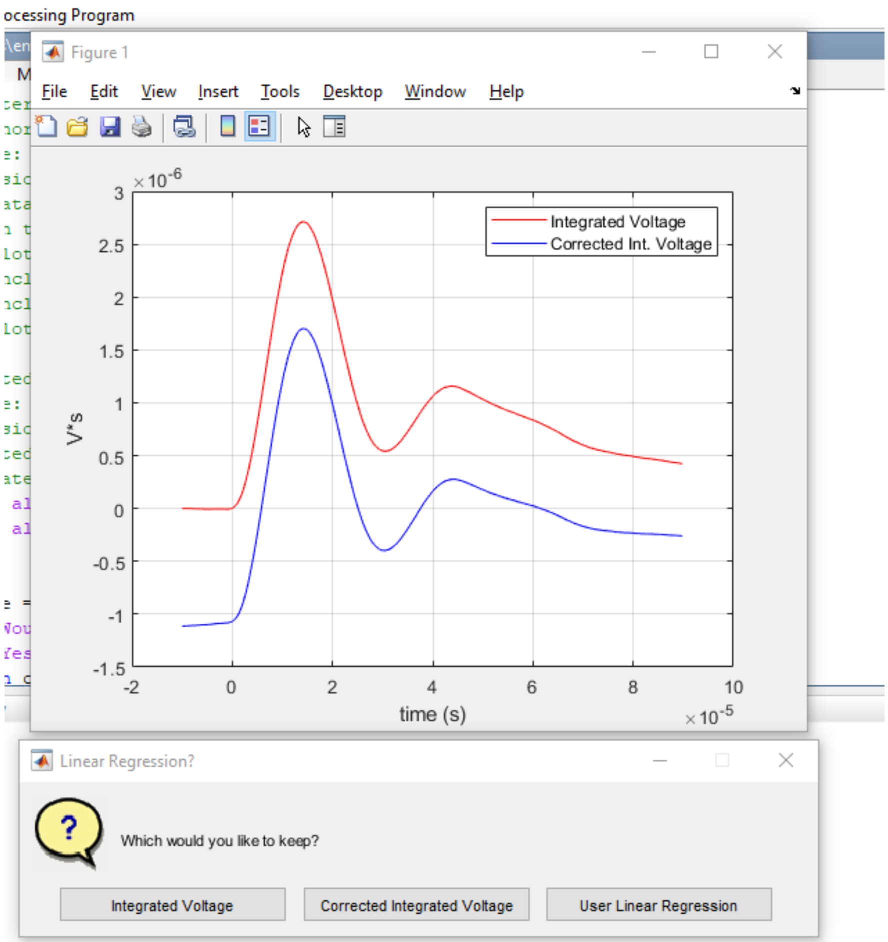


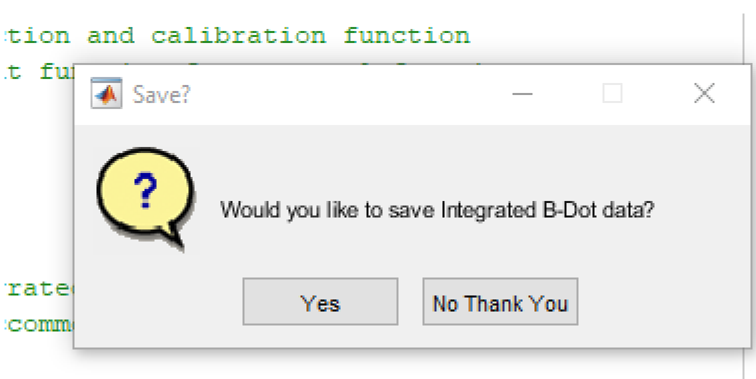
(v) Next is to select "B-Dot" since the dot is being used in the magnetic field mode.



This will generate an uncalibrated plot of the magnetic field vs time.

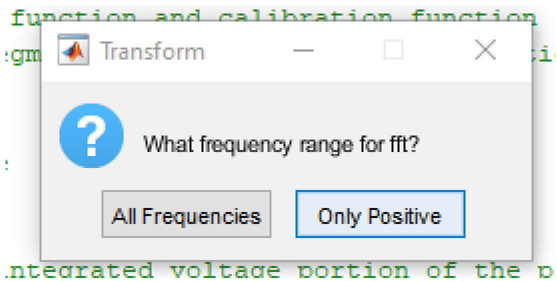
(vi) Select the “Integrated Voltage” option from below. The “Corrected integrated Voltage” option is to correct the plot using the ‘detrend’ command in Matlab if the signal is off the zero axes. The “User Linear Regression” is provided as an option for the user to write a customized regression model for plot correction as desired.

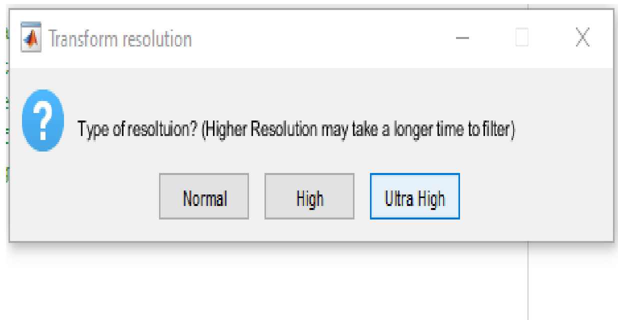




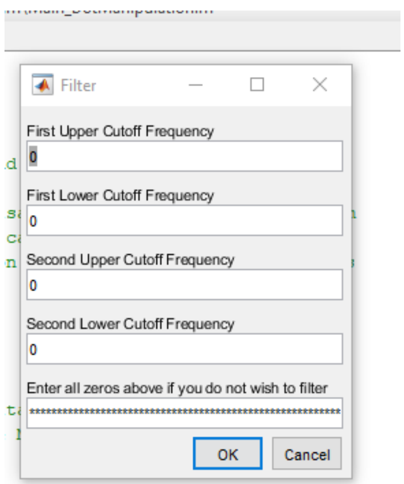
Please select “No Thank You” to proceed.

(vii) The next step is to transform the data into the Fourier domain (fft) and perform all the necessary computations including filtering and multiplying with the calibration numbers generated in the frequency domain.



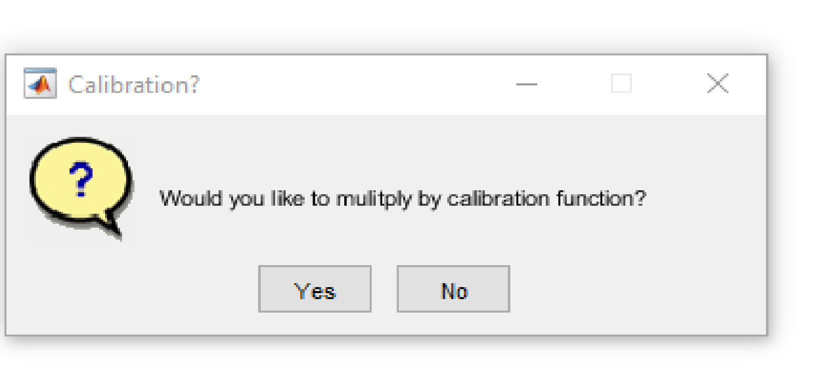


Please select "Only Positive" and "Normal" to save processing time.

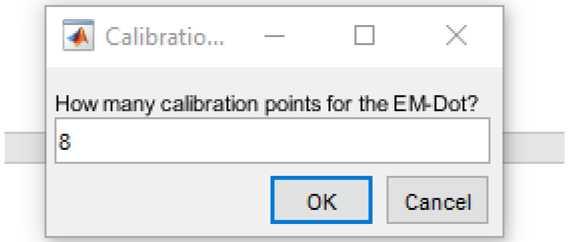


If filtering is not required, please leave all as zeros and click "OK"

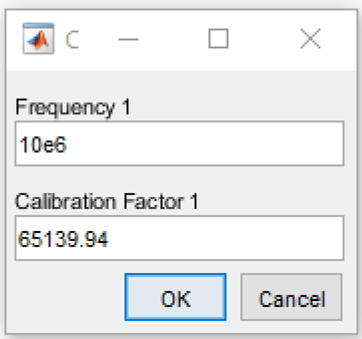
Next is to multiply by the calibration numbers. This is provided in a separate excel file containing the calibration numbers for the B-dot sensor.

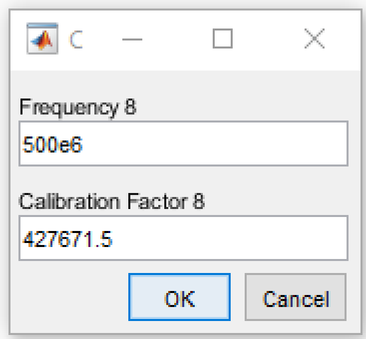


Select "Yes" and enter the number of calibration points you desired based on the frequency range as supplied in the excel file containing the calibration numbers, then click "OK".

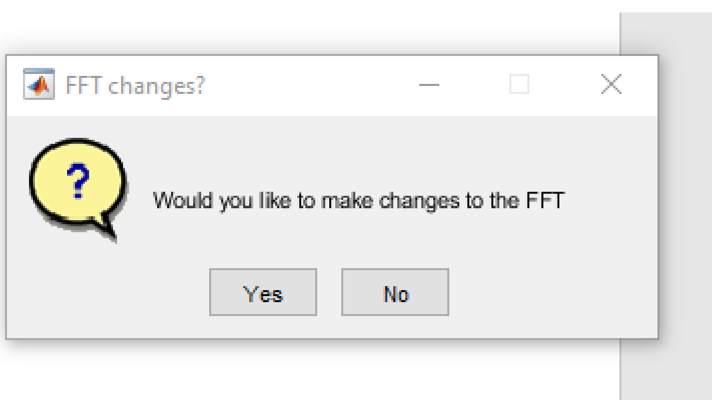


For this experiment, we used eight (8) points corresponding to 10MHz, 20MHz, 50MHz, 100MHz, 200MHz, 300MHz, 400MHz, and 500MHz. More points could be used if desired.



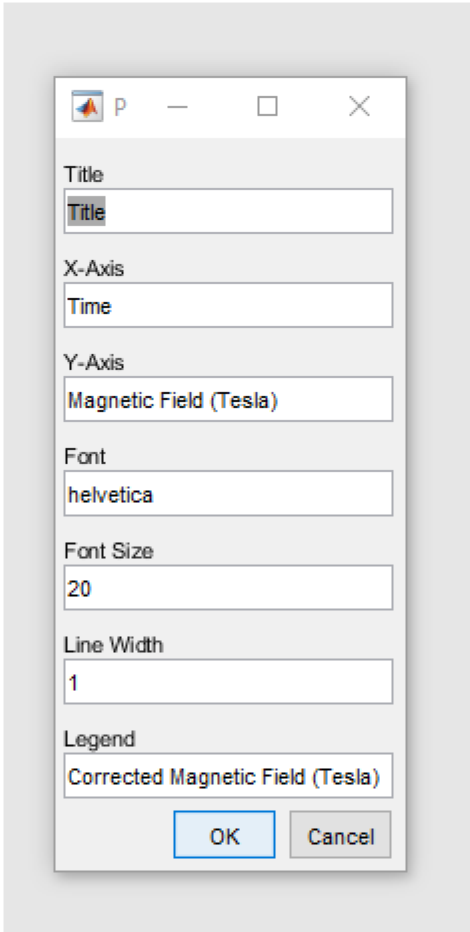


Please provide the frequency and the corresponding calibration number and click "OK" until the last number of points is reached.

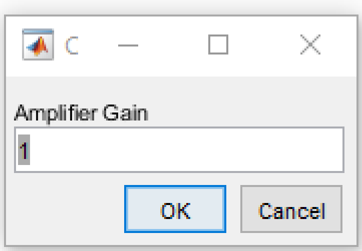


Please select "No" unless you wish to make changes to the fft.

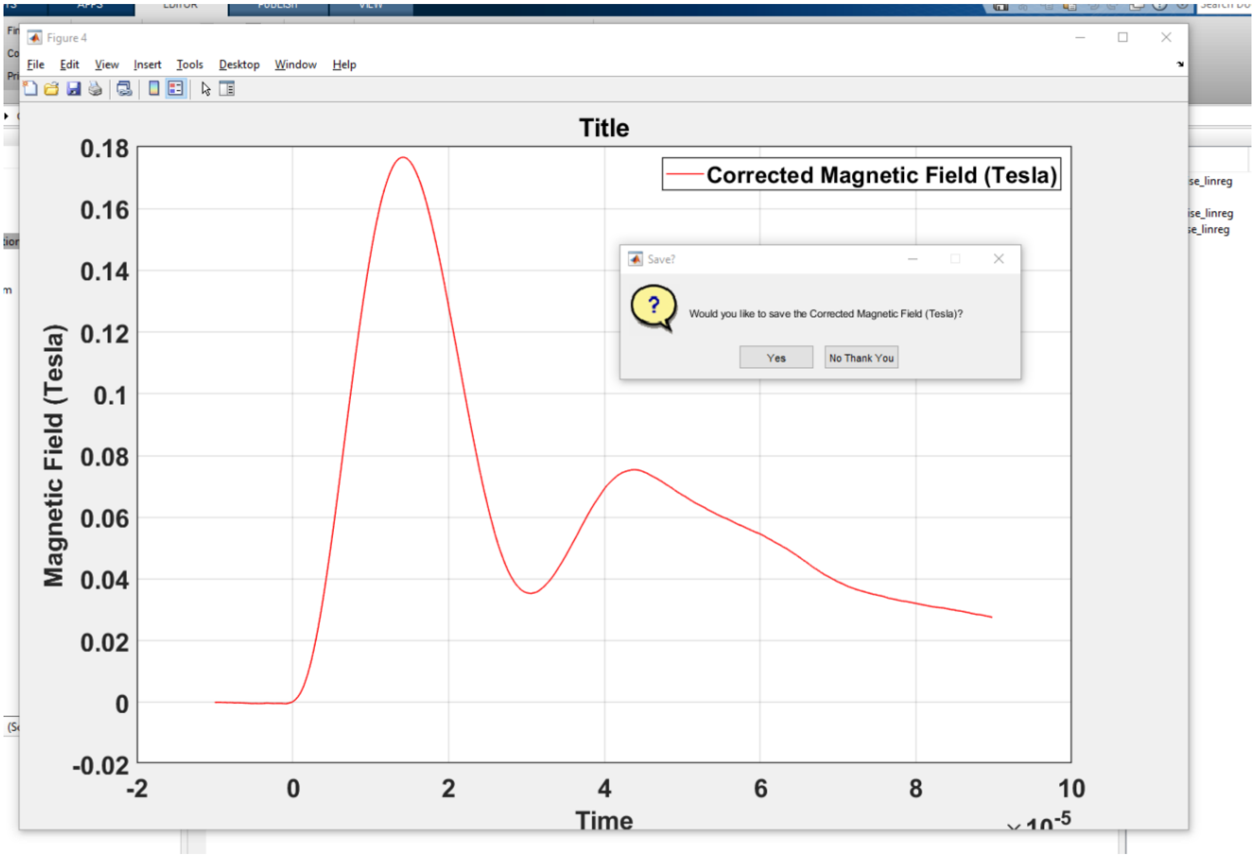
(viii) The next step is to label the plot as desired by changing the title, x-axis and y-axis, and other properties of the plot.



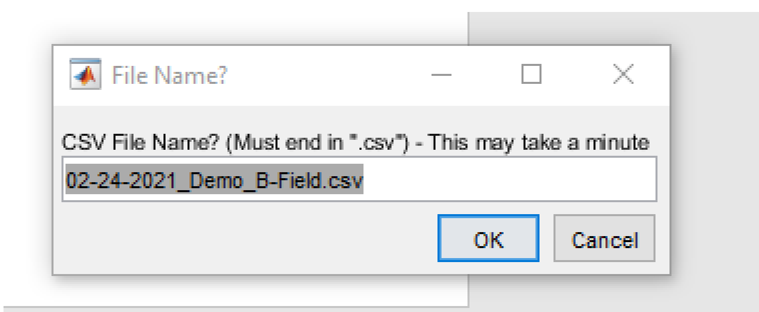
(ix) If an amplifier was not used for data collection, then the "Amplifier Gain" is one. Please input "1" and click "OK". If an amplifier was used for data collection, please provide the amplifier gain as a constant (not in dB).



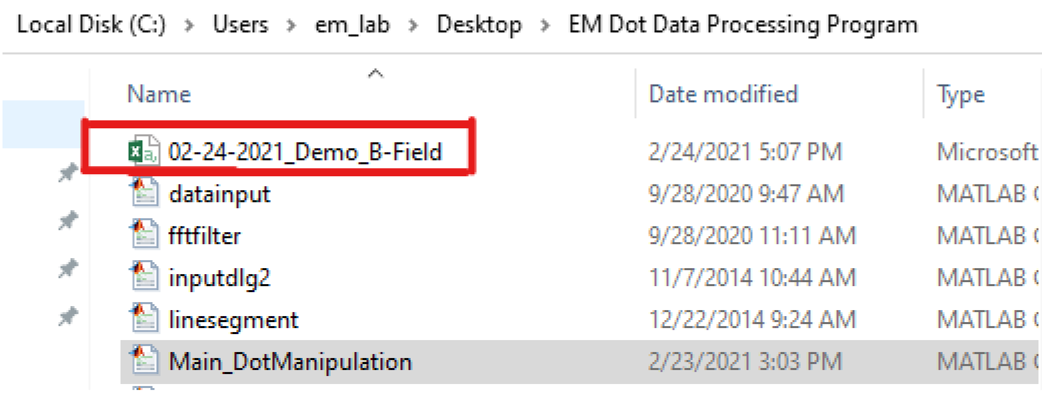
This is the last step. At this point, a final plot of the calibrated magnitude of the magnetic field versus time is generated with an option to save the result as a .csv file. The plot could also be saved if so desired.



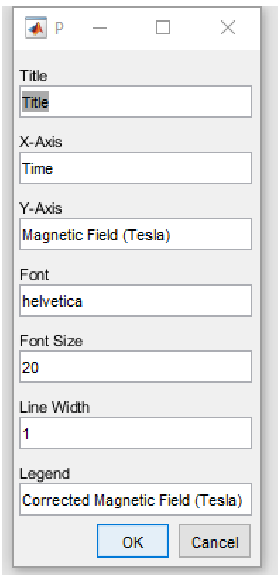
It is suggested to select "Yes" to save the "Corrected Magnetic Field(Tesla)" for further analysis. With this, analysis can be done using any other program of the user's choice.



Use any name of your choice and click "OK". The file is saved in the same folder as the rest of the processing files.



If the figure below is displayed again, please click "Cancel" to exit the program.



Appendix E: MATLAB Program Associated with Dot Processing [Main_DotManipulation.m]

(Main_DotManipulation.m - Required for Data Acquisition and Processing)

```
%   Main file for Data Processing and EM dot calibration
%   in the frequency and time domain.
%   Plot only positive frequencies, disable comparison part of program
%   Included is spectral function and calibration function
%   Included is linear segment function for spectral function purposes
%   Plot unit bug fixes

clear all
close all
clc

choice = MFquestdlg('Would you like to calculate Calibration Number?', ...
    'Would you like to calculate Calibration Number?','Task Option', ...
    'Yes','No','No');
switch choice
case 'Yes'
         calibration()
%          break
case 'No'
end

choice = MFquestdlg('What would like to do?', ...
    'What would like to do?','Task Option', ...
    'Comparison', ...
    'Multi-Frequency EM-Dot Processing', ...
    'Basic EM-Dot Processing','Basic EM-Dot Processing');
switch choice
    case 'Comparison'
        comparison();
    case 'Multi-Frequency EM-Dot Processing'
         Ddot_noise_linreg = 0;
         Bdot_noise_linreg = 0;
         user_noise_linreg = 0;
        [CorrB,CorrD,time] =
datainput(Ddot_noise_linreg,Bdot_noise_linreg,user_noise_linreg);
        [CorrB,CorrD,time] = spectralprocessing(CorrB,CorrD,time);
    case 'Basic EM-Dot Processing'
true = 0;
while true == 0
    close all
choice = MFquestdlg('Do you have a noise sample?', ...
    'Do you have a noise sample?','Noise Sample','Yes','No','No');
switch choice
    case 'Yes'
        [Ddot_noise_linreg,Bdot_noise_linreg,user_noise_linreg] =
noiseInput();
        [CorrB,CorrD,time] =
datainput(Ddot_noise_linreg,Bdot_noise_linreg,user_noise_linreg);
        [CorrB,CorrD,time] = fftfilter(CorrB,CorrD,time);
        [CorrB,CorrD,time] = plotter(CorrB,CorrD,time);
    case 'No'
```

```
        Ddot_noise_linreg  = 0;
        Bdot_noise_linreg  = 0;
        user_noise_linreg  = 0;
        [CorrB,CorrD,time] =
datainput(Ddot_noise_linreg,Bdot_noise_linreg,user_noise_linreg);
        [CorrB,CorrD,time] = fftfilter(CorrB,CorrD,time);
        [CorrB,CorrD,time] = plotter(CorrB,CorrD,time);
end

choice = MFquestdlg('Would You like to do another Data-set?', ...
 'Would You like to do another Data-set?','New Data', ...
 'Yes','No','No');
switch choice
    case'Yes'
        true = 0;
    case'No'
        true = 1;
end
end
end
```

Appendix F: MATLAB Program Associated with Dot Processing [datainput.m] (Code for Data Input Function File)

```
%   This function file uploads the .csv file and performs the integration and
%   all the required operations to obtain the magnetic field


function [CorrB,CorrD,time] =
datainput(Ddot_noise_linreg,Bdot_noise_linreg,user_noise_linreg)
CorrB = 0;
CorrD = 0;

choice = MFquestdlg('Which of the following would you like to import?', ...
 'Which of the following would you like to import?','Must be a CSV file', ...
 'Ch1 and Ch2 Data','EM-Dot Data', 'EM-Dot Data');
switch choice
    case 'Ch1 and Ch2 Data'
    Ch1File = uiimport;
    Ch2File = uiimport;
    time = Ch1File.data(:,1);
    Ch1 = Ch1File.data(:,2);
    Ch2 = Ch2File.data(:,2);
choice = MFquestdlg('For B-Dot or D-Dot?', ...
    'For B-Dot or D-Dot?','Which Data?', ...
    'B-Dot','D-Dot','D-Dot');
        switch choice
            case 'B-Dot'

                    Mminus = Ch1-Ch2;
                    Bdata=cumtrapz(time,Mminus);
                    dtBdata = detrend(Bdata);

if user_noise_linreg ~=0
                        UBnoiseLine = (user_noise_linreg(1,1)*time +
user_noise_linreg(1,2));
                        CorrB = Bdata - UBnoiseLine;

                        figure
                        plot(time,Bdata,'r', time, CorrB, 'b');
                        grid
                        legend('Integrated Voltage','Corrected Int. Voltage')
                        xlabel('time (s)') % x-axis label
                        ylabel('V*s') % y-axis label
                        disp('user linreg');
else if Bdot_noise_linreg ~=0
                            Bnoiseline = (Bdot_noise_linreg(1,1)*time +
Bdot_noise_linreg(1,2));
                            CorrB = Bdata - Bnoiseline;

                            figure
                            plot(time,Bdata,'r', time, CorrB, 'b');
                            grid
                            legend('Integrated Voltage','Corrected Int.
Voltage')
                            xlabel('time (s)') % x-axis label
```

```
                            ylabel('V*s') % y-axis label
                            disp('linreg');

                        else

                            figure
                            plot(time,Bdata,'r', time, dtBdata, 'b');
                            grid
                            legend('Integrated Voltage','Corrected Int.
Voltage')
                            xlabel('time (s)') % x-axis label
                            ylabel('V*s') % y-axis label
                        end
                    end

                    choice = MFquestdlg('Which would you like to keep?', ...
                    'Which would you like to keep?', 'Linear Regression?',
...
                    'Integrated Voltage','Corrected Integrated Voltage','User
Linear Regression','User Linear Regression');
                    switch choice
                        case 'Integrated Voltage'
                            CorrB = Bdata;
                        case 'Corrected Integrated Voltage'
                            CorrB = dtBdata;
                        case 'User Linear Regression'
                            true = 10;
                            while true > 1;
                                prompt = {'Enter M value:','Enter B value:'};
                                dlg_title = 'User Linear Regression in the
form Y = Mx+ B';
                                num_lines = 1;
                                def = {'0','0'};
                                options.Resize='on';
                                options.WindowStyle='normal';

answer=inputdlg(prompt,dlg_title,num_lines,def,options);

                                m = str2double(answer{1});
                                b = str2double(answer{2});
                                CBLR = [m,b];
                                CBLine = (CBLR(1,1)*time + CBLR(1,2));
                                CorrB = Bdata - CBLine;

                                figure
                                plot(time,Bdata,'g', time, CBLine, 'b');
                                grid
                                legend('Integrated Voltage','Linear Reg.')
                                xlabel('time (s)') % x-axis label
                                ylabel('V*s') % y-axis label

                                figure
                                plot(time,CorrB,'r');
                                grid
                                legend('Corrected Int. Voltage')
                                xlabel('time (s)') % x-axis label
```

```
                                ylabel('V*s') % y-axis label
choice = MFquestdlg('Would you like to try another linear regression?', ...
                                'Would you like to try another linear
regression?','Linear Regression', ...
                                'Yes', 'No Thank You', 'No Thank You');
                                switch choice
                                    case 'Yes'
                                        true = 10;
                                    case 'No Thank You'
                                        true = 0;
                                end
                            end
                            close all


                            choice = MFquestdlg('Would you like to invert the
Integrated Voltage chosen?', ...
                            'Would you like to Invert the Corrected B-
Data?','Invert Data', ...
                            'Yes', 'No Thank You', 'No Thank You');
                            switch choice
                                case 'Yes'
                                    CorrB = -CorrB;
                                    figure
                                    plot(time,CorrB,'b');
                                    grid
                                    legend('Corrected Int. Voltage')
                                    xlabel('time (s)') % x-axis label
                                    ylabel('V*s') % y-axis label

                            case 'No Thank You'
                            end
                    end
choice = MFquestdlg('Would you like to save Corrected Integrated Voltage?',
...
'Would you like to save Integrated B-Dot data?','Save?','Yes', 'No Thank
You', 'No Thank You');
switch choice
    case 'Yes'
        prompt = {'CSV File Name? (Must end in ".csv") - This may take a
minute'};
        dlg_title = 'File Name?';
        num_lines = 1;
        def = {'intBdot.csv'};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        fname = answer{1};
        structsave =
struct('filename',fname,'time',time,'Corr_Int_B_Volt',CorrB);
        struct2csv(structsave,fname);
    case 'No Thank You'
end

            case 'D-Dot'
```

```
                Mplus = Ch1+Ch2;
                Ddata=cumtrapz(time,Mplus);
                dtDdata = detrend(Ddata);

                    if user_noise_linreg ~=0
                        UDnoiseLine = (user_noise_linreg(1,1)*time +
user_noise_linreg(1,2));
                        CorrD = Ddata - UDnoiseLine;

                        figure
                        plot(time,Ddata,'r', time, CorrD, 'b');
                        grid
                        legend('Integrated Voltage','Corrected Int. Voltage')
                        xlabel('time (s)') % x-axis label
                        ylabel('V*s') % y-axis label
                        disp('user linreg');

                    else if Bdot_noise_linreg ~=0
                            Dnoiseline = (Ddot_noise_linreg(1,1)*time +
Ddot_noise_linreg(1,2));
                            CorrD = Ddata - Dnoiseline;

                            figure
                            plot(time,Ddata,'r', time, CorrD, 'b');
                            grid
                            legend('Integrated Voltage','Corrected Int.
Voltage')
                            xlabel('time (s)') % x-axis label
                            ylabel('V*s') % y-axis label
                            disp('noise linreg');

                        else
                            figure
                            plot(time,Ddata,'r', time, dtDdata, 'b');
                            grid
                            legend('Integrated Voltage','Corrected Int.
Voltage')
                            xlabel('time (s)') % x-axis label
                            ylabel('V*s') % y-axis label
                        end
                    end

                    choice = MFquestdlg('Which would you like to keep?', ...
                    'Which would you like to keep?','Linear Regression?', ...
                    'Integrated Voltage','Corrected Integrated Voltage','User
Linear Regression','User Linear Regression');
                    switch choice
                        case 'Integrated Voltage'
                        case 'Corrected Integrated Voltage'
                            CorrD = dtDdata;
                        case 'User Linear Regression'
                            true = 10;
                            while true >= 1
                                prompt = {'Enter M value:','Enter B value:'};
                                dlg_title = 'User Linear Regression in the
form Y = Mx+ B';
```

```
                                num_lines = 1;
                                def = {'0','0'};
                                options.Resize='on';
                                options.WindowStyle='normal';

answer=inputdlg(prompt,dlg_title,num_lines,def,options);

                                m = str2double(answer{1});
                                b = str2double(answer{2});
                                CDLR = [m,b];
                                CDLine = (CDLR(1,1)*time + CDLR(1,2));
                                CorrD = Ddata - CDLine;

                                figure
                                plot(time,Ddata,'g',time,CDLine,'b');
                                grid
                                legend('Integrated Votlage','Linear Reg.')
                                        xlabel('time (s)') % x-axis label
                                        ylabel('V * s') % y-axis label
                                figure
                                plot(time,CorrD,'r');
                                grid
                                legend('Corrected Int. Voltage')
                                xlabel('time (s)') % x-axis label
                                ylabel('V*s') % y-axis label

                                choice = MFquestdlg('Would you like to try
another linear regression?', ...
                                'Would you like to try another linear
regression?','Linear Regression', ...
                                'Yes', 'No Thank You', 'No Thank You');
                                switch choice
                                    case 'Yes'
                                        true = 10;
                                case 'No Thank You'
                                        true = 0;

                                end
                            end
                    end
choice = MFquestdlg('Would you like to save the Corrected Integrated
Voltage?', ...
'Would you like to save the Corrected Integrated Voltage?','Save?','Yes', 'No
Thank You', 'No Thank You');
switch choice
    case 'Yes'
        prompt = {'CSV File Name? (Must end in ".csv") - This may take a
minute'};
        dlg_title = 'File Name?';
        num_lines = 1;
        def = {'intDdot.csv'};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        fname = answer{1};
```

```
        structsave =
struct('filename',fname,'time',time,'Corr_Int_D_Volt',CorrD);
        struct2csv(structsave,fname);
    case 'No Thank You'
end

        end

    case 'EM-Dot Data'
        MFile = uiimport;
        time = MFile.data(:,1);
        IMC = MFile.data(:,2);
        choice = MFquestdlg('Is this for B-Dot or D-Dot?', ...
        'Is this for B-Dot or D-Dot?','Data Type?', ...
        'B-Dot','D-Dot','D-Dot');
            switch choice
                case 'B-Dot'
                    dtBdata = detrend(IMC);
                    if user_noise_linreg ~=0
                        UBnoiseLine = (user_noise_linreg(1,1)*time +
user_noise_linreg(1,2));
                        CorrB = IMC - UBnoiseLine;

                        figure
                        plot(time,IMC,'r', time, CorrB, 'b');
                        grid
                        legend('Integrated Voltage','Corrected Int. Voltage')
                        xlabel('time (s)') % x-axis label
                        ylabel('V*s') % y-axis label

                    else if Bdot_noise_linreg ~=0
                            Bnoiseline = (Bdot_noise_linreg(1,1)*time +
Bdot_noise_linreg(1,2));
                            CorrB = IMC - Bnoiseline;

                            figure
                            plot(time,IMC,'r', time, CorrB, 'b');
                            grid
                            legend('Integrated Voltage','Corrected Int.
Voltage')
                            xlabel('time (s)') % x-axis label
                            ylabel('V*s') % y-axis label

                        else
                            CorrB = dtBdata;
                            figure
                            plot(time,IMC,'r', time, CorrB, 'b');
                            grid
                            legend('Integrated Voltage','Corrected Int.
Voltage')
                            xlabel('time (s)') % x-axis label
                            ylabel('V*s') % y-axis label
                        end
                    end
```

```
                    choice = MFquestdlg('Which would you like to keep?', ...
                    'Which would you like to keep?', 'Linear Regression?',
...
                    'Integrated Voltage','Corrected Integrated Voltage','User
Linear Regression','User Linear Regression');
                    switch choice
                        case 'Integrated Voltage'
                        case 'Corrected Integrated Voltage'
                            CorrB = dtBdata;
                        case 'User Linear Regression'
                            true = 10;
                            while true > 1;
                                prompt = {'Enter M value:','Enter B value:'};
                                dlg_title = 'User Linear Regression in the
form Y = Mx+ B';
                                num_lines = 1;
                                def = {'0','0'};
                                options.Resize='on';
                                options.WindowStyle='normal';

answer=inputdlg(prompt,dlg_title,num_lines,def,options);

                                m = str2double(answer{1});
                                b = str2double(answer{2});
                                CBLR = [m,b];
                                CBLine = (CBLR(1,1)*time + CBLR(1,2));
                                CorrB = IMC - CBLine;

                                figure
                                plot(time,IMC,'g',time, CBLine,'b');
                                grid
                                legend('Integrated Voltage','Linear Reg')
                                xlabel('time (s)') % x-axis label
                                ylabel('V*s') % y-axis label
                                figure
                                plot(time,CorrB,'r');
                                grid
                                legend('Corrected Integrated Voltage')
                                xlabel('time (s)') % x-axis label
                                ylabel('V*s') % y-axis label

                                choice = MFquestdlg('Would you like to try
another linear regression?', ...
                                'Would you like to try another linear
regression?','Linear Regression', ...
                                'Yes', 'No Thank You', 'No Thank You');
                                switch choice
                                    case 'Yes'
                                        true = 10;
                                case 'No Thank You'
                                        true = 0;
                                end
                            end
                    end
                            choice = MFquestdlg('Would you like to Invert the
Corrected Integrated Voltage?', ...
```

```
                                'Would you like to Invert the fixed B-
Data?','Invert Data', ...
                                'Yes', 'No Thank You', 'No Thank You');
                                switch choice
                                    case 'Yes'
                                        CorrB = -CorrB;
                                        figure
                                        plot(time,CorrB,'b');
                                        grid
                                        legend('Corrected Int. Voltage')
                                        xlabel('time (s)') % x-axis label
                                        ylabel('V*s') % y-axis label
                                    case 'No Thank You'
                                end
choice = MFquestdlg('Would you like to save the Corrected Inegrated
Voltage?', ...
'Would you like to save the Corrected Integrated Voltage?','Save?','Yes', 'No
Thank You', 'No Thank You');
switch choice
    case 'Yes'
        prompt = {'CSV File Name? (Must end in ".csv") - This may take a
minute'};
        dlg_title = 'File Name?';
        num_lines = 1;
        def = {'intBdot.csv'};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        fname = answer{1};
        structsave =
struct('filename',fname,'time',time,'Corr_Int_B_Volt',CorrB);
        struct2csv(structsave,fname);
    case 'No Thank You'
end


                case 'D-Dot'
                    dtDdata = detrend(IMC);

                    if user_noise_linreg ~=0
                        UDnoiseLine = (user_noise_linreg(1,1)*time +
user_noise_linreg(1,2));
                        CorrD = IMC - UDnoiseLine;

                        figure
                        plot(time,IMC,'r', time, CorrD, 'b');
                        grid
                        legend('Integrated Voltage','Corrected Int. Voltage')
                        xlabel('time (s)') % x-axis label
                        ylabel('V*s') % y-axis label

                    else if Ddot_noise_linreg ~=0
                            Dnoiseline = (Ddot_noise_linreg(1,1)*time +
Ddot_noise_linreg(1,2));
                            CorrD = IMC - Dnoiseline;
```

```
                            figure
                            plot(time,IMC,'r', time, CorrD, 'b');
                            grid
                            legend('Integrated Voltage','Corrected Int.
Voltage')
                            xlabel('time (s)') % x-axis label
                            ylabel('V*s') % y-axis label

                        else
                            CorrD = dtDdata;
                            figure
                            plot(time,IMC,'r', time, CorrD, 'b');
                            grid
                            legend('Integrated Voltage','Corrected Int.
Voltage')
                            xlabel('time (s)') % x-axis label
                            ylabel('V*s') % y-axis label
                        end
                    end

                   choice = MFquestdlg('Which would you like to keep?', ...
                    'Which would you like to keep?','Linear Regression?', ...
                    'Integrated Voltage','Corrected Integrated Voltage','User
Linear Regression','User Linear Regression');
                    switch choice
                        case 'Integrated Voltage'
                        case 'Corrected Integrated Voltage'
                            CorrD = dtDdata;
                        case 'User Linear Regression'
                            true = 10;
                            while true > 1;
                                prompt = {'Enter M value:','Enter B value:'};
                                dlg_title = 'User Linear Regression in the
form Y = Mx+ B';
                                num_lines = 1;
                                def = {'0','0'};
                                options.Resize='on';
                                options.WindowStyle='normal';

answer=inputdlg(prompt,dlg_title,num_lines,def,options);

                                m = str2double(answer{1});
                                b = str2double(answer{2});
                                CBLR = [m,b];
                                CBLine = (CBLR(1,1)*time + CBLR(1,2));
                                CorrD = IMC - CBLine;

                                figure
                                plot(time,IMC,'g',time,CBLine,'b');
                                grid
                                legend('Integrated Voltage','Linear Reg')
                                        xlabel('time (s)') % x-axis label
                                        ylabel('V*s') % y-axis label
                                figure
                                plot(time,CorrD,'r');
                                grid
```

```
                                legend('Corrected Integrated Voltage')
                                        xlabel('time (s)') % x-axis label
                                        ylabel('V*s') % y-axis label

                                choice = MFquestdlg('Would you like to try
another linear regression?', ...
                                'Would you like to try another linear
regression?','Linear Regression', ...
                                'Yes', 'No Thank You', 'No Thank You');
                                switch choice
                                    case 'Yes'
                                        true = 10;
                                case 'No Thank You'
                                        true = 0;
                                end
                            end
                    end
choice = MFquestdlg('Would you like to save the Corrected Integrated
Voltage?', ...
'Would you like to save the Corrected Integrated Voltage?','Save?','Yes', 'No
Thank You', 'No Thank You');
switch choice
    case 'Yes'
        prompt = {'CSV File Name? (Must end in ".csv") - This may take a
minute'};
        dlg_title = 'File Name?';
        num_lines = 1;
        def = {'intDdot.csv'};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        fname = answer{1};
        structsave = struct('filename',fname,'time',time,'Corr_Int_D-
Volt',CorrD);
        struct2csv(structsave,fname);
    case 'No Thank You'
end

            end
end
close all
end
```

Appendix G: MATLAB Program Associated with Dot Processing [fftfilter.m] (Codes for Performing Fast Fourier Transform)

```
% This program performs the fast Fourier transform and filtering of the data
% in the frequency domain. That is, the spectral analysis for the basic
% processing option.
 function [CorrB,CorrD,time] = fftfilter(CorrB,CorrD,time)

if CorrD ~= 0
    Data = CorrD;
    disp('corrd');
% else if CorrB ~=0
else
        Data = CorrB;
        disp('corrb');
%     end
end

choice = questdlg('Would you like to do a Fast Fourier Transform?', ...
 'Transform', ...
 'Yes','No','No');
switch choice
    case 'No'
    case 'Yes'
true = 1;

choice = questdlg('What frequency range?', ...
 'Transform', ...
 'All Frequencies','Only Positive','Only Positive');
switch choice
    case 'All Frequencies'
        frange = 0;
    case 'Only Positive'
        frange = 1;
end

NS = length(time);

choice = questdlg('Type of resoltuion? (Higher Resolution may take a longer
time to filter)', ...
 'Transform resolution', ...
 'Normal','High','Ultra High','Ultra High');
switch choice
    case 'Normal'
        N = 2^nextpow2(2^nextpow2(NS)+1)+1;
    case 'High'
        N = (2^nextpow2(2^nextpow2(2^nextpow2(NS)+1)+1)+1);
    case 'Ultra High'
        N = (2^nextpow2(2^nextpow2(2^nextpow2(2^nextpow2(NS)+1)+1)+1)+1);
end

lz = N - NS;
dt = time(2)-time(1);
Fs = 1/dt;
z = zeros(lz,1,'single');
t = (time(NS)+dt):dt:(length(z)*dt + time(NS));
t = t';
etime = [time;t];
Data = padarray(Data,lz,'post');
```

```
while true == 1

freq = (Fs/2)*linspace(-1,1,N);
Y = fft(Data);
XR = real(fftshift(Y));
XI = imag(fftshift(Y));
X = fftshift(Y);
phase = atan((XI)./(XR));

endplot = length(freq);
start = (endplot + 1)/2;
if frange == 0;
    freqplot = freq;
    XRplot = XR;
    XIplot = XI;
    Xplot = X;
    phaseplot = phase;
else if frange == 1;
        freqplot = freq(start:endplot);
        XRplot = XR(start:endplot);
        XIplot = XI(start:endplot);
        Xplot = X(start:endplot);
        phaseplot = phase(start:endplot);
    end
end

figure
set(gcf, 'Position', [13,250,625,625])
subplot(2,2,1);
plot(freqplot,phaseplot);
grid
title('Phase VS Frequency');
xlabel('Frequency');
ylabel('Spectral Phase');

subplot(2,2,2);
plot(freqplot,XRplot);
grid
title('Real VS Frequency');
xlabel('Frequency');
ylabel('Real-Part of Spectral Signal');

subplot(2,2,3);
plot(freqplot,abs(Xplot));
grid
title('Magnitude VS Frequency');
xlabel('Frequency');
ylabel('Spectral Magnitude')

subplot(2,2,4);
plot(freqplot,XIplot);
grid
title('Imaginary VS Frequency');
xlabel('Frequency');
ylabel('Imaginary-Part of Spectral Signal');
```

```
prompt = {'First Upper Cutoff Frequency','First Lower Cutoff
Frequency','Second Upper Cutoff Frequency','Second Lower Cutoff
Frequency','Enter all zeros above if you do not wish to filter'};
dlg_title = 'Filter';
num_lines = 1;
def =
{'0','0','0','0','***********************************************************
*************'};
options.Resize='on';
options.WindowStyle='normal';
answer=inputdlg(prompt,dlg_title,num_lines,def,options);
hicutoff = str2double(answer{1});
lowcutoff= str2double(answer{2});
shicutoff = str2double(answer{3});
slowcutoff = str2double(answer{4});

filter = 1;

if frange == 0;
for j = 1:N
     if (hicutoff == 0 && lowcutoff == 0) && (shicutoff == 0 && slowcutoff ==
0)
         filter = 0;
         continue
     else if (freq(j) >= lowcutoff && freq(j) <= hicutoff) || (freq(j) >=
slowcutoff && freq(j) <= shicutoff)
             X(j) = X(j);
         else X(j) = 0e-100;
         end
    end
end

else if frange == 1;
for j = 1:N
     if (hicutoff == 0 && lowcutoff == 0) && (shicutoff == 0 && slowcutoff ==
0)
         filter = 0;
         continue
     else if (freq(j) >= lowcutoff && freq(j) <= hicutoff) || (freq(j) >=
slowcutoff && freq(j) <= shicutoff) || (freq(j) <= -lowcutoff && freq(j) >= -
hicutoff) || (freq(j) <= -slowcutoff && freq(j) >= -shicutoff)
             X(j) = X(j);
         else X(j) = 0e-100;
         end
    end
end
    end
end

DataF = ifft(ifftshift(X));
figure
set(gcf, 'Position', [650,250,625,625])
subplot(2,1,1);
grid
plot(etime,Data);
```

```
grid
title('Original Data with padding');
xlabel('Time(s)');
ylabel('Volt*s');

subplot(2,1,2);
grid
plot(time,DataF(1:length(time)));
grid
title('Inverse Fourier Transform (padding removed)')
xlabel('Time(s)');
ylabel('Volt*s');

if filter == 1
XR = real(X);
XI = imag(X);
phase = atan((XI)./(XR));

if frange == 0;
    freqplot = freq;
    XRplot = XR;
    XIplot = XI;
    Xplot = X;
    phaseplot = phase;
else if frange == 1;
        freqplot = freq(start:endplot);
        XRplot = XR(start:endplot);
        XIplot = XI(start:endplot);
        Xplot = X(start:endplot);
        phaseplot = phase(start:endplot);
    end
end

figure
set(gcf, 'Position', [13,250,625,625])
subplot(2,2,1);
plot(freqplot,phaseplot);
grid
title('New Phase VS Frequency');
xlabel('Frequency');
ylabel('Spectral Phase');

subplot(2,2,2);
plot(freqplot,XRplot);
grid
title('New Real VS Frequency');
xlabel('Frequency');
ylabel('Real-Part of Spectral Signal');

subplot(2,2,3);
plot(freqplot,abs(Xplot));
grid
title('New Magnitude VS Frequency');
xlabel('Frequency');
ylabel('Spectral Magnitude')
```

```
subplot(2,2,4);
plot(freqplot,XIplot);
grid
title('New Imaginary VS Frequency');
xlabel('Frequency');
ylabel('Imaginary-Part of Spectral Signal');

else if filter == 0;
    end
end

choice = MFquestdlg('Would you like to make changes to the FFT', ...
'Would you like to make changes to the FFT','FFT changes?','Yes','No','No');
    switch choice
        case 'No'
            true = 0;
        case 'Yes'
            close all
    end
end

if CorrD ~= 0
    disp('corrd2');
    CorrD = real(DataF(1:length(time)));
    CorrB = 0;
    spec = abs(X);
    choice = MFquestdlg('Would you like to save the spectral magnitude?', ...
'Would you like to save the spectral magnitude?','Save?','Yes', 'No Thank
You', 'No Thank You');
switch choice
    case 'Yes'
        prompt = {'CSV File Name? (Must end in ".csv") - This may take a
minute'};
        dlg_title = 'File Name?';
        num_lines = 1;
        def = {'FFT_Ddot.csv'};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        fname = answer{1};
        structsave =
struct('filename',fname,'frequency',freq,'SpectralMagnitude_Ddot',spec);
        struct2csv(structsave,fname);
    case 'No Thank You'
end
    choice = MFquestdlg('Would you like to save the Inverse Fourier
Transform?', ...
'Would you like to save the Inverse Fourier Transform?','Save?','Yes', 'No
Thank You', 'No Thank You');
switch choice
    case 'Yes'
        prompt = {'CSV File Name? (Must end in ".csv") - This may take a
minute'};
        dlg_title = 'File Name?';
        num_lines = 1;
```

```
        def = {'filtered-Ddot.csv'};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        fname = answer{1};
        structsave =
struct('filename',fname,'time',time,'Inv_Fourier_D_Dot',CorrD);
        struct2csv(structsave,fname);
    case 'No Thank You'
end
else if CorrB ~=0
        disp('corrb2');
        CorrB = real(DataF(1:length(time)));
        CorrD = 0;
        spec = abs(X);
        choice = MFquestdlg('Would you like to save the spectral magnitude?',
...
'Would you like to save the spectral magnitude?','Save?','Yes', 'No Thank
You', 'No Thank You');
switch choice
    case 'Yes'
        prompt = {'CSV File Name? (Must end in ".csv") - This may take a
minute'};
        dlg_title = 'File Name?';
        num_lines = 1;
        def = {'FFT-Bdot.csv'};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        fname = answer{1};
        structsave =
struct('filename',fname,'frequency',freq,'SpectralMagnitude_Bdot',spec);
        struct2csv(structsave,fname);
    case 'No Thank You'
end
        choice = MFquestdlg('Would you like to save the Inverse Fourier
Transform?', ...
'Would you like to save the Inverse Fourier Transform?','Save?','Yes', 'No
Thank You', 'No Thank You');
switch choice
    case 'Yes'
        prompt = {'CSV File Name? (Must end in ".csv") - This may take a
minute'};
        dlg_title = 'File Name?';
        num_lines = 1;
        def = {'filtered-Bdot.csv'};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        fname = answer{1};
        structsave =
struct('filename',fname,'time',time,'Inv_Fourier_B_Dot',CorrB);
        struct2csv(structsave,fname);
    case 'No Thank You'
end
    end
end
```

```
end
close all
end
```

Appendix H: MATLAB Program Associated with Dot Processing [inputdlg2.m] (Input Dialogue Program)

```
% This program is associated with creating the dialogue box for request
function Answer=inputdlg2(Prompt, Title, NumLines, DefAns, Resize)
%INPUTDLG Input dialog box.
error(nargchk(0,5,nargin));
error(nargoutchk(0,1,nargout));


if nargin<1
    Prompt='Input:';
end
if ~iscell(Prompt)
    Prompt={Prompt};
end
NumQuest=numel(Prompt);


if nargin<2,
    Title=' ';
end

if nargin<3
    NumLines=1;
end

if nargin<4
    DefAns=cell(NumQuest,1);
    for lp=1:NumQuest
        DefAns{lp}='';
    end
end

if nargin<5
    Resize = 'off';
end
WindowStyle='modal';
Interpreter='none';

Options = struct([]); %#ok
if nargin==5 && isstruct(Resize)
    Options = Resize;
    Resize  = 'off';
    if isfield(Options,'Resize'),      Resize=Options.Resize;
          end
    if isfield(Options,'WindowStyle'), WindowStyle=Options.WindowStyle;
            end
    if isfield(Options,'Interpreter'), Interpreter=Options.Interpreter;
            end
end

[rw,cl]=size(NumLines);
OneVect = ones(NumQuest,1);
```

```
if (rw == 1 & cl == 2) %#ok Handle []
    NumLines=NumLines(OneVect,:);
elseif (rw == 1 & cl == 1) %#ok
    NumLines=NumLines(OneVect);
elseif (rw == 1 & cl == NumQuest) %#ok
    NumLines = NumLines';
elseif (rw ~= NumQuest | cl > 2) %#ok
    error('MATLAB:inputdlg:IncorrectSize', 'NumLines size is incorrect.')
end

if ~iscell(DefAns),
    error('MATLAB:inputdlg:InvalidDefaultAnswer', 'Default Answer must be a
cell array of strings.');
end


FigWidth=175;
FigHeight=100;
FigPos(3:4)=[FigWidth FigHeight];  %#ok
FigColor=get(0,'DefaultUicontrolBackgroundcolor');

InputFig=dialog(                     ...
    'Visible'          ,'off'      , ...
    'KeyPressFcn'      ,@doFigureKeyPress, ...
    'Name'             ,Title      , ...
    'Pointer'          ,'arrow'    , ...
    'Units'            ,'pixels'   , ...
    'UserData'         ,'Cancel'   , ...
    'Tag'              ,Title      , ...
    'HandleVisibility' ,'callback' , ...
    'Color'            ,FigColor   , ...
    'NextPlot'         ,'add'      , ...
    'WindowStyle'      ,WindowStyle, ...
    'DoubleBuffer'     ,'on'       , ...
    'Resize'           ,Resize       ...
    );

DefOffset    = 5;
DefBtnWidth  = 53;
DefBtnHeight = 23;

TextInfo.Units              = 'pixels'   ;
TextInfo.FontSize           = get(0,'FactoryUIControlFontSize');
TextInfo.FontWeight         = get(InputFig,'DefaultTextFontWeight');
TextInfo.HorizontalAlignment= 'left'     ;
TextInfo.HandleVisibility   = 'callback' ;

StInfo=TextInfo;
StInfo.Style              = 'text'  ;
StInfo.BackgroundColor    = FigColor;


EdInfo=StInfo;
EdInfo.FontWeight      = get(InputFig,'DefaultUicontrolFontWeight');
EdInfo.Style           = 'edit';
```

```
EdInfo.BackgroundColor = 'white';

BtnInfo=StInfo;
BtnInfo.FontWeight          = get(InputFig,'DefaultUicontrolFontWeight');
BtnInfo.Style               = 'pushbutton';
BtnInfo.HorizontalAlignment = 'center';

% Add VerticalAlignment here as it is not applicable to the above.
TextInfo.VerticalAlignment  = 'bottom';
TextInfo.Color              = get(0,'FactoryUIControlForegroundColor');


% adjust button height and width
btnMargin=1.4;
ExtControl=uicontrol(InputFig   ,BtnInfo     , ...
                     'String'   ,'OK'        , ...
                     'Visible'  ,'off'         ...
                     );

% BtnYOffset  = DefOffset;
BtnExtent = get(ExtControl,'Extent');
BtnWidth  = max(DefBtnWidth,BtnExtent(3)+8);
BtnHeight = max(DefBtnHeight,BtnExtent(4)*btnMargin);
delete(ExtControl);

% Determine # of lines for all Prompts
TxtWidth=FigWidth-2*DefOffset;
ExtControl=uicontrol(InputFig   ,StInfo     , ...
                     'String'   ,''         , ...
                     'Position' ,[ DefOffset DefOffset 0.96*TxtWidth
BtnHeight ] , ...
                     'Visible'  ,'off'        ...
                     );

WrapQuest=cell(NumQuest,1);
QuestPos=zeros(NumQuest,4);

for ExtLp=1:NumQuest
    if size(NumLines,2)==2
        [WrapQuest{ExtLp},QuestPos(ExtLp,1:4)]= ...
            textwrap(ExtControl,Prompt(ExtLp),NumLines(ExtLp,2));
    else
        [WrapQuest{ExtLp},QuestPos(ExtLp,1:4)]= ...
            textwrap(ExtControl,Prompt(ExtLp),80);
    end
end % for ExtLp

delete(ExtControl);
QuestWidth =QuestPos(:,3);
QuestHeight=QuestPos(:,4);

TxtHeight=QuestHeight(1)/size(WrapQuest{1,1},1);
EditHeight=TxtHeight*NumLines(:,1);
EditHeight(NumLines(:,1)==1)=EditHeight(NumLines(:,1)==1)+4;
```

```
FigHeight=(NumQuest+2)*DefOffset    + ...
          BtnHeight+sum(EditHeight) + ...
          sum(QuestHeight);

TxtXOffset=DefOffset;

QuestYOffset=zeros(NumQuest,1);
EditYOffset=zeros(NumQuest,1);
QuestYOffset(1)=FigHeight-DefOffset-QuestHeight(1);
EditYOffset(1)=QuestYOffset(1)-EditHeight(1);

for YOffLp=2:NumQuest,
    QuestYOffset(YOffLp)=EditYOffset(YOffLp-1)-QuestHeight(YOffLp)-DefOffset;
    EditYOffset(YOffLp)=QuestYOffset(YOffLp)-EditHeight(YOffLp);
end % for YOffLp

QuestHandle=[]; %#ok
EditHandle=[];

AxesHandle=axes('Parent',InputFig,'Position',[0 0 1 1],'Visible','off');

inputWidthSpecified = false;

for lp=1:NumQuest,
    if ~ischar(DefAns{lp}),
        delete(InputFig);
        %error('Default Answer must be a cell array of strings.');
        error('MATLAB:inputdlg:InvalidInput', 'Default Answer must be a cell
array of strings.');
    end

    EditHandle(lp)=uicontrol(InputFig    , ...
                         EdInfo      , ...
                         'Max'        ,NumLines(lp,1)       , ...
                         'Position'   ,[ TxtXOffset EditYOffset(lp)
TxtWidth EditHeight(lp) ], ...
                         'String'     ,DefAns{lp}           , ...
                         'Tag'        ,'Edit',                ...
                          'Callback' ,@doEnter);


    QuestHandle(lp)=text('Parent'     ,AxesHandle, ...
                         TextInfo     , ...
                         'Position'   ,[ TxtXOffset QuestYOffset(lp)], ...
                         'String'     ,WrapQuest{lp}                 , ...
                         'Interpreter',Interpreter                   , ...
                         'Tag'        ,'Quest'                         ...
                         );

    MinWidth = max(QuestWidth(:));
    if (size(NumLines,2) == 2)
        % input field width has been specified.
        inputWidthSpecified = true;
        EditWidth = setcolumnwidth(EditHandle(lp), NumLines(lp,1),
NumLines(lp,2));
```

```
        MinWidth = max(MinWidth, EditWidth);
    end
    FigWidth=max(FigWidth, MinWidth+2*DefOffset);

end % for lp

if ~inputWidthSpecified
    TxtWidth=FigWidth-2*DefOffset;
    for lp=1:NumQuest
        set(EditHandle(lp), 'Position', [TxtXOffset EditYOffset(lp) TxtWidth
EditHeight(lp)]);
    end
end

FigPos=get(InputFig,'Position');

FigWidth=max(FigWidth,2*(BtnWidth+DefOffset)+DefOffset);
FigPos(1)=0;
FigPos(2)=0;
FigPos(3)=FigWidth;
FigPos(4)=FigHeight;

set(InputFig,'Position',getnicedialoglocation(FigPos,get(InputFig,'Units')));

OKHandle=uicontrol(InputFig     ,              ...
                   BtnInfo      , ...
                   'Position'   ,[ FigWidth-2*BtnWidth-2*DefOffset DefOffset
BtnWidth BtnHeight ] , ...
                   'KeyPressFcn',@doControlKeyPress , ...
                   'String'     ,'OK'        , ...
                   'Callback'   ,@doCallback , ...
                   'Tag'        ,'OK'        , ...
                   'UserData'   ,'OK'          ...
                   );

setdefaultbutton(InputFig, OKHandle);

CancelHandle=uicontrol(InputFig     ,              ...
                       BtnInfo      , ...
                       'Position'   ,[ FigWidth-BtnWidth-DefOffset DefOffset
BtnWidth BtnHeight ]           , ...
                       'KeyPressFcn',@doControlKeyPress            , ...
                       'String'     ,'Cancel'    , ...
                       'Callback'   ,@doCallback , ...
                       'Tag'        ,'Cancel'    , ...
                       'UserData'   ,'Cancel'      ...
                       ); %#ok

handles = guihandles(InputFig);
handles.MinFigWidth = FigWidth;
handles.FigHeight   = FigHeight;
handles.TextMargin  = 2*DefOffset;
guidata(InputFig,handles);
set(InputFig,'ResizeFcn', {@doResize, inputWidthSpecified});
```

```
% make sure we are on screen
movegui(InputFig)

% if there is a figure out there and it's modal, we need to be modal too
if ~isempty(gcbf) && strcmp(get(gcbf,'WindowStyle'),'modal')
    set(InputFig,'WindowStyle','modal');
end

set(InputFig,'Visible','on');
drawnow;

if ~isempty(EditHandle)
    uicontrol(EditHandle(1));
end

uiwait(InputFig);

if ishandle(InputFig)
    Answer={};
    if strcmp(get(InputFig,'UserData'),'OK'),
        Answer=cell(NumQuest,1);
        for lp=1:NumQuest,
            Answer(lp)=get(EditHandle(lp),{'String'});
        end
    end
    delete(InputFig);
else
    Answer={};
end

end

function doFigureKeyPress(obj, evd) %#ok
switch(evd.Key)
 case {'return','space'}
  set(gcbf,'UserData','OK');
  uiresume(gcbf);
 case {'escape'}
  delete(gcbf);
end

end

function doControlKeyPress(obj, evd) %#ok
switch(evd.Key)
 case {'return'}
  if ~strcmp(get(obj,'UserData'),'Cancel')
      set(gcbf,'UserData','OK');
      uiresume(gcbf);
  else
      delete(gcbf)
  end
 case 'escape'
  delete(gcbf)
end
```

```
end

function doCallback(obj, evd) %#ok
if ~strcmp(get(obj,'UserData'),'Cancel')
    set(gcbf,'UserData','OK');
    uiresume(gcbf);
else
    delete(gcbf)
end

end

function doEnter(obj, evd) %#ok

h = get(obj,'Parent');
x = get(h,'CurrentCharacter');
if unicode2native(x) == 13
    doCallback(obj,evd);
end

end

function doResize(FigHandle, evd, multicolumn) %#ok
Data=guidata(FigHandle);

resetPos = false;

FigPos = get(FigHandle,'Position');
FigWidth = FigPos(3);
FigHeight = FigPos(4);

if FigWidth < Data.MinFigWidth
    FigWidth  = Data.MinFigWidth;
    FigPos(3) = Data.MinFigWidth;
    resetPos = true;
end

% make sure edit fields use all available space if
% number of columns is not specified in dialog creation.
if ~multicolumn
    for lp = 1:length(Data.Edit)
        EditPos = get(Data.Edit(lp),'Position');
        EditPos(3) = FigWidth - Data.TextMargin;
        set(Data.Edit(lp),'Position',EditPos);
    end
end

if FigHeight ~= Data.FigHeight
    FigPos(4) = Data.FigHeight;
    resetPos = true;
end

if resetPos
```

```
    set(FigHandle,'Position',FigPos);
end

end

% set pixel width given the number of columns
function EditWidth = setcolumnwidth(object, rows, cols)
% Save current Units and String.
old_units = get(object, 'Units');
old_string = get(object, 'String');
old_position = get(object, 'Position');

set(object, 'Units', 'pixels')
set(object, 'String', char(ones(1,cols)*'x'));

new_extent = get(object,'Extent');
if (rows > 1)
    % For multiple rows, allow space for the scrollbar
    new_extent = new_extent + 19; % Width of the scrollbar
end
new_position = old_position;
new_position(3) = new_extent(3) + 1;
set(object, 'Position', new_position);

% reset string and units
set(object, 'String', old_string, 'Units', old_units);

EditWidth = new_extent(3);

end

function figure_size = getnicedialoglocation(figure_size, figure_units)
% adjust the specified figure position to fig nicely over GCBF
% or into the upper 3rd of the screen

%  Copyright 1999-2010 The MathWorks, Inc.

parentHandle = gcbf;
convertData.destinationUnits = figure_units;
if ~isempty(parentHandle)
    % If there is a parent figure
    convertData.hFig = parentHandle;
    convertData.size = get(parentHandle,'Position');
    convertData.sourceUnits = get(parentHandle,'Units');
    c = [];
else
    % If there is no parent figure, use the root's data
    % and create a invisible figure as parent
    convertData.hFig = figure('visible','off');
    convertData.size = get(0,'ScreenSize');
    convertData.sourceUnits = get(0,'Units');
    c = onCleanup(@() close(convertData.hFig));
end

% Get the size of the dialog parent in the dialog units
```

```
container_size = hgconvertunits(convertData.hFig, convertData.size ,...
    convertData.sourceUnits, convertData.destinationUnits,
get(convertData.hFig,'Parent'));

delete(c);

figure_size(1) = container_size(1)  + 1/2*(container_size(3) -
figure_size(3));
figure_size(2) = container_size(2)  + 2/3*(container_size(4) -
figure_size(4));

end

function setdefaultbutton(figHandle, btnHandle)
narginchk(1,2)

if (usejava('awt') == 1)
    % We are running with Java Figures
    useJavaDefaultButton(figHandle, btnHandle)
else
    % We are running with Native Figures
    useHGDefaultButton(figHandle, btnHandle);
end

    function useJavaDefaultButton(figH, btnH)
        % Get a UDD handle for the figure.
        fh = handle(figH);
        % Call the setDefaultButton method on the figure handle
        fh.setDefaultButton(btnH);
    end

    function useHGDefaultButton(figHandle, btnHandle)
        % First get the position of the button.
        btnPos = getpixelposition(btnHandle);

        % Next calculate offsets.
        leftOffset   = btnPos(1) - 1;
        bottomOffset = btnPos(2) - 2;
        widthOffset  = btnPos(3) + 3;
        heightOffset = btnPos(4) + 3;

        h1 = uipanel(get(btnHandle, 'Parent'), 'HighlightColor', 'black', ...
            'BorderType', 'etchedout', 'units', 'pixels', ...
            'Position', [leftOffset bottomOffset widthOffset heightOffset]);

        % Make sure it is stacked on the bottom.
        uistack(h1, 'bottom');
    end
end
```

Appendix I: MATLAB Program Associated with Dot Processing [plotter.m] (Plotter)

```
% This program plots the output of the processed data. It generates the
% magnetic field vs. time. The program also provides the .csv file option to
% to be saved
function [CorrB,CorrD,time] = plotter(CorrB,CorrD,time)

if CorrD ~= 0
    defaultY = 'Electric Field (V/m)';
    defaultL = 'Corrected Electric Field (V/m)';
    else
        defaultY = 'Magnetic Field (Tesla)';
        defaultL = 'Corrected Magnetic Field (Tesla)';

end

        true = 1;
     while true == 1
        prompt = {'Title','X-Axis','Y-Axis','Font','Font Size','Line
Width','Legend'};
        dlg_title = 'Plot Options';
        num_lines = 1;
        def = {'Title','Time',defaultY,'helvetica','20','1',defaultL};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        titlename = answer{1};
        xaxis = answer{2};
        yaxis = answer{3};
        font = answer{4};
        fontsize = str2double(answer{5});
        linewidth = str2double(answer{6});
        leg = answer{7};

            if CorrD ~= 0

            prompt = {'D-Dot Calibration Factor','Amplifier Gain'};
            dlg_title = 'Calibration';
            num_lines = 1;
            def = {'1','1'};
            options.Resize='on';
            options.WindowStyle='normal';
            answer=inputdlg(prompt,dlg_title,num_lines,def,options);
            Dcalnum = (str2double(answer{1})/str2double(answer{2}));


            figure;
            pause(0.0000001);
            frame_h = get(handle(gcf),'JavaFrame');
            set(frame_h,'Maximized',1);
            CorrD = CorrD * Dcalnum;
            plot(time, CorrD, 'r')
            grid
            legend(leg)
            title(titlename);
```

```
            xlabel(xaxis);
            ylabel(yaxis);
            set(get(gca,'children'),'linewidth',linewidth);
            set(gca,'linewidth',linewidth,'gridlinestyle','-
','fontname',font,'fontweight','bold','fontsize',fontsize);
            set(gca,'position',[0.1005,0.0905,.80,.85],'ticklength',[0,0]);
set(get(gca,'xlabel'),'fontname',font,'fontweight','bold','fontsize',fontsize
)
set(get(gca,'ylabel'),'fontname',font,'fontweight','bold','fontsize',fontsize
)
set(get(gca,'title'),'fontname',font,'fontweight','bold','fontsize',fontsize)

            choice = MFquestdlg('Would you like to save the Corrected
Electric Field (V/m)?', ...
        'Would you like to save the Corrected Electric Field
(V/m)?','Save?','Yes', 'No Thank You', 'No Thank You');
               switch choice
                    case 'Yes'
                prompt = {'CSV File Name? (Must end in ".csv") - This may
take a minute'};
                dlg_title = 'File Name?';
                num_lines = 1;
                def = {'E-Field.csv'};
                options.Resize='on';
                options.WindowStyle='normal';
                answer=inputdlg(prompt,dlg_title,num_lines,def,options);
                fname = answer{1};
                structsave =
struct('filename',fname,'time',time,'Corr_EField',CorrD);
                struct2csv(structsave,fname);
                    case 'No Thank You'
                end

            else
                prompt = {'B-Dot Calibration Factor','Amplifier Gain'};
                dlg_title = 'Calibration';
                num_lines = 1;
                def = {'1','1'};
                options.Resize='on';
                options.WindowStyle='normal';
                answer=inputdlg(prompt,dlg_title,num_lines,def,options);
                Bcalnum = (str2double(answer{1})/str2double(answer{2}));

                figure;
                pause(0.00001);
                frame_h = get(handle(gcf),'JavaFrame');
                set(frame_h,'Maximized',1);
                CorrB = CorrB * Bcalnum;
                plot(time, CorrB, 'r')
                grid
                legend(leg)
                title(titlename);
                xlabel(xaxis);
                ylabel(yaxis);
                set(get(gca,'children'),'linewidth',linewidth);
```

```
                set(gca,'linewidth',linewidth,'gridlinestyle','-
','fontname',font,'fontweight','bold','fontsize',fontsize);

set(gca,'position',[0.1005,0.0905,.80,.85],'ticklength',[0,0]);
set(get(gca,'xlabel'),'fontname',font,'fontweight','bold','fontsize',fontsize
)
set(get(gca,'ylabel'),'fontname',font,'fontweight','bold','fontsize',fontsize
)
set(get(gca,'title'),'fontname',font,'fontweight','bold','fontsize',fontsize)

                choice = MFquestdlg('Would you like to save the Corrected
Magnetic Field (Tesla)?', ...
        'Would you like to save the Corrected Magnetic Field
(Tesla)?','Save?','Yes', 'No Thank You', 'No Thank You');
                switch choice
                    case 'Yes'
                prompt = {'CSV File Name? (Must end in ".csv") - This may
take a minute'};
                dlg_title = 'File Name?';
                num_lines = 1;
                def = {'B-Field.csv'};
                options.Resize='on';
                options.WindowStyle='normal';
                answer=inputdlg(prompt,dlg_title,num_lines,def,options);
                fname = answer{1};
                structsave =
struct('filename',fname,'time',time,'Corr_BField',CorrB);
                struct2csv(structsave,fname);
                    case 'No Thank You'
                end
                    end
      end

        choice = MFquestdlg('Would you like to re-edit the labels?',...
        'Would you like to re-edit the labels?','Plot Editor',...
        'Yes', 'No','No');
         switch choice
             case 'Yes'
                 true = 1;
             case 'No'
                 true = 0;
         end
end
```

Appendix J: MATLAB Program Associated with Dot Processing [splotter.m] (Spectral Plotter)

```
% This is also a plot routine in the frequency domain after the fft has been
performed on the data.
function [CorrB,CorrD,time] = splotter(CorrB,CorrD,time)

if CorrD ~= 0
    defaultY = 'Electric Field (V/m)';
    defaultL = 'Corrected Electric Field (V/m)';
else
        defaultY = 'Magnetic Field (Tesla)';
        defaultL = 'Corrected Magnetic Field (Tesla)';

end

        true = 1;
        while true == 1
        prompt = {'Title','X-Axis','Y-Axis','Font','Font Size','Line
Width','Legend'};
        dlg_title = 'Plot Options';
        num_lines = 1;
        def = {'Title','Time',defaultY,'helvetica','20','1',defaultL};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        titlename = answer{1};
        xaxis = answer{2};
        yaxis = answer{3};
        font = answer{4};
        fontsize = str2double(answer{5});
        linewidth = str2double(answer{6});
        leg = answer{7};

        if CorrD ~= 0

            prompt = {'Amplifier Gain'};
            dlg_title = 'Calibration';
            num_lines = 1;
            def = {'1','1'};
            options.Resize='on';
            options.WindowStyle='normal';
            answer=inputdlg(prompt,dlg_title,num_lines,def,options);
            Dcalnum = 1/str2double(answer{1});


            figure;
            pause(0.0000001);
            frame_h = get(handle(gcf),'JavaFrame');
            set(frame_h,'Maximized',1);
            CorrD = CorrD * Dcalnum;
            plot(time, CorrD, 'r')
            grid
            legend(leg)
```

```
            title(titlename);
            xlabel(xaxis);
            ylabel(yaxis);
            set(get(gca,'children'),'linewidth',linewidth);
            set(gca,'linewidth',linewidth,'gridlinestyle','-
','fontname',font,'fontweight','bold','fontsize',fontsize);
            set(gca,'position',[0.1005,0.0905,.80,.85],'ticklength',[0,0]);

set(get(gca,'xlabel'),'fontname',font,'fontweight','bold','fontsize',fontsize
)
set(get(gca,'ylabel'),'fontname',font,'fontweight','bold','fontsize',fontsize
)
set(get(gca,'title'),'fontname',font,'fontweight','bold','fontsize',fontsize)
            choice = MFquestdlg('Would you like to save the Corrected
Electric Field (V/m)?', ...
        'Would you like to save the Corrected Electric Field
(V/m)?','Save?','Yes', 'No Thank You', 'No Thank You');
        switch choice
            case 'Yes'
                prompt = {'CSV File Name? (Must end in ".csv") - This may
take a minute'};
                dlg_title = 'File Name?';
                num_lines = 1;
                def = {'E-Field.csv'};
                options.Resize='on';
                options.WindowStyle='normal';
                answer=inputdlg(prompt,dlg_title,num_lines,def,options);
                fname = answer{1};
                structsave =
struct('filename',fname,'time',time,'Corr_EField',CorrD);
                struct2csv(structsave,fname);
            case 'No Thank You'
        end

        else
                prompt = {'Amplifier Gain'};
                dlg_title = 'Calibration';
                num_lines = 1;
                def = {'1','1'};
                options.Resize='on';
                options.WindowStyle='normal';
                answer=inputdlg(prompt,dlg_title,num_lines,def,options);
                Bcalnum = 1/str2double(answer{1});

                figure;
                pause(0.00001);
                frame_h = get(handle(gcf),'JavaFrame');
                set(frame_h,'Maximized',1);
                CorrB = CorrB * Bcalnum;
                plot(time, CorrB, 'r')
                grid
                legend(leg)
                title(titlename);
                xlabel(xaxis);
                ylabel(yaxis);
                set(get(gca,'children'),'linewidth',linewidth);
```

```
                set(gca,'linewidth',linewidth,'gridlinestyle','-
','fontname',font,'fontweight','bold','fontsize',fontsize);

set(gca,'position',[0.1005,0.0905,.80,.85],'ticklength',[0,0]);
set(get(gca,'xlabel'),'fontname',font,'fontweight','bold','fontsize',fontsize
)
set(get(gca,'ylabel'),'fontname',font,'fontweight','bold','fontsize',fontsize
)
set(get(gca,'title'),'fontname',font,'fontweight','bold','fontsize',fontsize)

                choice = MFquestdlg('Would you like to save the Corrected
Magnetic Field (Tesla)?', ...
        'Would you like to save the Corrected Magnetic Field
(Tesla)?','Save?','Yes', 'No Thank You', 'No Thank You');
        switch choice
            case 'Yes'
                prompt = {'CSV File Name? (Must end in ".csv") - This may
take a minute'};
                dlg_title = 'File Name?';
                num_lines = 1;
                def = {'B-Field.csv'};
                options.Resize='on';
                options.WindowStyle='normal';
                answer=inputdlg(prompt,dlg_title,num_lines,def,options);
                fname = answer{1};
                structsave =
struct('filename',fname,'time',time,'Corr_BField',CorrB);
                struct2csv(structsave,fname);
            case 'No Thank You'
        end
            end
        end
        choice = MFquestdlg('Would you like to re-edit the labels?',...
        'Would you like to re-edit the labels?','Plot Editor',...
        'Yes', 'No','No');
         switch choice
             case 'Yes'
                 true = 1;
             case 'No'
                 true = 0;
         end
       end
```

Appendix K: MATLAB Program Associated with Dot Processing [spectralprocessing.m] (Spectral Processing file)

```
% This program performs the fast Fourier transform and filtering of the data
% in the frequency domain. That is, the spectral analysis for the basic
% processing option.
function [CorrB,CorrD,time] = spectralprocessing(CorrB,CorrD,time)

close all
if CorrD ~= 0
    Data2 = CorrD;
    disp('corrd');
else
        Data2 = CorrB;
        disp('corrb');

end
time2 = time;
figure
plot(time2,Data2,'r');
grid
title('Experimental');
xlabel('Time(s)');
ylabel('Volt*s');
legend('Experimental')

choice = questdlg('What frequency range for fft?', ...
'Transform', ...
'All Frequencies','Only Positive','Only Positive');
switch choice
    case 'All Frequencies'
    frange = 0;
    case 'Only Positive'
    frange = 1;
end

NS2 = length(time2);

choice = questdlg('Type of resoltuion? (Higher Resolution may take a longer
time to filter)', ...
 'Transform resolution', ...
 'Normal','High','Ultra High','Ultra High');
switch choice
    case 'Normal'
        N2 = 2^nextpow2(2^nextpow2(NS2)+1)+1;
    case 'High'
        N2 = (2^nextpow2(2^nextpow2(2^nextpow2(NS2)+1)+1)+1);
    case 'Ultra High'
        N2 = (2^nextpow2(2^nextpow2(2^nextpow2(2^nextpow2(NS2)+1)+1)+1)+1);
end

dt2 = time2(2)-time2(1);
Fs2 = 1/dt2;
lz2 = N2 - NS2;
```

```
z2 = zeros(lz2,1,'single');
t2 = (time2(NS2)+dt2):dt2:(length(z2)*dt2 + time2(NS2));
t2 = t2';
etime2 = [time2;t2];
eData2 = padarray(Data2,lz2,'post');
true = 0;

while true == 0;

    freq = (Fs2/2)*linspace(-1, 1, N2);
    Y = fft(eData2);
    XR = real(fftshift(Y));
    XI = imag(fftshift(Y));
    X = fftshift(Y);
    phase = atan((XI)./(XR));

    endplot = length(freq);
    start = (endplot + 1)/2;
    if frange == 0;
        freqplot = freq;
        XRplot = XR;
        XIplot = XI;
        Xplot = X;
        phaseplot = phase;
    else if frange == 1;
            freqplot = freq(start:endplot);
            XRplot = XR(start:endplot);
            XIplot = XI(start:endplot);
            Xplot = X(start:endplot);
            phaseplot = phase(start:endplot);
        end
    end

    figure
    set(gcf, 'Position', [13,250,625,625])
    subplot(2,2,1);
    plot(freqplot,phaseplot);
    grid
    title('Phase VS Frequency');
    xlabel('Frequency');
    ylabel('Spectral Phase');

    subplot(2,2,2);
    plot(freqplot,XRplot);
    grid
    title('Real VS Frequency');
    xlabel('Frequency');
    ylabel('Real-Part of Spectral Signal');

    subplot(2,2,3);
    plot(freqplot,abs(Xplot));
    grid
    title('Magnitude VS Frequency');
    xlabel('Frequency');
    ylabel('Spectral Magnitude')
```

```
    subplot(2,2,4);
    plot(freqplot,XIplot);
    grid
    title('Imaginary VS Frequency');
    xlabel('Frequency');
    ylabel('Imaginary-Part of Spectral Signal');

    prompt = {'First Upper Cutoff Frequency','First Lower Cutoff
Frequency','Second Upper Cutoff Frequency','Second Lower Cutoff
Frequency','Enter all zeros above if you do not wish to filter'};
    dlg_title = 'Filter';
    num_lines = 1;
    def =
{'0','0','0','0','************************************************************
***'};
    options.Resize='on';
    options.WindowStyle='normal';
    answer=inputdlg(prompt,dlg_title,num_lines,def,options);
    hicutoff = str2double(answer{1});
    lowcutoff= str2double(answer{2});
    shicutoff = str2double(answer{3});
    slowcutoff = str2double(answer{4});


    filter = 1;

    if frange == 0;
    for j = 1:N2
         if (hicutoff == 0 && lowcutoff == 0) && (shicutoff == 0 &&
slowcutoff == 0)
             filter = 0;
             continue
         else if (freq(j) >= lowcutoff && freq(j) <= hicutoff) || (freq(j) >=
slowcutoff && freq(j) <= shicutoff)
                 X(j) = X(j);
             else X(j) = 0e-100;
             end
        end
    end

    else if frange == 1;
    for j = 1:N2
         if (hicutoff == 0 && lowcutoff == 0) && (shicutoff == 0 &&
slowcutoff == 0)
             filter = 0;
             continue
         else if (freq(j) >= lowcutoff && freq(j) <= hicutoff) || (freq(j) >=
slowcutoff && freq(j) <= shicutoff) || (freq(j) <= -lowcutoff && freq(j) >= -
hicutoff) || (freq(j) <= -slowcutoff && freq(j) >= -shicutoff)
                 X(j) = X(j);
             else X(j) = 0e-100;
             end
         end
    end
        end
```

```
    end


   choice = MFquestdlg('Would you like to mulitply by calibration function?',
...
    'Would you like to mulitply by calibration
function?','Calibration?','Yes','No','No');
switch choice
    %{
    case 'Polynomial'
        [poly,freqz] = spectralfunction();
        H = polyval(poly,freq);
        lengthH = length(H);
        for k = 1:lengthH
            if H(k) > (freqz(length(freqz))) || H(k) < (freqz(1))
                H(k) = 0;
            end
        end
        X = X.*H'; %redefines X for rest of program (change to z to stop)
        Data2F = ifft(ifftshift(X));
    %}
    case 'Yes'
        [H] = linesegment(freq);
        X = X.*H';
        Data2F = ifft(ifftshift(X));
        t = 1;
        ylab = 'Volts';
    case 'No'
        Data2F = ifft(ifftshift(X));
        t = 0;
        ylab = 'Volts*s';
end

    figure
    set(gcf, 'Position', [650,250,625,625])
    subplot(2,1,1);
    plot(etime2,eData2);
    grid
    title('Original Data with padding');
    xlabel('Time(s)');
    ylabel('Volts*s');
    subplot(2,1,2);
    grid
    plot(time2,Data2F(1:length(time2)));
    grid
    title('Inverse Fourier Transform (padding removed)');
    xlabel('Time(s)');
    ylabel(ylab);

    if filter == 1
    XR = real(X);
    XI = imag(X);
    phase = atan((XI)./(XR));

    if frange == 0;
    freqplot = freq;
```

```
    XRplot = XR;
    XIplot = XI;
    Xplot = X;
    phaseplot = phase;
else if frange == 1;
        freqplot = freq(start:endplot);
        XRplot = XR(start:endplot);
        XIplot = XI(start:endplot);
        Xplot = X(start:endplot);
        phaseplot = phase(start:endplot);
    end
    end

    figure
    set(gcf, 'Position', [13,250,625,625])
    subplot(2,2,1);
    plot(freqplot,phaseplot);
    grid
    title('New Phase VS Frequency');
    xlabel('Frequency');
    ylabel('Spectral Phase');

    subplot(2,2,2);
    plot(freqplot,XRplot);
    grid
    title('New Real VS Frequency');
    xlabel('Frequency');
    ylabel('Real-Part of Spectral Signal');

    subplot(2,2,3);
    plot(freqplot,abs(Xplot));
    grid
    title('New Magnitude VS Frequency');
    xlabel('Frequency');
    ylabel('Spectral Magnitude')

    subplot(2,2,4);
    plot(freqplot,XIplot);
    grid
    title('New Imaginary VS Frequency');
    xlabel('Frequency');
    ylabel('Imaginary-Part of Spectral Signal');

    else if filter == 0;
        end
    end

    choice = MFquestdlg('Would you like to make changes to the FFT', ...
    'Would you like to make changes to the FFT','FFT
changes?','Yes','No','No');
        switch choice
            case 'No'
                true = 1;
            case 'Yes'
                close all
                t = 1;
```

```
        end

if t == 0;
choice = MFquestdlg('Would you like to save the Inverse Fourier Transform
(padding removed)?', ...
'Would you like to save the Inverse Fourier Transform (padding
removed)?','Save?','Yes', 'No Thank You', 'No Thank You');
switch choice
    case 'Yes'
        prompt = {'CSV File Name? (Must end in ".csv") - This may take a
minute'};
        dlg_title = 'File Name?';
        num_lines = 1;
        def = {'inverseFourier.csv'};
        options.Resize='on';
        options.WindowStyle='normal';
        answer=inputdlg(prompt,dlg_title,num_lines,def,options);
        fname = answer{1};
        structsave =
struct('filename',fname,'time',time2,'InverseFourier',Data2F(1:length(time2))
);
        struct2csv(structsave,fname);
    case 'No Thank You'
end
end

 if t == 1;
     if CorrD ~= 0
        CorrD = Data2F(1:length(time2));
%      elseif CorrB ~=0
     else
        CorrB = Data2F(1:length(time2));
    end
     end
     splotter(CorrB,CorrD,time);
%  else
 end
end
```

Appendix L: MATLAB Program Associated with Dot Processing [MFquestdlg.m] (Input Dialogue box file)

```
 % This program controls the properties of the dialogue box

function ButtonName=MFquestdlg(Pos,Question,Title,Btn1,Btn2,Btn3,Default)
if nargin<1
  error('MATLAB:questdlg:TooFewArguments', 'Too few arguments for QUESTDLG');
end

Interpreter='none';
if ~iscell(Question),Question=cellstr(Question);
end
Black      =[0        0         0      ]/255;
if nargout>1
  error('MATLAB:questdlg:WrongNumberOutputs', 'Wrong number of output
arguments for QUESTDLG');
end
if nargin==2,Title=' ';
end
if nargin<=3, Default='Yes';
end
if nargin==4, Default=Btn1 ;
end
if nargin<=4, Btn1='Yes'; Btn2='No'; Btn3='Cancel';NumButtons=3;
end
if nargin==5, Default=Btn2;Btn2=[];Btn3=[];NumButtons=1;
end
if nargin==6, Default=Btn3;Btn3=[];NumButtons=2;
end
if nargin==7, NumButtons=3;
end
if nargin>7
  error('MATLAB:questdlg:TooManyInputs', 'Too many input
arguments');NumButtons=3; %#ok
end

if isstruct(Default),
  Interpreter=Default.Interpreter;
  Default=Default.Default;
end

set(0,'Units','pixels');  % <<<<<<<<<<----------MODIFIED
scnsize=get(0,'ScreenSize');  % <<<<<<<<<<----------MODIFIED
% FigPos    = get(0,'DefaultFigurePosition'); % <<<<<<<<<<----------MODIFIED
FigPos(1)=Pos(1)*scnsize(3);  % <<<<<<<<<<----------MODIFIED
FigPos(2)=Pos(2)*scnsize(4);  % <<<<<<<<<<----------MODIFIED
FigPos(3) = 267;
FigPos(4) =  70;

QuestFig=dialog(                                    ...
  'Visible'         ,'off'                      , ...
  'Name'            ,Title                      , ...
  'Pointer'         ,'arrow'                    , ...
  'Position'        ,FigPos                     , ...
```

```
  'KeyPressFcn'      ,@doFigureKeyPress            , ...
  'IntegerHandle'    ,'off'                        , ...
  'WindowStyle'      ,'normal'                     , ...
  'HandleVisibility','callback'                   , ...
  'CloseRequestFcn' ,@doDelete                     , ...
  'Tag'              ,Title                          ...
  );

DefOffset  =10;

IconWidth  =54;
IconHeight =54;
IconXOffset=DefOffset;
IconYOffset=FigPos(4)-DefOffset-IconHeight;  %#ok
IconCMap=[Black;get(QuestFig,'Color')];  %#ok

DefBtnWidth =56;
BtnHeight   =22;

BtnYOffset=DefOffset;

BtnWidth=DefBtnWidth;

ExtControl=uicontrol(QuestFig   , ...
  'Style'    ,'pushbutton', ...
  'String'   ,' '          ...
  );

btnMargin=1.4;
set(ExtControl,'String',Btn1);
BtnExtent=get(ExtControl,'Extent');
BtnWidth=max(BtnWidth,BtnExtent(3)+8);
if NumButtons > 1
  set(ExtControl,'String',Btn2);
  BtnExtent=get(ExtControl,'Extent');
  BtnWidth=max(BtnWidth,BtnExtent(3)+8);
  if NumButtons > 2
    set(ExtControl,'String',Btn3);
    BtnExtent=get(ExtControl,'Extent');
    BtnWidth=max(BtnWidth,BtnExtent(3)*btnMargin);
  end
end
BtnHeight = max(BtnHeight,BtnExtent(4)*btnMargin);

delete(ExtControl);

MsgTxtXOffset=IconXOffset+IconWidth;

FigPos(3)=max(FigPos(3),MsgTxtXOffset+NumButtons*(BtnWidth+2*DefOffset));
set(QuestFig,'Position',FigPos);

BtnXOffset=zeros(NumButtons,1);

if NumButtons==1,
```

```
  BtnXOffset=(FigPos(3)-BtnWidth)/2;
elseif NumButtons==2,
  BtnXOffset=[MsgTxtXOffset
    FigPos(3)-DefOffset-BtnWidth];
elseif NumButtons==3,
  BtnXOffset=[MsgTxtXOffset
    0
    FigPos(3)-DefOffset-BtnWidth];
  BtnXOffset(2)=(BtnXOffset(1)+BtnXOffset(3))/2;
end

MsgTxtYOffset=DefOffset+BtnYOffset+BtnHeight;
% Calculate current msg text width and height. If negative,
% clamp it to 1 since its going to be recalculated/corrected later
% based on the actual msg string
MsgTxtWidth=max(1, FigPos(3)-DefOffset-MsgTxtXOffset-IconWidth);
MsgTxtHeight=max(1, FigPos(4)-DefOffset-MsgTxtYOffset);

MsgTxtForeClr=Black;
MsgTxtBackClr=get(QuestFig,'Color');

CBString='uiresume(gcbf)';
DefaultValid = false;
DefaultWasPressed = false;
BtnHandle = cell(NumButtons, 1);
DefaultButton = 0;

% Check to see if the Default string passed does match one of the
% strings on the buttons in the dialog. If not, throw a warning.
for i = 1:NumButtons
  switch i
    case 1
      ButtonString=Btn1;
      ButtonTag='Btn1';
      if strcmp(ButtonString, Default)
        DefaultValid = true;
        DefaultButton = 1;
      end

    case 2
      ButtonString=Btn2;
      ButtonTag='Btn2';
      if strcmp(ButtonString, Default)
        DefaultValid = true;
        DefaultButton = 2;
      end
    case 3
      ButtonString=Btn3;
      ButtonTag='Btn3';
      if strcmp(ButtonString, Default)
        DefaultValid = true;
        DefaultButton = 3;
      end
  end

  BtnHandle{i}=uicontrol(QuestFig            , ...
```

```
    'Style'              ,'pushbutton', ...
    'Position'           ,[ BtnXOffset(1) BtnYOffset BtnWidth BtnHeight ]
, ...
    'KeyPressFcn'        ,@doControlKeyPress , ...
    'CallBack'           ,CBString    , ...
    'String'             ,ButtonString, ...
    'HorizontalAlignment','center'    , ...
    'Tag'                ,ButtonTag     ...
    );
end

if ~DefaultValid
  warnstate = warning('backtrace','off');
  warning('MATLAB:QUESTDLG:stringMismatch','Default string does not match any
button string name.');
  warning(warnstate);
end

MsgHandle=uicontrol(QuestFig            , ...
  'Style'              ,'text'         , ...
  'Position'           ,[MsgTxtXOffset MsgTxtYOffset 0.95*MsgTxtWidth
MsgTxtHeight ]              , ...
  'String'             ,{' '}          , ...
  'Tag'                ,'Question'     , ...
  'HorizontalAlignment','left'         , ...
  'FontWeight'         ,'bold'         , ...
  'BackgroundColor'    ,MsgTxtBackClr  , ...
  'ForegroundColor'    ,MsgTxtForeClr    ...
  );

[WrapString,NewMsgTxtPos]=textwrap(MsgHandle,Question,75);

% NumLines=size(WrapString,1);

AxesHandle=axes('Parent',QuestFig,'Position',[0 0 1 1],'Visible','off');

texthandle=text( ...
    'Parent'              ,AxesHandle                      , ...
    'Units'               ,'pixels'                        , ...
    'Color'               ,get(BtnHandle{1},'ForegroundColor')   , ...
    'HorizontalAlignment' ,'left'                          , ...
    'FontName'            ,get(BtnHandle{1},'FontName')    , ...
    'FontSize'            ,get(BtnHandle{1},'FontSize')    , ...
    'VerticalAlignment'   ,'bottom'                        , ...
    'String'              ,WrapString                      , ...
    'Interpreter'         ,Interpreter                     , ...
    'Tag'                 ,'Question'                        ...
    );

textExtent = get(texthandle, 'extent');

% (g357851)textExtent and extent from uicontrol are not the same. For window,
extent from uicontrol is larger
%than textExtent. But on Mac, it is reverse. Pick the max value.
MsgTxtWidth=max([MsgTxtWidth NewMsgTxtPos(3)+2 textExtent(3)]);
```

```
MsgTxtHeight=max([MsgTxtHeight NewMsgTxtPos(4)+2 textExtent(4)]);

MsgTxtXOffset=IconXOffset+IconWidth+DefOffset;
FigPos(3)=max(NumButtons*(BtnWidth+DefOffset)+DefOffset, ...
  MsgTxtXOffset+MsgTxtWidth+DefOffset);


% Center Vertically around icon
if IconHeight>MsgTxtHeight,
  IconYOffset=BtnYOffset+BtnHeight+DefOffset;
  MsgTxtYOffset=IconYOffset+(IconHeight-MsgTxtHeight)/2;
  FigPos(4)=IconYOffset+IconHeight+DefOffset;
  % center around text
else
  MsgTxtYOffset=BtnYOffset+BtnHeight+DefOffset;
  IconYOffset=MsgTxtYOffset+(MsgTxtHeight-IconHeight)/2;
  FigPos(4)=MsgTxtYOffset+MsgTxtHeight+DefOffset;
end

if NumButtons==1,
  BtnXOffset=(FigPos(3)-BtnWidth)/2;
elseif NumButtons==2,
  BtnXOffset=[(FigPos(3)-DefOffset)/2-BtnWidth
    (FigPos(3)+DefOffset)/2
    ];

elseif NumButtons==3,
  BtnXOffset(2)=(FigPos(3)-BtnWidth)/2;
  BtnXOffset=[BtnXOffset(2)-DefOffset-BtnWidth
    BtnXOffset(2)
    BtnXOffset(2)+BtnWidth+DefOffset
    ];
end
set(QuestFig ,'Position',FigPos); % <<<<<<<<<<----------MODIFIED
% set(QuestFig ,'Position',getnicedialoglocation(FigPos,
get(QuestFig,'Units'))); % <<<<<<<<<<----------MODIFIED
assert(iscell(BtnHandle));
BtnPos=cellfun(@(bh)get(bh,'Position'), BtnHandle, 'UniformOutput', false);
BtnPos=cat(1,BtnPos{:});
BtnPos(:,1)=BtnXOffset;
BtnPos=num2cell(BtnPos,2);

assert(iscell(BtnPos));
cellfun(@(bh,pos)set(bh, 'Position', pos), BtnHandle, BtnPos,
'UniformOutput', false);

if DefaultValid
%   setdefaultbutton(QuestFig, BtnHandle{DefaultButton}); % <<<<<<<<<<-------
---MODIFIED
end

delete(MsgHandle);


set(texthandle, 'Position',[MsgTxtXOffset MsgTxtYOffset 0]);
```

```
IconAxes=axes(                                              ...
  'Parent'      ,QuestFig              , ...
  'Units'       ,'Pixels'              , ...
  'Position'    ,[IconXOffset IconYOffset IconWidth IconHeight], ...
  'NextPlot'    ,'replace'             , ...
  'Tag'         ,'IconAxes'              ...
  );

set(QuestFig ,'NextPlot','add');

load dialogicons.mat questIconData questIconMap;
IconData=questIconData;
questIconMap(256,:)=get(QuestFig,'color');
IconCMap=questIconMap;

Img=image('CData',IconData,'Parent',IconAxes);
set(QuestFig, 'Colormap', IconCMap);
set(IconAxes, ...
  'Visible','off'           , ...
  'YDir'   ,'reverse'       , ...
  'XLim'   ,get(Img,'XData'), ...
  'YLim'   ,get(Img,'YData')  ...
  );

% make sure we are on screen
movegui(QuestFig)


set(QuestFig ,'Visible','on');
drawnow;

if DefaultButton ~= 0
  uicontrol(BtnHandle{DefaultButton});
end

if ishghandle(QuestFig)
  % Go into uiwait if the figure handle is still valid.
  % This is mostly the case during regular use.
  uiwait(QuestFig);
end

% Check handle validity again since we may be out of uiwait because the
% figure was deleted.
if ishghandle(QuestFig)
  if DefaultWasPressed
    ButtonName=Default;
  else
    ButtonName=get(get(QuestFig,'CurrentObject'),'String');
  end
  doDelete;
else
  ButtonName='';
end
```

```
  function doFigureKeyPress(obj, evd)  %#ok
    switch(evd.Key)
      case {'return','space'}
        if DefaultValid
          DefaultWasPressed = true;
          uiresume(gcbf);
        end
      case 'escape'
        doDelete
    end
  end

  function doControlKeyPress(obj, evd)  %#ok
    switch(evd.Key)
      case {'return'}
        if DefaultValid
          DefaultWasPressed = true;
          uiresume(gcbf);
        end
      case 'escape'
        doDelete
    end
  end

  function doDelete(varargin)
    delete(QuestFig);
  end
end
```

Appendix M: MATLAB Program Associated with Dot Processing [struct2csv.m] (Structure to .CSV file Program)

```
% This program generates a .csv file from the Matlab structures.
function struct2csv(s,fn)
FID = fopen(fn,'w');
headers = fieldnames(s);
m = length(headers);
sz = zeros(m,2);

t = length(s);

for rr = 1:t
    l = '';
    for ii = 1:m
        sz(ii,:) = size(s(rr).(headers{ii}));
        if ischar(s(rr).(headers{ii}))
            sz(ii,2) = 1;
        end
        l = [l,'"',headers{ii},'",',repmat(',',1,sz(ii,2)-1)];
    end

    l = [l,'\n'];

    fprintf(FID,l);

    n = max(sz(:,1));

    for ii = 1:n
        l = '';
        for jj = 1:m
            c = s(rr).(headers{jj});
            str = '';

            if sz(jj,1)<ii
                str = repmat(',',1,sz(jj,2));
            else
                if isnumeric(c)
                    for kk = 1:sz(jj,2)
                        str = [str,num2str(c(ii,kk)),','];
                    end
                elseif islogical(c)
                    for kk = 1:sz(jj,2)
                        str = [str,num2str(double(c(ii,kk))),','];
                    end
                elseif ischar(c)
                    str = ['"',c(ii,:),'",'];
                elseif iscell(c)
                    if isnumeric(c{1,1})
                        for kk = 1:sz(jj,2)
                            str = [str,num2str(c{ii,kk}),','];
                        end
                    elseif islogical(c{1,1})
                        for kk = 1:sz(jj,2)
```

```
                            str = [str,num2str(double(c{ii,kk})),','];
                        end
                    elseif ischar(c{1,1})
                        for kk = 1:sz(jj,2)
                            str = [str,'"',c{ii,kk},'",'];
                        end
                    end
                end
            end
            l = [l,str];
        end
        l = [l,'\n'];
        fprintf(FID,l);
    end
    fprintf(FID,'\n');
end

fclose(FID);
```

ACKNOWLEDGEMENTS

The authors wish to thank and acknowledge the US Department of Transportation, University Transportation Center Program (RailTEAM UTC) for funding support for this research. Additionally, the authors wish to acknowledge CSX Transportation for providing data for this research.

ABOUT THE AUTHOR

Robert A. Schill, Jr., Ph.D.,

Dr. Robert A. Schill, Jr.,Ph.D., Electrical Engineering, from University of Wisconsin Madison in Aug. 1986. Majored in Plasma and Controlled Fusion with Electromagnetics as a secondary area and Math/Physics as a minor area. M.S., Electrical Engineering, from University of Wisconsin Madison in May 1981; B.S., Electrical Engineering from Milwaukee School of Engineering in 1979. Senior Member, Institute of Electrical and Electronics Engineers (IEEE). Was an Assistant Professor at the University of Chicago Illinois from 1986 to 1993. Joined the faculty at the University of Nevada, Las Vegas in 1993. Currently, Full Professor at the University of Nevada, Las Vegas. Director & founder of the Energy-Materials Interaction Technology Initiative of Nevada (EMITION) Center for pulsed power and pulsed beam studies.