Chapter 2. Background Theory

2.1. Audio distortion effect: nonlinear signal processing

In audio signal processing, distortion is a term that falls under the concept of saturation. It refers to the effect that distorts the sound, as if crumpling it, by adding harmonics to the original signal that were not originally present. For example, a square wave is a distorted sine wave signal composed of the sine waves of the fundamental frequency and its odd harmonics. When viewed from the perspective of wave shaping, which alters the waveform of the signal, such acoustic effects are referred to as nonlinear signal processing. This indicates that within a system, the changes in output is not proportional to the changes in input (Reiss and McPherson, 2014). Figure 2.1 illustrates various wave-shaping curves, where the x-axis represents the input, and the y-axis represents the output.

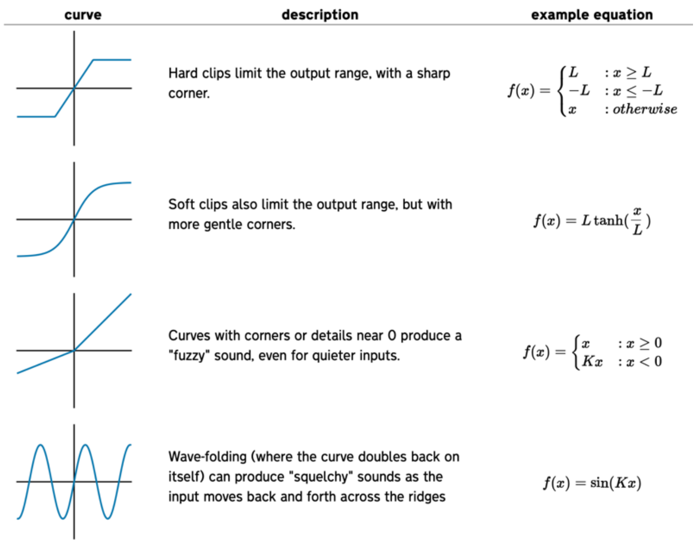

Figure 2.1 – Waveshaping curves (Luff, 2022)

When an audio signal undergoes such nonlinear signal processing, it retains its original pitch while acquiring a new timbre. In analogue systems, this nonlinear distortion/saturation effect is produced by op-amps, valves, and diodes, etc. The resulting warm or aggressive sound has long been cherished by musicians. On the other hand, digital systems are fundamentally linear (Chowdhury, 2019). However, using the wave shaping functions from Figure 2.1 or the VA Modelling method, it is possible to replicate the distortion effects generated by electronic circuit or to create digital signal systems with nonlinearity that does not originate from analogue world.

2.2. Nodal Voltage Analysis (NVA) and circuit discretization

Nodal Voltage Analysis (NVA) is one of the methods used for circuit analysis. It involves setting the voltage values at each node relative to a reference voltage as unknowns, and then using Kirchhoff's Current Law (KCL) or Kirchhoff's Voltage Law (KVL) to formulate equations for these node unknowns ("Electronics Tutorials", n.d.). Solutions are then found as needed to understand the electrical behaviour of the circuit. For example, through NVA, it is possible to calculate the current flowing through specific components within a circuit or to mathematically represent the electrical behaviour of the entire circuit.

The circuit discretization using the NVA method is a key procedure in conducting this VA Modelling project. It assumes the conversion of continuous signals in analogue circuits, which continuously flow over time, into a collection of discrete signals according to the sample rate. The purpose of circuit discretization is to obtain an equation called a transfer function, which indicates what output signal is derived from the input signal after going through the whole circuit.

2.3. Nodal Discrete Kirchhoff (DK) method and DK substitution for dynamic components

2.3.1. Dynamic characteristics of capacitors and necessity of DK substitution

To obtain a transfer function representing an electrical circuit's behaviour, formulas that can calculate the currents flowing through the components of the circuit are required. For instance, the relationship between current I and voltage V across a resistor R is defined according to Ohm's Law (Equation 2.1). The current flowing through the resistor, as calculated by Ohm's Law, remains constant regardless of the passage of time.

$$V = I * R, \qquad I = V/R \tag{2.1}$$

Conversely, the capacitor, another frequently encountered component in audio circuits, is dynamic. Unlike a resistor, which maintains a constant current, the voltage and current across a capacitor continually change over time as it goes through the processes of charging and discharging. For a capacitor with capacitance C, the relationship between the varying voltage

V_C and current I_C is given by Equation 2.2. The voltage across a capacitor is equal to the reciprocal of the capacitance multiplied by the integral of the current that has flowed through the capacitor over time. Inversely, the current flowing through the capacitor is equal to the capacitance multiplied by the rate of change of the voltage across the capacitor.

$$V_c = \frac{1}{C}\int I_c dt, \quad I_c = C\frac{dV_C}{dt} \tag{2.2}$$

When calculating the voltage and current on a capacitor based on a discrete signal rather than a continuous signal, one capacitor in the circuit to be discretized is replaced by a circuit where one resistor and one reverse current source are connected in parallel. This is known as the DK substitution. For example, the DK substitution circuit of Figure 2.2 is shown in Figure 2.3.

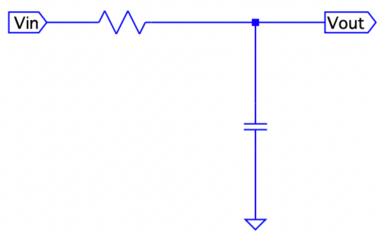

Figure 2.2 – A capacitor in a circuit

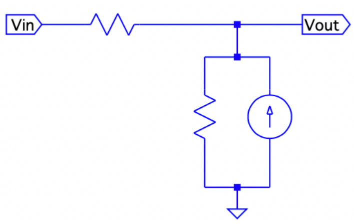

Figure 2.3 – DK substitution circuit of Figure 2.2

Since the substituted part of the circuit is connected in parallel (Figure 2.3), the voltage across this substituted part can be determined by the voltage across the resistor, and the current can be calculated by subtracting the current of the reverse current source from the current flowing through the resistor. Therefore, using DK substitution, it is possible to easily formulate the variable values of current and voltage flowing through a capacitor in response to the discrete input signals.

The theoretical basis and principles underlying the DK substitution for capacitors are detailed in the following subsections.

2.3.2. The principles behind the DK substitution for capacitors

In calculus, integration calculates the total change of a function up to a specific point, which is determined by the area under the curve for that segment. In contrast, differentiation represents the instantaneous rate of change of a function, determined by the slope at a specific point. These values can be approximated when the x-axis of the function's graph is discrete. For example, the integral up to point x_2 in the graph of function $f(x)$ in Figure 2.4 is the area under the red curve up to x_2. One method to approximate this area with minimal error is the trapezoidal rule, which uses the formula for calculating the area of a rectangle, i.e., multiplying the vertical length by the horizontal length, but with the vertical length taken as the average of two different vertical lengths of a trapezoid. The smaller the intervals between discrete signals, the smaller the error in the approximation of the integral value, allowing for a more accurate calculation of the area under the curve. In Equation 2.3, the trapezoidal rule is applied to calculate the integral of function $f(x)$ up to the discrete signal index x_2, where Ts represents the interval between one signal being extracted and the next, i.e., the distance between samples along the x-axis in Figure 2.4. Therefore, the formula for integrating function $f(x)$ using the trapezoidal rule can be generalized as Equation 2.4.

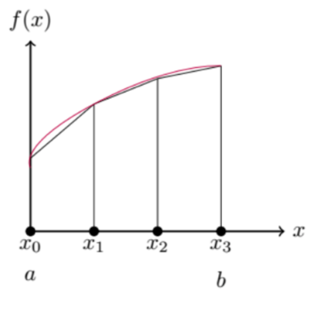

Figure 2.4 – Trapezoidal rule for calculating integral

$$\frac{f(x_0)+f(x_1)}{2}\cdot Ts+\frac{f(x_1)+f(x_2)}{2}\cdot Ts \tag{2.3}$$

$$\int f(x)dx \approx Ts \cdot \sum_{n=0}^{N-1} \frac{f(x_n) + f(x_{n+1})}{2} \tag{2.4}$$

Meanwhile, a discrete approximation of the derivative of the function *f(x)* at the point *x2*, is obtained by subtracting the integral value up to *x2*, from the integral up to *x3*. This calculation is shown in Equation 2.5.

$$\begin{aligned}
&\int f(x_3)dt - \int f(x_2)dt \\
&\qquad = \left(\frac{f(x_0) + f(x_1)}{2} \cdot Ts + \frac{f(x_1) + f(x_2)}{2} \cdot Ts + \frac{f(x_2) + f(x_3)}{2} \cdot Ts\right) \\
&\qquad - \left(\frac{f(x_0) + f(x_1)}{2} \cdot Ts + \frac{f(x_1) + f(x_2)}{2} \cdot Ts\right) \\
&\qquad = \left(\frac{f(x_2) + f(x_3)}{2}\right) \cdot Ts
\end{aligned} \tag{2.5}$$

By applying the interpretation of Equation 2.5 to Equation 2.6, which rearranges Equation 2.2 regarding current and voltage across a capacitor, the derivative of the voltage across the capacitor in a discrete system can be formulated as shown in Equation 2.7. The left side of Equation 2.7 refers to the instantaneous change in the voltage V across the capacitor in regard of discrete input signals.

$$\frac{dV_C}{dt} = I_C \cdot \frac{1}{C} \tag{2.6}$$

$$V[n] - V[n-1] = \frac{i[n] + i[n-1]}{2} \cdot T_s \cdot \frac{1}{C} \tag{2.7}$$

When discretizing a circuit, to set up an equation for a node using KCL, it is necessary to know the current flowing through the components. Therefore, to find the current *i[n]* flowing through a capacitor when the sample index is *n*, rearranging Equation 2.7 with respect to *i[n]* results in Equation 2.8.

$$i[n] = \frac{2C}{Ts} \cdot V[n] - \frac{2C}{Ts} \cdot V[n-1] - i[n-1] \tag{2.8}$$

In Equation 2.8, $\frac{2C}{Ts} \cdot V[n]$ is the value in regard of the current sample index *n*, and all subsequent parts are represented by the value in regard of the previous sample index *n-1*. Therefore, the values which are in regard of the previous sample *n-1* that affects *i[n]* for the current sample *n* is called the "state" and replaced by *X[n-1]* (Equation 2.9), then Equation 2.8 can be rewritten as Equation 2.10.

$$X[n-1] = \frac{2C}{Ts} \cdot V[n-1] + i[n-1] \tag{2.9}$$

$$i[n] = \frac{2C}{Ts} \cdot V[n] - X[n-1] \tag{2.10}$$

In Equation 2.10, the current *i[n]* flowing through a capacitor in regard of the sample index n, is represented in the form of subtracting the state value *X[n-1]* from $\frac{2C}{Ts} \cdot V[n]$ which is in regard of the current sample index *n*. In other words, in a discrete system, a capacitor can be substituted in a parallel circuit consisting of two current sources, one forward and one reverse, as shown in Figure 2.5.

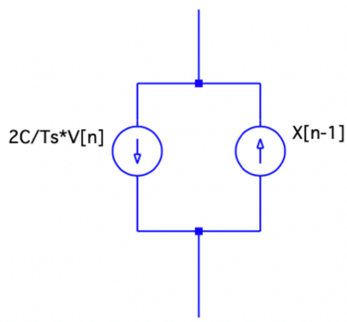

Figure 2.5 – Capacitor substitution in discrete system

According to Ohm's law (2.11.a), $\frac{2C}{Ts}$ in Equation 2.11.b, which is the value of the forward current source in Figure 2.5, can be considered as the reciprocal of the resistance (2.11.c). Therefore, when the forward current source is represented as a resistor, its resistance can be obtained by the period *Ts* and the capacitance *C* (2.11.d).

$$i = V \cdot \frac{1}{R} \quad (Ohm's\ Law) \tag{2.11.a}$$

$$i = V[n] \cdot \frac{2C}{Ts} \tag{2.11.b}$$

$$\frac{1}{R} = \frac{2C}{Ts} \tag{2.11.c}$$

$$R = \frac{Ts}{2C} \tag{2.11.d}$$

Therefore, the forward current source in Figure 2.5 can be expressed by replacing it with a resistor with a resistance of $\frac{Ts}{2C}\Omega$ as shown in Figure 2.6.

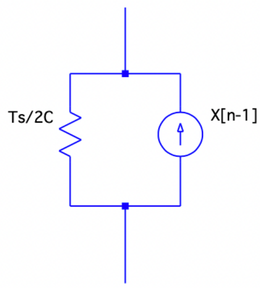

Figure 2.6 – DK substitution of a capacitor

Based on this principle, to express the varying value of the current flowing through the capacitor in a discrete system, one capacitor can be replaced by a parallel circuit consisting of one resistor and one reverse current source. This is called DK Substitution for the capacitor. In all the processes of circuit discretization performed for VA Modelling in Chapter 4 Methodology, the capacitors included in the reference circuit will be expressed as DK substitution.

2.3.3. State update equation for DK substitution circuit

In this subsection, the equation for obtaining the updated state value in the DK substitution circuit for capacitors is investigated as the index of the discrete input signal increases.

The equation representing a state *X[n-1]* is as shown in Equation 2.9. In a discrete system, when the index of the input signal increases from *n* to *n+1*, the state value *X[n-1]* is updated to *X[n]*, as expressed in Equation 2.12.a. According to Equation 2.11.c, Equation 2.12.a can be re-expressed as 2.12.b.

$$X[n] = \frac{2C}{Ts} \cdot V[n] + i[n] \tag{2.12.a}$$

$$X[n] = \frac{1}{R} \cdot V[n] + i[n] \tag{2.12.b}$$

The equation of *i[n]* on the right side of Equation 2.12.b was examined in Equation 2.10. By substituting $\frac{2C}{Ts}$ in Equation 2.10 for the reciprocal of resistance according to Equation 2.11.c, Equation 2.10 can be re-expressed as Equation 2.13.

$$i[n] = \frac{1}{R} \cdot V[n] - X[n-1] \tag{2.13}$$

Substituting *i[n]* of Equation 2.12.b into the right side of Equation 2.13, the equation for *X[n]* becomes as Equation 2.14.

$$\therefore \; X[n] = \frac{2}{R} \cdot V[n] - X[n-1] \tag{2.14}$$

This formula in Figure 2.14 is called state update equation. Whenever the discrete signal index is updated from *n* to *n+1*, the state value is updated according to this equation.

2.4. Diode and Newton-Raphson (N-R) method

2.4.1. Diodes and diode clipping circuit

An ideal diode allows current to flow infinitely when the input voltage is positive, functioning like a short circuit, and blocks current like an open circuit when the input voltage is negative, thereby allowing current to flow in only one direction. However, actual diodes, unlike the ideal assumptions, start conducting current only when a minimum voltage, known as the forward voltage, is applied. As shown in the graph in Figure 2.7, the forward voltage required to conduct current in a silicon diode is about 0.7V, and for a germanium diode, it is lower at 0.3V. Moreover, actual diodes do allow a small amount of reverse current to flow even when the input voltage is negative.

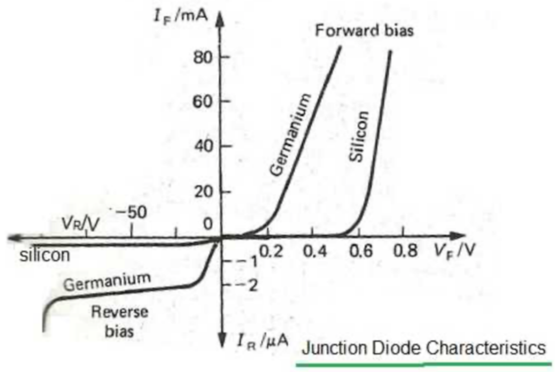

Figure 2.7 – I-V curves of actual diodes

These characteristics of diodes are used to be utilized in audio circuits, as shown in Figure 2.8 and Figure 2.9, in a back-to-back configuration of two (or more) diodes. This setup allows the current to flow when the bidirectional voltages of the signal reach the forward voltage of each diode, thereby limiting the voltage to the forward voltage and clipping the signal, which produces a distortion effect.

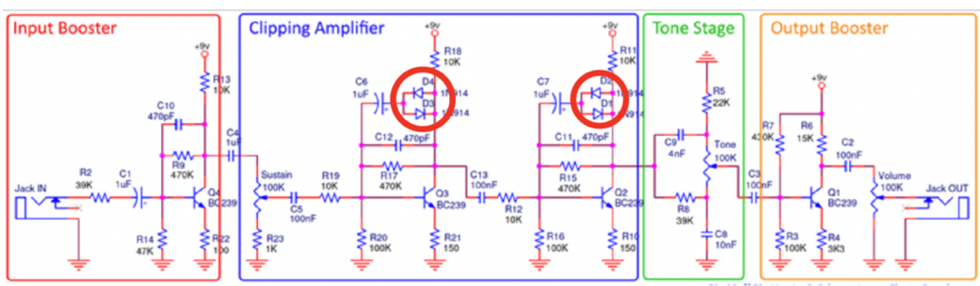

Figure 2.8 – Big Muff Pi ("Electrosmash", n.d.)

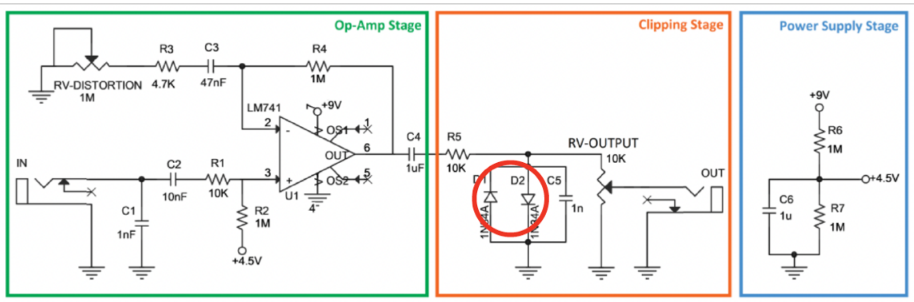

Figure 2.9 – MXR Distortion+ ("Electrosmash", n.d.)

2.4.2. Shockley diode equation: calculating the current flowing through the diode

The Shockley diode equation, named after William Shockley, co-inventor of the transistor, describes the formulas for calculating the current i_d flowing through a diode (Figure 2.15.a) and the current i_{d_pair} flowing through a pair of diodes in a back-to-back configuration (Figure 2.15.b).

$$i_d = I_s * [e^{(\frac{V_d}{V_T \cdot \eta})} - 1] \tag{2.15.a}$$

$$i_{d\_pair} = 2 \cdot I_s \cdot \sinh\left(\frac{V_d}{\eta \cdot V_T}\right) \tag{2.15.b}$$

The equations involve two parameters of diodes, I_s and V_T, and one constant, η (eta). I_s represents the saturation current of the diode, which is a very small reverse current generated by heat even when the practical diode is not conducting current. V_T is the thermal voltage of the diode, which represents the voltage across the diode under a temperature condition of approximately 27 degrees Celsius. η (eta) is the diode's emission coefficient, which indicates the current emission capability of the diode based on its material. For the germanium diodes used in the *MXR Distortion+* circuit, which is modeled in this project, the saturation current is approximately 10^{-6}A, the thermal voltage is about 26mV, and the emission coefficient ranges from 1.5 to 2.

In Equation 2.15.a and 2.15.b, V_d represents the voltage across the diode. When setting up equations for nodes using NVA and KCL in a circuit that includes a diode, if the diode voltage V_d is related to the output voltage of a circuit, it is necessary to solve the equation for V_d to derive the system's transfer function. However, in the equation derived from KCL, V_d would appear in the exponent of a natural constant *e* due to the Shockley diode equation (Figure 2.15.a, Figure 2.15.b), making it algebraically unsolvable for V_d. To tackle this, one approach is to rearrange all terms of the derived equation to one side, setting it as a function $f(V_d)$, and then seek the value of V_d where this function $f(V_d)$ equals zero using a numerical method. The Newton-Raphson method, which will be covered in the subsequent subsection 2.4.3, is specifically designed to rapidly find solutions for V_d.

2.4.3. Newton-Raphson (N-R) method and N-R update equation

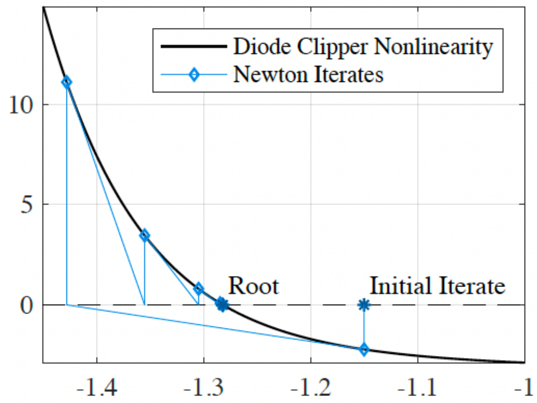

Figure 2.10 – Newton's method solving the diode clipper nonlinearity (Holmes, 2019)

In Figure 2.10, the x-axis represents the voltage V_d applied to the diode, and the y-axis depicts the function $f(V_d)$. The solution sought is the value of V_d at which $f(V_d)$ equals zero. The Newton-Raphson method sets an arbitrary initial guess for V_d, then calculates the derivative $f'(V_d)$, which is the slope of the function $f(V_d)$ at the point of the initial guess of V_d, and checks the value of $f(V_d)$ where this slope intersects the x-axis. If this value is not zero, the process is repeated to find the value of V_d that makes $f(V_d)$ equal to zero. According to Figure 2.10, after a total of just seven iterations, the value of V_d that makes $f(V_d)$ zero was found (Holmes, 2019). This method dramatically shortens the procedure it takes to find solutions of equations that are not solved algebraically.

If the initial guess for the value of V_d is called x_0, the slope $f'(x_0)$ which is the gradient of the function $f(x_0)$ matches the slope where $f(x_0)$ meets the x-axis. Equation 2.16.a describes $f'(x_0)$, where the numerator on the right side represents the difference between two coordinates, $f(x_0)$ and 0, based on the y-axis. The denominator represents the difference between two coordinates, x_0 and x_1, based on the x-axis, where x_1 denotes the next guess for V_d for the next Newton-Raphson iteration. Therefore, Equation 2.16.b, which is solved Equation 2.16.a for x_1, is the N-R update equation that obtains the updated guess of V_d for the next N-R iteration.

$$f'(x_0) = \frac{f(x_0)}{x_0 - x_1} \tag{2.16.a}$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{2.16.b}$$

Chapter 4. Methodology

4.1. Research Process Flowchart

Figure 4.1 is a flowchart that outlines the entire process of the methodology of this project. The "Op-amp stage" and "Clipping stage" represented in the connector blocks of the flowchart refer to the sub-circuits in the *MXR Distortion+ (D+)* as shown in Figure 4.2, while the Power Supply Stage is considered unnecessary for the modelling into the digital system in this project and is therefore not included in the replication.

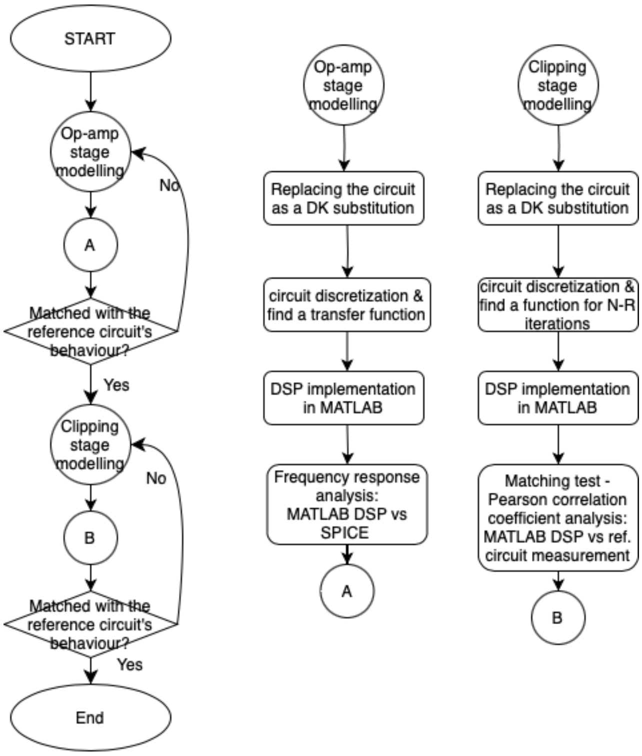

Figure 4.1 – Research process flowchart

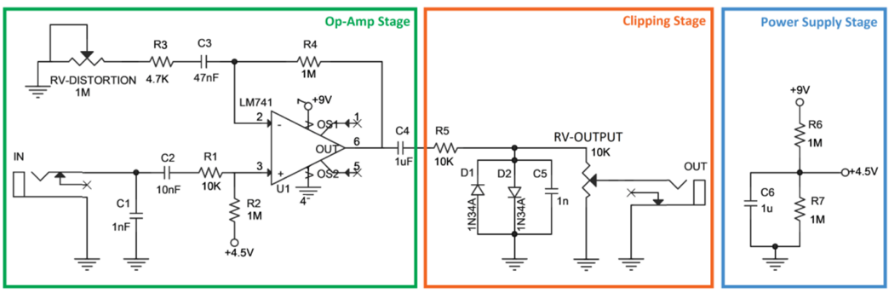

Figure 4.2 – MXR Distortion+ ("Electrosmash", n.d.)

4.2. Op-amp stage modelling

In this section, the op-amp stage of the *D+* is modelled as a digital signal processor. To this end, the transfer function representing the relationship between the input and output signals of the circuit is derived, and based on this equation, code is written in MATLAB to create a system that behaves identically to the reference circuit's electrical behaviour. Then, by comparing the frequency response graph of the replicated digital model with that of the original, the accuracy of circuit replication is evaluated.

The concepts and theories applied in the circuit discretization process, such as Nodal Voltage Analysis (NVA) and Discrete Kirchhoff (DK) substitution, are thoroughly detailed in Section 2.2 and 2.3 of Chapter 2.

4.2.1. DK substitution of the op-amp stage

Figure 4.4 is a circuit that has undergone DK Substitution of capacitors from the *D+*'s op-amp stage in Figure 4.3. In this process, components that do not affect the principal signal processing in the reference circuit (Figure 4.3) are either removed or adjusted. First, C1, which is placed at the input-end of the circuit to protect the signal's quality and match impedance when other units are connected in front of this circuit, does not influence the main signal processing of the reference circuit, and is therefore removed. Second, the 4.5V connected to R2 is a setting for DC offset in consideration of the power supply, which is unnecessary in a digital system, thus, it is replaced with a normal reference voltage, i.e., ground. Finally, C4, a coupling capacitor that blocks DC, also does not impact the main signal processing of the reference circuit and is therefore excluded.

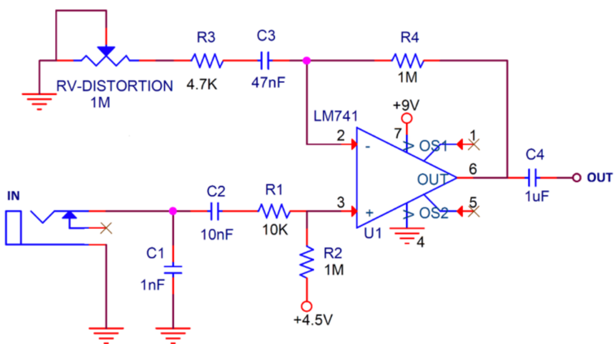

Figure 4.3 – Op-amp stage of D+ (“Electrosmash”, n.d.)

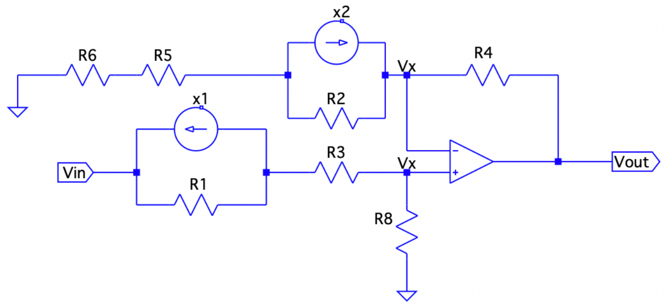

Figure 4.4 – DK substitution circuit of Figure 4.3

In Figure 4.4, *Vin* and *Vout* represent the voltages of the input and output signals, respectively, and both will be interpreted as the amplitude for discrete signals. Furthermore, assigning the same unknown voltage *Vx* to the op-amp's inverting/non-inverting input-end's nodes is based on the virtual short-circuit theory of op-amps, which suggests that the voltage difference between the two input terminals of the op-amp is close to zero.

Component numbering has been newly assigned in Figure 4.4, and the calculations for the circuit discretization performed in the following subsection are based on this new numbering.

4.2.2. Circuit discretization

The purpose of circuit discretization is to derive equations that can determine the voltage of the output signal based on the voltage of the input signal when the signal to the system is not continuous but discrete. These equations include the transfer function of the system, and the state-update equation for *x1* and *x2* in the system (Figure 4.4). The detailed process of circuit discretization to obtain these two equations is as follows.

Process 1: In the input section, obtain the equation for *Vx* expressed only with the component values and *Vin*, using KCL.

Process 2: In the distortion potentiometer section, find the equation for the current flowing at any point of this section.

Process 3: In the feedback loop section, obtain the equation for *Vx* expressed only with the component values and *Vout*, using KCL.

Process 4: The two equations for *Vx* obtained in *Process 1* and *Process 3* are formed as a simultaneous equation. *Vx* is eliminated, and the transfer function of the system is obtained by solving the remained equation for *Vout*.

Process 5:	Set the state-update equation

In the following subsections, the above processes are performed in order.

4.2.2.1.Input section (a), (b)

Process 1:	In the input section, obtain the equation for *Vx* expressed only with the component values and *Vin*, using KCL.

In Figure 4.4, centered around the node named *Vx* at the non-inverting input of the op-amp, two branches of the circuit extend: one branch from *Vin* to *Vx*, and the other from *Vx* to ground. These two stages are respectively named input section (a) and (b). Based on Kirchhoff's Current Law (KCL), which states that the current entering and leaving node *Vx* must be equal, an equation for *Vx* is derived.

Input section (a):

First, in Figure 4.5, according to KVL, the input voltage *Vin* can be represented by Equation 4.1. At this point, the voltage drops in the subcircuit, which is depicted as a parallel connection of *R1* and *x1*, can be calculated as V_{R1} which is the voltage across *R1*, due to the circuit being in parallel.

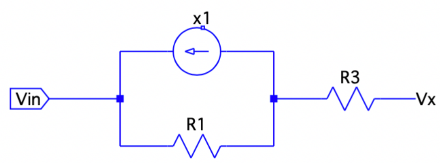

Figure 4.5 – Input section (a)

$$V_{in} = V_{R1} + V_{R3} + V_x \tag{4.1}$$

Solving Equation 4.1 for V_x results in Equation 4.2.

$$V_x = V_{in} - V_{R1} - V_{R3} \tag{4.2}$$

In Figure 4.5, the parallel sub-circuit consisting of resistor *R1* and a reverse current source *x1* represents the DK Substitution circuit for the original circuit's capacitor, renamed as *C1*, which was originally *C2* in Figure 4.3. The current flowing through this capacitor, i_{C1}, is calculated by subtracting the value of the reverse current source *x1* from the value of the current flowing through *R1* according to Ohm's law. This is expressed in Equation 4.3.

$$i_{C1} = \frac{V_{R1}}{R1} - x1 \tag{4.3}$$

The current flowing through *R3* is expressed as Equation 4.4 according to Ohm's Law.

$$i_{R3} = \frac{V_{R3}}{R3} \tag{4.4}$$

Because the parallel circuit consisting of *R1* and *x1* represents a DK substitution circuit for the single component C1, Figure 4.5 essentially shows a circuit with one capacitor and one resistor connected in series. Therefore, since the values of i_{C1} and i_{R3} are the same (Equation 4.5), the right-hand sides of Equations 4.3 and 4.4 can be shown to be equal (Equation 4.6).

$$i_{C1} = i_{R3} \tag{4.5}$$

$$\therefore \frac{V_{R1}}{R1} - x1 = \frac{V_{R3}}{R3} \tag{4.6}$$

Solving Equation 4.6 for V_{R3} results in Equation 4.7.

$$V_{R3} = \frac{R3}{R1} \cdot V_{R1} - (R3 \cdot x1) \tag{4.7}$$

When the value of V_{R3} obtained from Equation 4.7 is substituted into Equation 4.2, it results in Equation 4.8.

$$V_x = V_{in} - V_{R1} - \frac{R3}{R1} \cdot V_{R1} + (R3 \cdot x1) \tag{4.8}$$

To solve Equation 4.8 for V_{R1}, calculations are made (Equation 4.9).

$$\left(1+\frac{R3}{R1}\right)V_{R1} \ = \ V_{in} \ - \ V_x \ + \ (R3 \cdot x1) \tag{4.9}$$

For ease of calculation, substituting $\left(1+\frac{R3}{R1}\right)$ in Equation 4.9 with G_a results in Equation 4.10.

$$G_a \cdot V_{R1} \ = \ V_{in} \ - \ V_x \ + \ (R3 \cdot x1) \tag{4.10}$$

Therefore, solving Equation 4.10 for V_{R1}, the previously unknown value of V_{R1} is transformed into an equation comprised only of determinable elements, excluding *Vx*, as shown in Equation 4.11.

$$\therefore V_{R1} \ = \ \frac{V_{in} - V_x + (R3 \cdot x1)}{G_a} \tag{4.11}$$

When the equation for V_{R1} obtained from Equation 4.11 is substituted into Equation 4.3, which represents the equation for i_{C1}, it results in Equation 4.12.

$$i_{C1} \ = \ \frac{V_{R1}}{R1} \ - \ x1 \ = \ \frac{V_{in} - V_x + (R3 \cdot x1)}{G_a \cdot R1} \ - \ x1 \tag{4.12}$$

Equation 4.5 demonstrates that i_{C1} and i_{R3} are equal, and it has been established that the DK substitution circuit in Figure 4.5 essentially comprises a single capacitor and a resistor connected in series. This indicates that the current at any point in the circuit of Figure 4.5 is consistent. Thus, the current flowing uniformly through this circuit and entering node *Vx* is denoted as i_a in Equation 4.13.

$$\therefore i_a \ = \ \frac{V_{in} - V_x + (R3 \cdot x1)}{G_a \cdot R1} \ - \ x1 \tag{4.13}$$

Equation 4.13 consists only of known component values and *Vin*, as well as *Vx*, which will later be eliminated in the system of equations, rather than unknown values such as V_{R1} or V_{R3}.

Input section (b):

In the input section (b) of Figure 4.6, the current flowing through *R8*, i_{R8}, is determined by Ohm's Law as Equation 4.14.

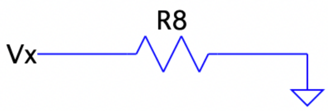

Figure 4.6 – Input section (b)

$$i_{R8} = \frac{V_x}{R8} \tag{4.14}$$

The current i_{R8}, calculated in Equation 4.14, flows at a consistent value throughout the circuit in Figure 4.6, and is equal to the current flowing out from the node *Vx*; thus, it can be denoted as i_b.

According to KCL, the current entering node *Vx*, calculated in Equation 4.13 as i_a, and the current exiting node *Vx*, calculated in Equation 4.14 as i_b, are equal (Equation 4.15); therefore, it can be expressed as Equation 4.16.

$$i_a = i_b \tag{4.15}$$

$$\therefore \ \frac{V_x}{R8} = \frac{V_{in} - V_x + (R3 \cdot x1)}{G_a \cdot R1} - x1 \tag{4.16}$$

Now, Equation 4.16 will be solved with respect to *Vx* (Equation 4.17).

$$\left(\frac{1}{R8} + \frac{1}{Ga \cdot R1}\right) \cdot V_x = \frac{1}{G_a \cdot R1} \cdot V_{in} - \left(1 - \frac{R3}{G_a \cdot R1}\right) \cdot x1 \tag{4.17}$$

For simplicity of the equation, substitute $\left(\frac{1}{R8}+\frac{1}{Ga \cdot R1}\right)$ with G_x, thus re-expressing Equation 4.17 as Equation 4.18.

$$G_x \cdot V_x = \frac{1}{G_a \cdot R1} \cdot V_{in} - \left(1 - \frac{R3}{G_a \cdot R1}\right)\left(\frac{x1}{G_x}\right) \tag{4.18}$$

When the above equation is finally solved for Vx, it results in Equation 4.19.

$$V_x = \frac{1}{G_a \cdot R1 \cdot G_x} \cdot V_{in} - \left(1 - \frac{R3}{G_a \cdot R1}\right)\left(\frac{x1}{G_x}\right) \tag{4.19}$$

Through *Process 1*, in the input section of the op-amp stage, Equation 4.19 has been derived for node *Vx* at the non-inverting input of the op-amp, expressed solely in terms of component values, state values, and *Vin*.

4.2.2.2. Distortion potentiometer section

Process 2: In the distortion potentiometer section, find the equation for the current flowing at any point of this section.

Figure 4.7 illustrates a subcircuit, which starts from the inverting input of the op-amp in the overall circuit (Figure 4.4) and connects to the ground. The resistor *R6* is a reverse-log variable resistor that varies from 0 to 1 megaohm, hence this subcircuit is referred to as the distortion potentiometer section. In this context, the resistances *R5* and the variable resistor *R6* are combined as shown in Equation 4.20 and represented as *Rn*, with the circuit reappearing as Figure 4.8.

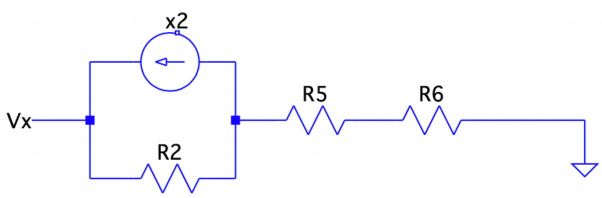

Figure 4.7 – Distortion potentiometer section with R5, R6

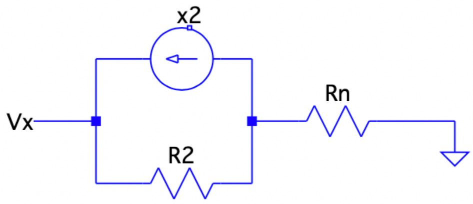

Figure 4.8 – Distortion potentiometer section with Rn

$$R5 + R6 = Rn \tag{4.20}$$

First, the circuit in Figure 4.8 is represented as Equation 4.21.a according to KVL. V_{R2} and V_{Rn} denote the voltages across *R2* and *Rn*, respectively. Solving for *Vx* yields Equation 4.21.b.

$$V_x - V_{R2} - V_{Rn} = 0 \tag{4.21.a}$$

$$\therefore \quad V_x = V_{R2} + V_{Rn} \tag{4.21.b}$$

The current i_{Rn} flowing through *Rn* is as in Equation 4.22 according to Ohm's Law

$$i_{Rn} = \frac{V_{Rn}}{Rn} \tag{4.22}$$

The current i_{C2} flowing through capacitor *C2*, represented in the DK Substitution circuit in Figure 4.8, is determined by first expressing the current flowing through *R2* according to Ohm's Law, and then subtracting the reverse current source *x2*. Therefore, it is represented as Equation 4.23.

$$i_{C2} = \frac{V_{R2}}{R2} - x2 \tag{4.23}$$

The parallel circuit composed of *R2* and *x2* acts as the DK substitution circuit for the single component *C2*. Therefore, the distortion potentiometer section's circuit (Figure 4.8) essentially resembles a circuit with one capacitor and one resistor connected in series. Consequently, the current flowing through *Rn* and *C2*, i_{Rn} and i_{C2}, are equal as shown in Equation 4.24.a, which allows it to be represented as Equation 4.24.b.

$$i_{Rn} = i_{C2} \tag{4.24.a}$$

$$\therefore \ \frac{V_{Rn}}{Rn} = \frac{V_{R2}}{R2} - x2 \tag{4.24.b}$$

Solving Equation 4.24.b for V_{R2} yields Equation 4.25.

$$V_{R2} = \frac{V_{Rn} \cdot R2}{Rn} + R2 \cdot x2 \tag{4.25}$$

Substituting V_{R2}, derived from Equation 4.25, into Equation 4.21.b for V_x, results in Equation 4.26.

$$V_x = \frac{V_{Rn} \cdot R2}{Rn} + R2 \cdot x2 + V_{Rn} \tag{4.26}$$

The above Equation 4.26 is then solved for V_{Rn}. In Equation 4.27.a, for simplicity of expression, substituting $\left(\frac{R2}{Rn} + 1\right)$ with G_b results in Equation 4.27.b. Solving Equation 4.27.b for V_{Rn} yields Equation 4.27.c.

$$V_x = V_{Rn}\left(\frac{R2}{Rn} + 1\right) + R2 \cdot x2 \tag{4.27.a}$$

$$G_b \cdot V_{Rn} = V_x - R2 \cdot x2 \tag{4.27.b}$$

$$V_{Rn} = \frac{V_x - R2 \cdot x2}{G_b} \tag{4.27.c}$$

Substituting the value of V_{Rn} derived from Equation 4.27.c into Equation 4.22 for i_{Rn} reconfigures it into Equation 4.28.

$$i_{Rn} = \frac{V_x - R2 \cdot x2}{G_b \cdot Rn} \tag{4.28}$$

Before the DK substitution, the circuit in this section originally consisted of a single capacitor, one resistor, and one variable resistor connected in series. Therefore, the i_{Rn} calculated in Equation 4.28 is equivalent to the current flowing at any point in this circuit. Consequently, this value can also be seen as the current leaving the node V_x, and is designated as i_c (Equation 4.29).

$$i_c = \frac{V_x - R2 \cdot x2}{G_b \cdot Rn} \tag{4.29}$$

Through *Process 2*, the current i_c exiting the node V_x in the distortion potentiometer section was calculated. This value is used along with the current i_d, which is calculated in the following subsection and enters the node V_x, to derive an expression for V_x according to KCL.

4.2.2.3. Feedback loop section

Process 3: In the feedback loop section, obtain the equation for *Vx* expressed only with the component values and *Vout*, using KCL.

Figure 4.9 illustrates the feedback loop section connected to the inverting input of the op-amp in the overall circuit (Figure 4.3), which is referred to as the feedback loop section.

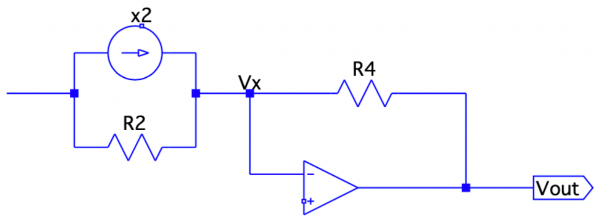

Figure 4.9 – Feedback loop section

In Figure 4.9, the current flowing into V_x is denoted as i_d. Together with i_c, the current exiting V_x as calculated in Equation 4.29, KCL establishes Equation 4.30.

$$i_d = i_c \tag{4.30}$$

The feedback loop section is connected solely to *R4*, and the current flowing through *R4* is equal to the current entering node V_x, denoted as i_d . According to Ohm's Law, i_d is represented as in Figure 4.31. The voltage drops across *R4*, using NVA, is expressed as $V_{out} - V_x$.

$$i_d = \frac{V_{out} - V_x}{R4} \tag{4.31}$$

Therefore, i_c, derived from Equation 4.29, and i_d, derived from Equation 4.31, are equal according to KCL, as expressed in Equation 4.32.a. Solving for V_x, and for simplicity in the subsequent formulation, substituting $\left(1 + \frac{R4}{G_b \cdot Rn}\right)$ with G_h in Equation 4.32.b, the final expression for V_x is presented as Equation 4.32.c.

$$\therefore \quad \frac{V_{out} - V_x}{R4} = \frac{V_x - (R2 \cdot x2)}{G_b \cdot Rn} \tag{4.32.a}$$

$$\left(1 + \frac{R4}{G_b \cdot Rn}\right) \cdot V_x = V_{out} + \frac{R2 \cdot R4}{G_b \cdot Rn} \cdot x2 \tag{4.32.b}$$

$$V_x = \frac{1}{G_h} \cdot V_{out} + \frac{R2 \cdot R4}{G_b \cdot Rn \cdot G_h} \cdot x2 \qquad (4.32.c)$$

As a result, through *Process 3*, the equation for V_x at the inverting input of the op-amp was derived, as shown in Equation 4.32.c.

4.2.2.4. Transfer function

Process 4: The two equations for *Vx* obtained in *Process 1* and *Process 3* are formed as a simultaneous equation. *Vx* is eliminated, and the transfer function of the system is obtained by solving the remained equation for *Vout*.

Equations 4.33 and 4.34 represent the expressions for V_x derived respectively from *Processes 1* and *Process 3*. Since both equations have V_x on their left sides, it can be assumed that their right sides are also equal, allowing for the elimination of V_x and the establishment of Equation 4.35.a. To determine the system's transfer function, solving this Equation 4.35.a for V_{out} results in Equation 4.35.b.

$$V_x = \frac{1}{G_a \cdot R1 \cdot G_x} \cdot V_{in} - (1 - \frac{R3}{G_a \cdot R1})(\frac{x1}{G_x}) \qquad (4.33)$$

$$V_x = \frac{1}{G_h} \cdot V_{out} + \frac{R2 \cdot R4}{G_b \cdot Rn \cdot G_h} \cdot x2 \qquad (4.34)$$

$$\frac{1}{G_a \cdot R1 \cdot G_x} \cdot V_{in} - \left(1 - \frac{R3}{G_a \cdot R1}\right)\left(\frac{x1}{G_x}\right) = \frac{1}{G_h} \cdot V_{out} + \frac{R2 \cdot R4}{G_b \cdot Rn \cdot G_h} \cdot x2 \qquad (4.35.a)$$

$$V_{out} = \frac{G_h}{G_a \cdot R1 \cdot G_x} \cdot V_{in} + \left(\frac{R3}{G_a \cdot R1} - 1\right)\left(\frac{G_h}{G_x}\right) \cdot x1 + (\frac{-R2 \cdot R4}{G_b \cdot Rn}) \cdot x2 \qquad (4.35.b)$$

Equation 4.35.b represents the transfer function that mathematically describes the relationship between the discrete input voltage V_{in} and the corresponding discrete output voltage V_{out} at the op-amp stage of the MXR Distortion+. Additionally, the coefficients of this transfer function will later be entered in MATLAB code and are named as follows: the coefficient for V_{in} as b_0 in Equation 4.36.a, the coefficient for *x1* as b_1 in Equation 4.36.b, and the coefficient for *x2* as b_2 in Equation 4.36.c.

$$b_0 = \frac{G_h}{G_a \cdot R1 \cdot G_x} \tag{4.36.a}$$

$$b_1 = \left(\frac{R3}{G_a \cdot R1} - 1\right)\left(\frac{G_h}{G_x}\right) \tag{4.36.b}$$

$$b_2 = \frac{-R2 \cdot R4}{G_b \cdot Rn} \tag{4.36.c}$$

Additionally, Equation 4.37 contains the *G* values substituted for simplicity during the calculation from *Process 1* to *Process 3*. These values will also be entered into MATLAB code later.

$$Ga = 1 + \frac{R3}{R1}, \quad Gb = \frac{R2}{Rn} + 1, \quad Gx = \frac{1}{R8} + \frac{1}{Ga \cdot R1}, \quad Gh = 1 + \frac{R4}{G_b \cdot Rn} \tag{4.37}$$

4.2.2.5. State-update equations

Process 5: Set the state-update equations

This process pertains to the reverse current sources represented states in the DK substitution for capacitors. Upon the completion of signal processing a given input discrete signal and the subsequent arrival of the next discrete signal, the purpose of Process 5 is to formulate an equation for updating the state values associated with the previous discrete signal. This update occurs every time as the index number of the discrete signals entering the system increases.

Initially, when the first discrete signal is input, the previous signal does not exist, so the initial state value is 0. Equations 4.38.a and 4.38.b indicate that the state values in the DK substitution circuit for *C1* and *C2* are 0 when the index of the discrete input signal *n* is 1, that is, when the very first sample of the discrete signal is entered into the system.

$$x_1[n] = 0, \qquad n = 1 \tag{4.38.a}$$

$$x_2[n] = 0, \qquad n = 1 \tag{4.38.b}$$

After the signal processing of the first signal is completed and the next discrete signal is introduced, the index *n* is updated to *n+1*, and the state value is then represented as *x[n+1]*. According to the theory outlined in Section 2.3 of Chapter 2 and Equation 2.14, the state values for *C1* and *C2* in the DK substitution circuit when the *n+1*th discrete signal is introduced are as stated in Equations 4.39.a and 4.39.b respectively.

$$x_1[n+1] = \frac{2}{R1} \cdot V_{R1} - x_1[n] \tag{4.39.a}$$

$$x_2[n+1] = \frac{2}{R2} \cdot V_{R2} - x_2[n] \tag{4.39.b}$$

The V_{R1} and V_{R2} included in Equations 4.39.a and 4.39.b are derived from *Processes 1* and *Process 2*, respectively, as expressed in Equations 4.40.b and 4.40.d. Additionally, the expressions for V_x and V_{Rn} included in Equation 4.40.b and Equation 4.40.d are as determined in *Processes 2* and *Process 3*, corresponding to Equations 4.40.a and 4.40.c, respectively. These expressions must be entered before declaring the state-update equations in the MATLAB script based on the discretization of the circuit.

$$V_x = \frac{1}{G_h} \cdot V_{out} + \frac{R2 \cdot R4}{G_b \cdot Rn \cdot G_h} \cdot x_2 \tag{4.40.a}$$

$$V_{R1} = \frac{V_{in} - V_x + (R3 \cdot x_1)}{G_a} \tag{4.40.b}$$

$$V_{Rn} = \frac{V_x - (R2 \cdot x_2)}{G_b} \tag{4.40.c}$$

$$V_{R2} = \frac{R2}{Rn} \cdot V_{Rn} + (R2 \cdot x_2) \tag{4.40.d}$$

Thus, through *Process 1* to *Process 5*, the circuit discretization process of the op-amp stage in the MXR Distortion+ circuit has been successfully completed, deriving the transfer function and its coefficients that determine the output discrete signal from the input discrete signal, as well as the state-update equations for DK substitution circuits.

4.2.3. DSP implementation in MATLAB

In this subsection, based on the transfer function, coefficients, and state update equation derived from the circuit discretization in the previous section, the objective is to prototype a discrete signal processor using MATLAB that replicates the electrical behaviour of the reference circuit in a digital system. The complete code is available in the Appendix.

4.2.3.1. Flowchart

Figure 4.10 is a flowchart for the MATLAB code that digitally replicates the electrical behavior of the MXR Distortion+ op-amp stage, based on the results of the circuit discretization discussed in the previous subsection. The details of the code are analyzed in the subsequent subsection.

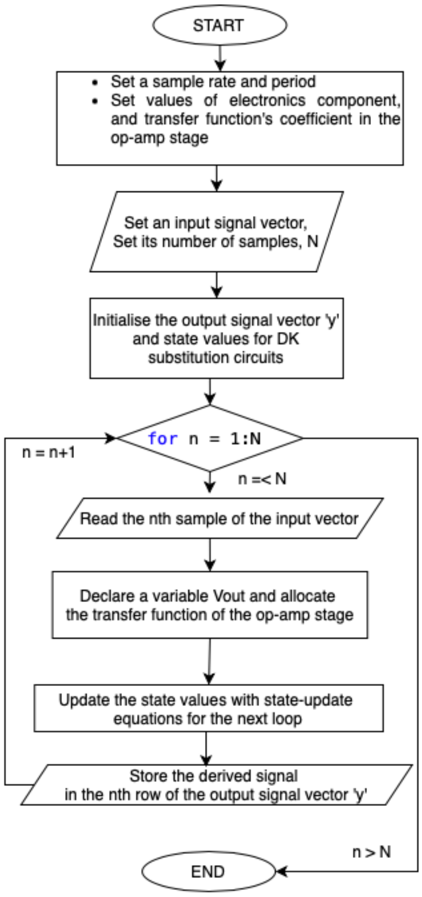

Figure 4.10 – Flowchart of MATLAB codes for the op-amp stage

4.2.3.2. Code Analysis

```
clear; clc;
% This script is emulating the signal processing of
% MXR Distortion+ guitar pedal in digital


%sample rate and period
Fs = 44100;
Ts = 1/Fs;

% Op-amp stage

% component values of the DK substitution circuit
C1 = 10e-9;
R1 = Ts/(2*C1);

C2 = 47e-9;
R2 = Ts/(2*C2);

R3 = 10e3;
R4 = 1e6;
R5 = 4.7e3;
R8 = 1e6;

```

Figure 4.11 – MATLAB code for the op-amp stage (1)

Figure 4.11 depicts the initial part of the code that digitally replicates the electrical behaviour of the op-amp stage circuit. It declares the variables representing the sample rate, period, and the values of components from the reference circuit (Figure 4.3). Line 7 sets the sample rate of the system to 44,100, meaning it samples 44,100 discrete signals per second. Line 8 represents the period, which is the reciprocal of the sample rate, indicating the time it takes from extracting one sample to the next. Lines 13 through 22 specify the values of components that constitute the op-amp stage circuit. Here, *R1* and *R2* are components in the DK substitution circuit for *C1* and *C2*, and according to the theory identified in the sub-section 2.3.2, the values of *R1* and *R2* are $\frac{Ts}{2 \cdot C1}$ and $\frac{Ts}{2 \cdot C2}$, respectively. Hence, the values of *C1* and *C2* are declared first in lines 13 and 16, before *R1* and *R2* are declared.

```
24  % "Distortion" RV-POT position from 0 to 1
25  % 1-Mega Ohm reverse-log pot
26  pot = 0.75; % from 0 to 1
27  k = 8; % sensitivity factor of the reverse-log pot
28  R6 = (exp(-k * pot) - exp(-k)) / (1 - exp(-k)) * 1000000;
29  Rn = R5 + R6; % from 4.7k + 0  to  4.7k + 1Meg
30
31  % transfer function's G-values
32  Ga = 1 + (R3/R1);
33  Gb = 1 + (R2/Rn);
34  Gh = 1 + (R4/(Gb*Rn));
35  Gx = (1/R8) + (1/(Ga*R1));
36
37  % transfer function's coefficients
38  b0 = Gh/(Ga*R1*Gx);
39  b1 = ((R3/(Ga*R1))-1)*(Gh/Gx);
40  b2 = (-R2*R4)/(Gb*Rn);
41
42
```

Figure 4.12 – MATLAB code for the op-amp stage (2)

In the code illustrated in Figure 4.12, the equation for the distortion potentiometer are identified, and the variables are declared to represent the coefficients of the transfer function and the *G*-values determined from the process of circuit discretization.

According to Electrosmash's documentation on the *D+* circuit, the op-amp stage's distortion potentiometer utilizes a 10 megaohm reverse-log variable resistor ("Electrosmash.com", n.d). Known as anti-logarithmic pots, reverse log pots do not produce outputs proportionate to the angular control setting of the variable resistor (Self, 2020). For example, turning the knob of this reverse log potentiometer to its halfway point does not halve the resistance value. The variable '*pot'* in line 26 represents the position of the variable resistor as a number between 0 and 1, where 0.5 simulates the knob turned to its halfway position. Line 28 contains the formula for calculating the resistance of *R6* based on the knob position '*pot'*, which on a graph would display an exponentially decreasing curve. With '*pot*' at 0, *R6* reaches its maximum of 1 megaohm, and at '*pot*' at 1, the minimum value of 0, decreasing sharply as '*pot*' increases from 0 to 1. The constant `k` in line 27 determines the steepness of the exponential decrease in *R6* values in response to changes in '*pot*'. The sensitivity value of 8 was determined based on a Pearson correlation coefficient measure with actual audio samples from the pedal since no specific data on the sensitivity of the variable resistor used in this circuit could be found. Ideally, however, the value for '*k*' should be derived from separate tests on the actual variable resistor used in the pedal.

Lines 32 to 35 specify the *G*-values that were substituted for simplification of the formulas in the circuit discretization process, while lines 38 to 40 designate the coefficients of the transfer function derived from Equation 4.35.b.

```
43      % input signal: audio file
44      [input, Fs] = audioread('testaudio.wav');
45      N = length(input);
46
47      % output signal setting
48      y = zeros(N,1);
49
50      % Initial state value for DK substitution circuit of capacitors
51      x1 = 0; x2 = 0;
52
```

Figure 4.13 – MATLAB code for the op-amp stage (3)

Figure 4.13 illustrates the sections of the code where the input/output signal vectors and initial state values are specified. Line 44 indicates the loading of an audio file as the input signal for this digital system. Line 45 uses the '*length*' function to determine the number of samples, i.e., the sample index values, of the loaded audio file. According to the MATLAB Workspace, the total number of samples in the audio file is 1,235,355. Line 48 declares a vector, denoted as '*y*', which will receive the discrete signal output after the input vector has passed through the system; this vector is initialized with all sample values set to zero and is of the same size, *N*, as the input vector. Line 51 sets the initial value of the state representing the reverse current sources in the DK substitution circuit for *C1* and *C2* to zero.

```
53  % opamp-stage sample-by-sample processing
54  for n = 1:N
55      Vin = input(n,1); % discrete input signal
56      Vout = b0 * Vin + b1 * x1 + b2 * x2;
57  
58      % some values included in state-update equations
59      Vx = (1/Gh)*Vout + ((R2*R4)/(Gb*Gh*Rn))*x2;
60      VR1 = (Vin-Vx+(R3*x1))/Ga;
61  
62      VRn = (Vx-(R2*x2))/Gb;
63      VR2 = (R2/Rn)*VRn + (R2*x2);
64  
65      % state-update equations
66      x1 = (2/R1)*VR1 - x1;
67      x2 = (2/R2)*VR2 - x2;
68  
69      % create a vector 'y' for discrete output signals
70      y(n,1) = Vout;
71  end
72  
73  
```

Figure 4.14 – MATLAB code for the op-amp stage (4)

The code in Figure 4.14 employs a for loop to process each discrete signal comprising the input signal, converting it into the output signal using the transfer function derived during the circuit discretization process. Line 54 indicates that this for loop will execute *N* times, where *N* is the total number of samples in the input signal, specifically 1,235,355. Line 55 designates a single discrete signal from the input signal vector at row n, column 1, as *Vin*. Line 56 implements the transfer function derived from Equation 4.35.b and declares it as *Vout*. Lines 66 and 67 update the state values of *x1* and *x2* with each iteration of the for loop, using the state-update equations derived from Equations 4.39.a and 4.39.b. Lines 59 through 63 set the values of the unknowns included in the state-update equations. Finally, line 70 prints out the *Vout* value, processed through this for loop, into row '*n'* of the output vector '*y'*.

4.2.4. Comparison of frequency response between the reference circuit and the MATLAB code

In this subsection, the fidelity between the *D+*'s op-amp stage circuit and its MATLAB code replica is assessed by comparing their frequency responses. For simulating the op-amp stage, SPICE, known for its high-accuracy physical modeling of individual components, was utilized (Holmes, 2019). The specific program employed here is LT SPICE. To compare frequency responses, a delta function was inputted into MATLAB, while a sine sweep was used as the input signal in SPICE.

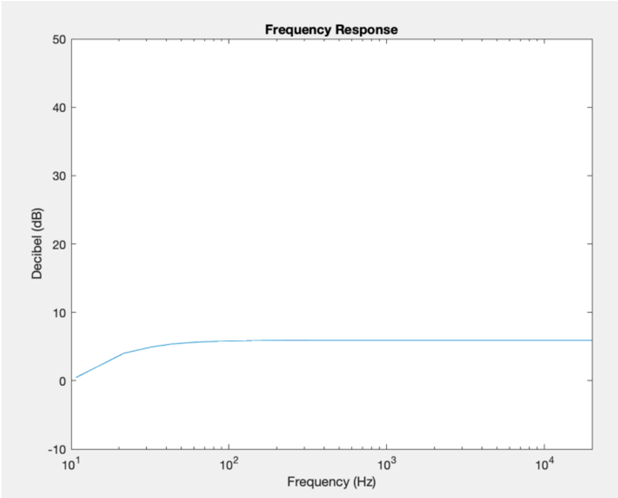

Figure 4.15 – MATLAB Frequency response at 0% pot position

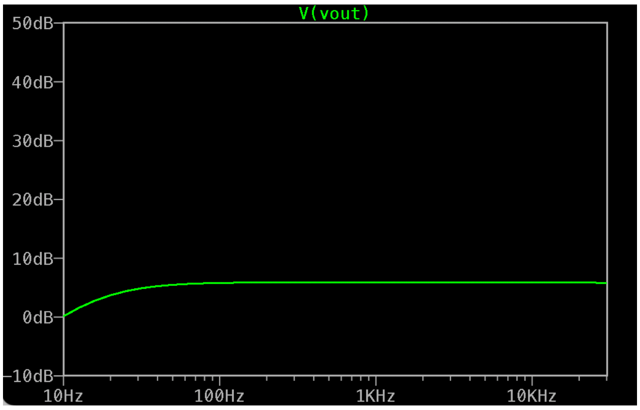

Figure 4.16 – SPICE Frequency response at 0% pot position

Figures 4.15 and 4.16 each display the frequency response results from MATLAB and SPICE when the distortion knob is fully turned down, setting the variable resistor *R6* to its maximum value of 1 Megaohm. The results demonstrate a high level of agreement between MATLAB and SPICE.

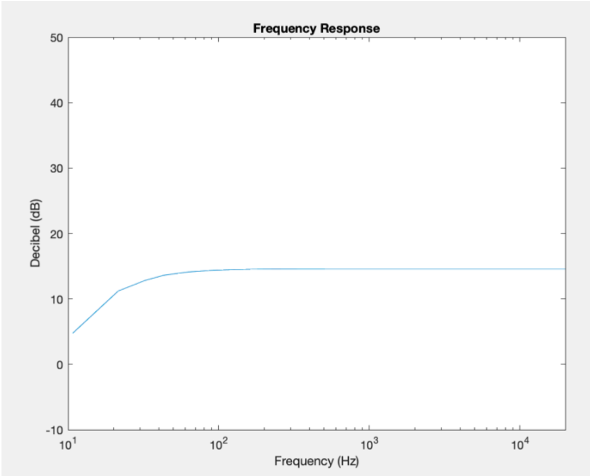

Figure 4.17 – MATLAB Frequency response at 25% pot position

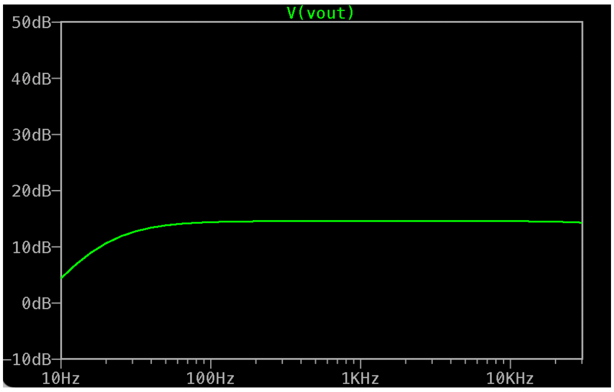

Figure 4.18 – SPICE Frequency response at 25% pot position

Figures 4.17 and 4.18 illustrate the frequency response results from MATLAB and SPICE when the distortion knob is positioned at 25%, setting the variable resistor *R6* to a value of 221,199 ohms. Similarly, the results appear to demonstrate a high level of agreement.

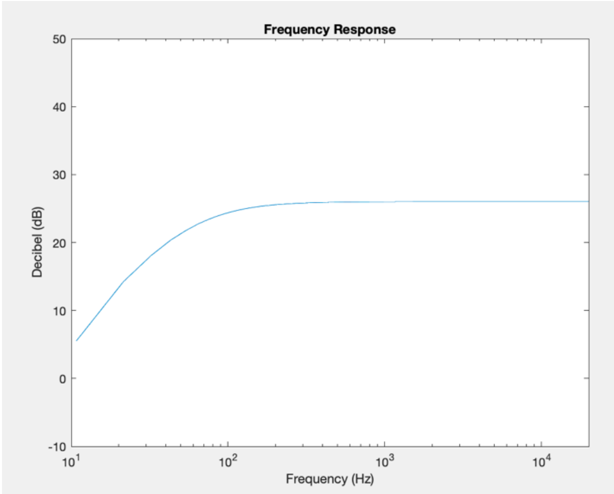

Figure 4.19 – MATLAB Frequency response at 50% pot position

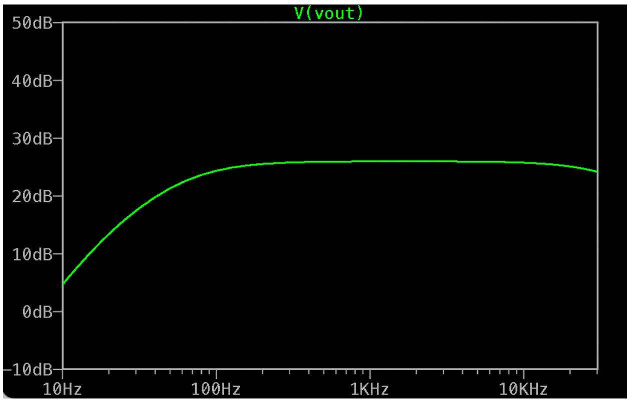

Figure 4.20 – SPICE Frequency response at 50% pot position

Figures 4.19 and 4.20 display the frequency response results from MATLAB and SPICE when the distortion knob is set to 50%, with the variable resistor *R6* at a value of 47,425 ohms. While the cut-off frequency and decibel values align, the SPICE result exhibits a slight roll-off at high frequencies end that is not present in the MATLAB results.

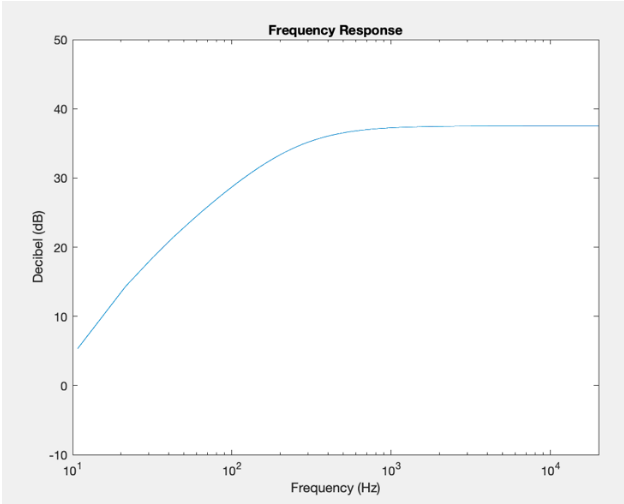

Figure 4.21 – MATLAB Frequency response at 75% pot position

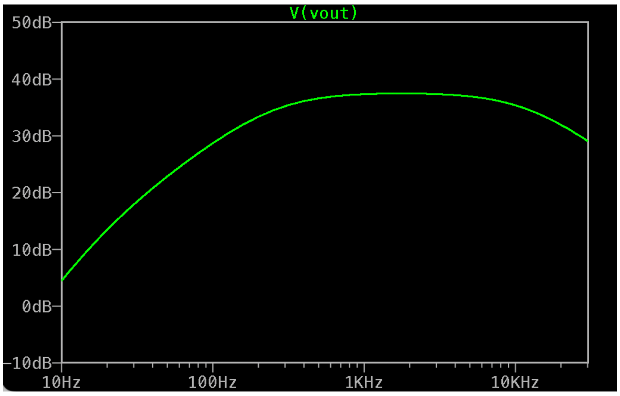

Figure 4.22 – SPICE Frequency response at 75% pot position

Figures 4.21 and 4.22 depict the frequency response results from MATLAB and SPICE when the distortion knob is positioned at 75%, setting the variable resistor *R6* to a value of 8,651 ohms. The SPICE result clearly shows a roll-off in the high-frequency range, where differences from the MATLAB result become apparent.

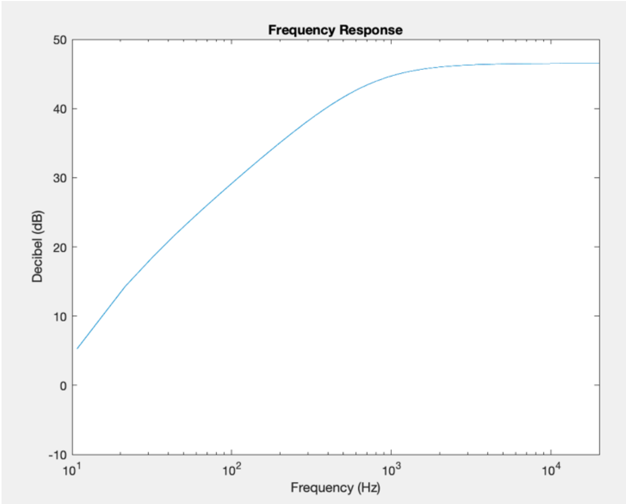

Figure 4.23 – MATLAB Frequency response at 100% pot position

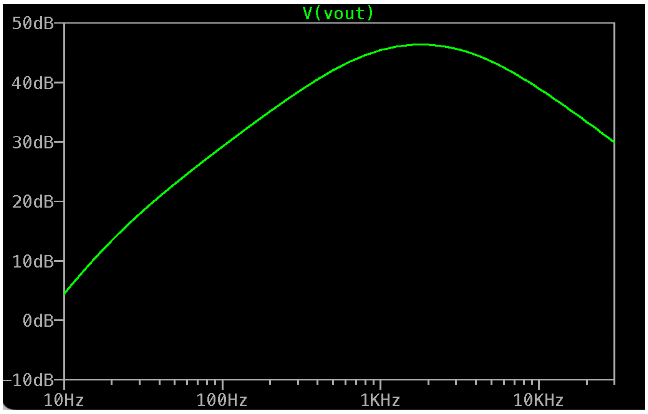

Figure 4.24 – SPICE Frequency response at 100% pot position

Finally, Figures 4.23 and 4.24 illustrate the frequency response results from MATLAB and SPICE when the distortion knob is fully turned to 100%, setting the variable resistor R6 to a value of 0 ohms. Similar to Figure 4.22, Figure 4.24 also exhibits a roll-off in the high-frequency range which doesn't appear in the MATLAB result.

Distortion knob settings above 50% revealed a high-frequency roll-off in SPICE, which was not observed in MATLAB results. This phenomenon is hypothesized to be due to the physical characteristic of operational amplifiers known as the gain bandwidth product, whereby an increased gain setting results in a narrower frequency bandwidth. This behaviour is not replicated during the circuit discretization process because, in this process, the op-amp is assumed to be ideal with the only consideration being that the voltages at the input terminals are equal. Conversely, SPICE incorporate the physical properties of the actual op-amp used, LM741, reflecting its real-world characteristics.

The circuit discussed in this sub-section is the op-amp stage in *D+*, followed by a clipping stage to which a low-pass filter with a cut-off frequency of 15.9 kHz is applied. This arrangement is expected to reduce the difference in frequency response between the original and the replica in the overall circuit. However, the imperfections in replication due to the characteristics of the op-amp still create discrepancies between the actual and modelled results, thereby reducing the accuracy of the replication. Methods to tackle this issue will be discussed in Chapter 5.

4.3. Clipping stage modelling

4.3.1. DK substitution

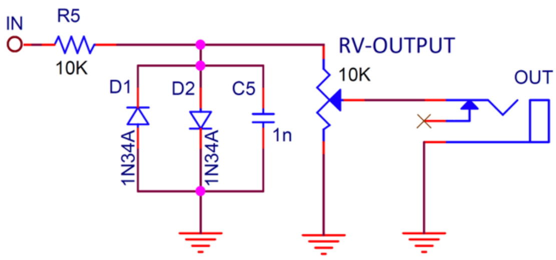

Figure 4.25 – Clipping stage of D+

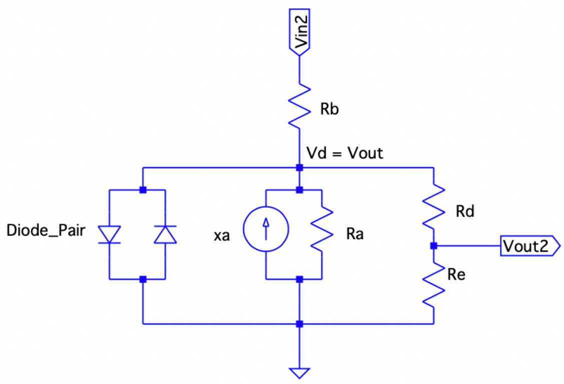

Figure 4.26 – DK substitution circuit of Figure 4.25

Figure 4.26 illustrates the DK substitution circuit for the clipping stage of the *D+* (Figure 4.25). As a result, the original circuit's capacitor *C5* has been replaced with a parallel circuit consisting of resistor *Ra* and a reverse current source *xa*. Calculations for the discretization of the circuit are based on the newly named components in Figure 4.26.

4.3.2. Circuit discretization

In Figure 4.26, the voltage at the node between resistor *Rb* and the parallel circuits below, it can be represented as the voltage V_d across the diode pair circuit, and *Vout* is influenced by a voltage divider circuit dependent on V_d. Applying KCL to this node V_d, Equation 4.41 can be formulated. The left side of the equation represents the current through resistor *Rb*, calculated using NVA and Ohm's Law. The right side, $2 \cdot Is \cdot \sinh\left(\frac{V_d}{\eta \cdot V_T}\right)$ corresponds to the current through a pair of back-to-back diodes circuit, derived from the Shockley diode equation as discussed in Equation 2.15.b of Chapter 2. The following $\frac{V_d}{Ra} - x_a$ represents the current flowing through the DK substitution circuit for the capacitor, and $\frac{V_d}{Rc}$ represents the current flowing through resistor *Rd*.

$$\frac{V_{in}2 - V_d}{Rb} = 2 \cdot Is \cdot \sinh\left(\frac{V_d}{\eta \cdot V_T}\right) + \frac{V_d}{Ra} - x_a + \frac{V_d - V_{out}2}{Rd} \tag{4.41}$$

Consolidate all components of Equation 4.41 to one side to equate to zero (Equation 4.42.a) and establish this equation as a function $f(V_d)$ with respect to V_d (Equation 4.42.b).

$$0 = \frac{V_d}{Rb} - \frac{V_{in}2}{Rb} + 2 \cdot Is \cdot \sinh\left(\frac{V_d}{\eta \cdot V_T}\right) + \frac{V_d}{Ra} - x_a + \frac{V_d}{Rd} - \frac{V_{out}2}{Rd} \tag{4.42.a}$$

$$f(V_d) = \frac{V_d}{Rb} - \frac{V_{in}2}{Rb} + 2 \cdot Is \cdot \sinh\left(\frac{V_d}{\eta \cdot V_T}\right) + \frac{V_d}{Ra} - x_a + \frac{V_d}{Rd} - \frac{V_{out}2}{Rd} \tag{4.42.b}$$

Taking the derivative of the function $f(V_d)$ from Equation 4.42.b results in Equation 4.43.

$$f'(V_d) = \frac{1}{Rb} + \frac{1}{Ra} + \frac{1}{Rd} + \frac{2 \cdot Is}{\eta \cdot V_T} \cdot \cosh\left(\frac{V_d}{\eta \cdot V_T}\right) \tag{4.43}$$

The output $V_{out}2$ for the discrete signal $V_{in}2$ is essentially part of the V_d corresponding to $V_{in}2$. Therefore, N-R iterations described in sub-section 2.4.3 of Chapter 2 must be conducted to find the V_d bringing the function $f(V_d)$ close to zero. This iterative process to find the V_d for all discrete input signals is performed in the MATLAB code in the subsequent subsection.

On the other hand, to derive the transfer function for $V_{out}2$, an equation using KCL is established for the node $V_{out}2$, as shown in Equation 4.44.a. In solving this equation for $V_{out}2$ (Equation 4.44.b), for simplicity, $\left(\frac{1}{R_e} + \frac{1}{R_d}\right)$ is substituted with *Gg*, resulting in the final transfer function for $V_{out}2$ being represented as Equation 4.44.c.

$$\frac{V_d - V_{out}2}{R_d} = \frac{V_{out}2}{R_e} \tag{4.44.a}$$

$$V_{out}2 \cdot \left(\frac{1}{R_e} + \frac{1}{R_d}\right) = \frac{V_d}{R_d} \tag{4.44.b}$$

$$V_{out}2 = \frac{V_d}{R_d \cdot G_g} \tag{4.44.c}$$

4.3.3. DSP implementation in MATLAB

4.3.3.1. Flowchart

Figure 4.27 presents a flowchart for the MATLAB script developed based on the circuit discretization results for the clipping stage of the *D+*. A detailed analysis of the code is conducted in the subsequent subsection.

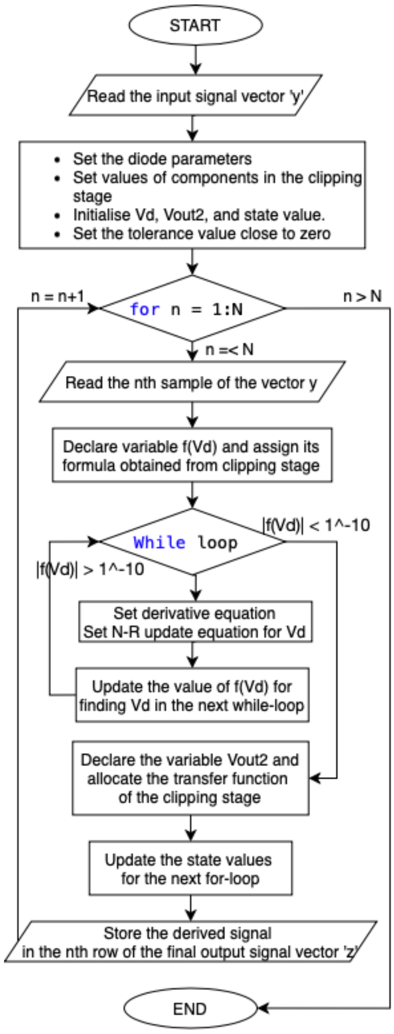

Figure 4.27 – Flowchart of MATLAB codes for the clipping stage

4.3.3.2. Code Analysis

The MATLAB code to be discussed in this sub-section follows the code as covered in sub-section 4.3.3 for replicating the op-amp stage and plays a role in digitally replicating the electrical behavior of the *D+*'s clipping stage within the comprehensive MATLAB script that replicates the entire *MXR D+* circuit. The integrated code encompassing the complete circuit, including this segment, is in the appendix.

```
74    % clipping stage
75
76    % Germanium Diode parameters
77    Is = 100e-9; % saturation current
78    Vt = 0.026; % thermal voltage
79    eta = 2;% emission coefficient
80
81    % clipping-stage component values w/ DK substitution
82    Rb = 10000;
83    Ca = 1e-9;
84    Ra = Ts/(2*Ca);
85
86    % "OUTPUT" POT position from 0 to 0.99
87    outputpot = 0.76;
88
89    Re = 10000 * (1 - log10(1 + 9 * (1 - outputpot)));
90    Rd = 10000 - Re;
91
92    Gg = (1/Rd) + (1/Re);
93
94
```

Figure 4.28 – MATLAB code for the clipping stage (1)

Figure 4.28 represents the initial segment of the code. From line 77 to line 79, the parameters of the germanium diode used in the reference circuit, such as saturation current, thermal voltage, and the emission coefficient, are declared and assigned values as variables *Is*, *Vt*, and *eta*, respectively. The meanings and values of these variables were investigated in sub-section 2.4.2 of Chapter 2. Line 82 declares a variable for the resistor *Rb* from Figure 4.26. Subsequent lines 83 and 84 declare and assign values for the substituted resistor *Ra* of the DK substitution circuit for the capacitor and declare the capacitance (Line 83) required to represent the value of *Ra* (Line 84). Line 87 declares the variable '*outputpot'* to represent the position of the OUTPUT potentiometer in Figure 4.25. Lines 89 and 90 use an equation typical of audio taper used for volume adjustment to represent the resistance value of a variable resistor as a voltage divider. Line 92 assigns the *G*-value substituted in Equation 4.44.b for ease of calculation.

```
95     Vd = 0; % initial guess of Vd
96     Vout2 = Vd/(Rd*Gg); % initial Vout2 val.
97
98     TOL = 1e-10; % a very small value close enough to zero
99     xa=0; % initial state value for DK substitution circuit of a capacitor
100
```

Figure 4.29 – MATLAB code for the clipping stage (2)

Line 95 in Figure 4.29 declares a variable representing the node V_d and arbitrarily assigns an initial guess value for V_d to be used in the first iteration of the N-R iteration process. Line 96 represents the transfer function derived from Equation 4.44.c. Line 98 declares a variable *TOL*, representing tolerance, to conserve computational resources by stopping the iteration process once the value of the function $f(V_d)$ is sufficiently close to zero, though not exactly zero; a very small value is arbitrarily assigned to indicate sufficient proximity to zero. Line 99 declares the variable *xa* as the state value of the DK substitution for the capacitor and assigns an initial value of zero to it.

```
101     % clipping-stage sample-by-sample processing
102     for n = 1:N
103
104         Vin2 = y(n,1); % discrete input signal
105
106         fVd = 2*Is*sinh(Vd/(eta*Vt))+(Vd/Rb)+(-Vin2/Rb)+(Vd/Ra)-xa + (Vd/Rd)+(-Vout2/Rd);
107
108         count=0;
109
110         % a nested loop to find the Vd value for each input
111         while((abs(fVd) > TOL) && (count<10))
112
113
114             der = ((2*Is/(eta*Vt)) * cosh(Vd/(eta*Vt))) + (1/Ra) + (1/Rb) + (1/Rd);
115
116             Vd = Vd - fVd/der; % N-R update equation
117
118             fVd = 2*Is*sinh(Vd/(eta*Vt))+(Vd/Rb)+(-Vin2/Rb)+(Vd/Ra)-xa + (Vd/Rd)+(-Vout2/Rd);
119             count = count+1;
120
121         end
122
123         Vout2 = Vd/(Rd*Gg);
124         xa = ((2/Ra) * Vd) - xa; % state-update equation
125
126         % create a vector 'z' for the final discrete output signals
127         z(n,1) = Vout2;
128
129     end
130
```

Figure 4.30 – MATLAB code for the clipping stage (3)

Figure 4.30 illustrates the code that applies processing to convert discrete input signals into discrete output signals for the clipping stage system. This code processes the samples from the output signal vector *'y'*, derived from the op-amp stage's signal processing discussed in Subsection 4.2.3.2. Thus, under the For loop (line 102), the discrete input signals corresponding to the n^{th} row of the signal vector '*y*' are read as *Vin2* (line 104). Line 106 declares the variable *fVd* for the function *f(Vd)* defined in Equation 4.42.b.

The output signal *Vout2* for the input *Vin2*, as observed in Figure 4.26, corresponds to a portion of the *Vd* which is the voltage across the pair of diodes circuit. Therefore, to find *Vout2*, it is first necessary to determine the value of *Vd* for each input signal. This is achieved through a While loop within the For loop, utilizing N-R iterations. The conditions set in line 111, where the absolute value of the confirmed *fVd* remains greater than *TOL*, and the loop iterates fewer than ten times, are designed to limit the time and computational resources required to find the *Vd* for each discrete input signal *Vin2*. Inside the While loop, the derivative of the function *f(Vd)* derived in Equation 4.43 is declared as the variable '*der*' (line 114). Subsequently, the *Vd* is updated using the N-R update equation from Equation 2.16.b in Chapter 2 (line 116). In line 118, the updated *Vd* is used to recalculate the function *fVd*, and the iteration either continues or exits the while loop based on its conditions. Once the *Vd* is determined and the while loop is exited, the obtained *Vd* is applied in line 123 to derive the transfer function for *Vout2*. Line 124 updates the state value for the capacitor's substitution circuit in preparation for the next for loop. Finally, line 127 declares the final output signal vector '*z*' and stores the discrete output signals derived from the system for each discrete input signal.

```
131
132     audiowrite('input.wav', input, Fs);
133     [input, Fs] = audioread('input.wav');
134
135     audiowrite('output.wav', z, Fs);
136     [output,Fs] = audioread('output.wav');
137
138     % audioplayer for output
139     output=audioplayer(output,Fs);
140     input=audioplayer(input,Fs);
141
```

Figure 4.31 – MATLAB code for the clipping stage (4)

Figure 4.31 represents the last segment of the complete MATLAB script intended for auditory verification of signals before and after processing. Lines 132 and 135 employ the *audiowrite()* function to export the input signal vector '*input*' and the output signal vector '*z*' into respective wave files. Lines 139 and 140 utilize the *audioplayer()* function to facilitate playback of these extracted wave files following the execution of the script.

4.3.4. Result analysis with integrated scripts

In this section, the entire MATLAB script processes an input signal of a 1 kHz sine wave with an amplitude of 0.075, while the distortion knob is set to 75%. The waveform and DFT (Discrete Fourier Transform) results of the output signal after passing through the system are compared with the outcomes from a SPICE simulation.

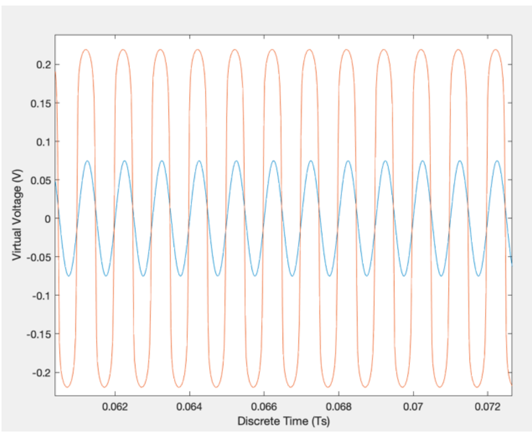

Figure 4.32 – MATLAB waveform result

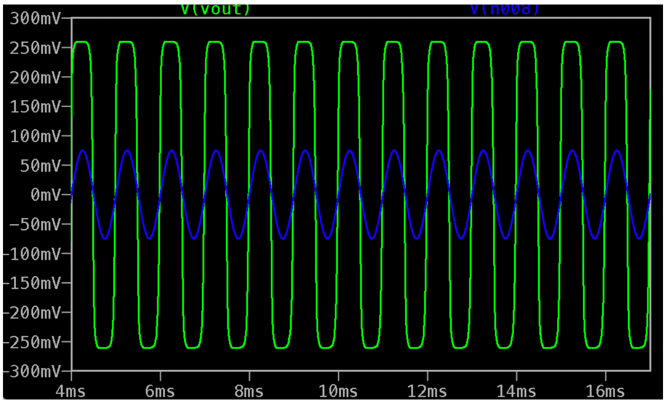

Figure 4.33 – SPICE waveform result

In Figures 4.32 and 4.33, the red and green lines respectively represent the output waveforms of signals processed through MATLAB and SPICE. While these appear to be similar, closer inspection reveals that the waveform in Figure 4.33 (SPICE result) exhibits hard clipping not observed in Figure 4.32. This discrepancy arises because, in the circuit simulation, the distortion knob is adjusted to a level where the op-amp gain increases sufficiently to cause the signal to clip against the op-amp's power supply rails. In the digital replication process via circuit discretization, the power supply rails are not considered, leading to this inconsistency.

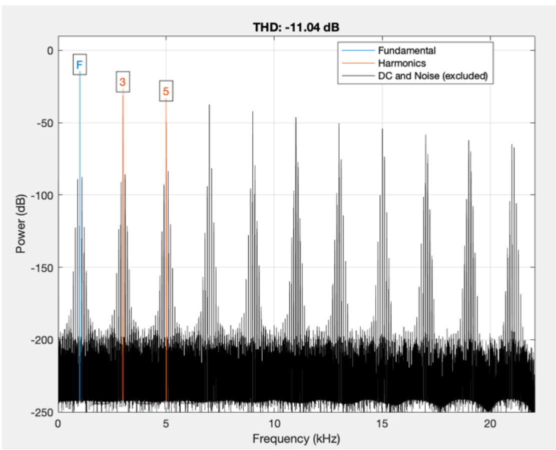

Figure 4.34 – MATLAB DFT result

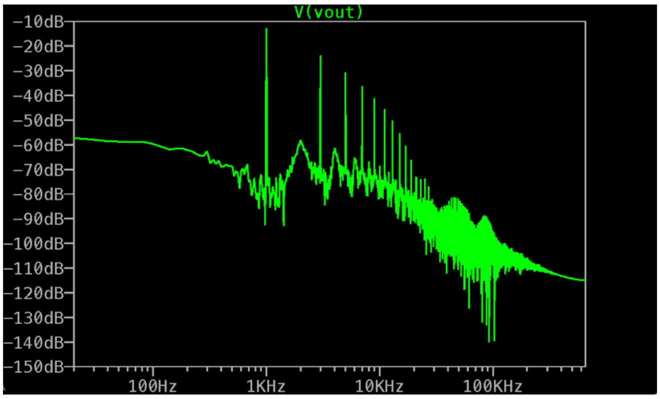

Figure 4.35 – SPICE FFT result

When comparing the DFT/FFT results depicted in Figures 4.34 and 4.35 from MATLAB and SPICE, respectively, both demonstrate the addition of odd harmonics to the fundamental frequency. However, the levels of these harmonics are approximately 3dB higher in the SPICE results representing more distortion.

4.4. Comparison evaluation with actual pedal measurements

4.4.1. Testing method

In Section 4.4, the conformity of outputs from an actual pedal measurement and its replicated MATLAB script is compared using the same audio input. For the actual pedal measurements, a *D+* pedal was connected to the output of an audio interface, with the pedal's output reconnected to the interface's input to allow the audio input played from the DAW to be processed through the real pedal and then recorded onto a new track in the DAW. The OUTPUT knob controlling the volume was fixed at 50%, and the Distortion knob was adjusted to 25%, 50%, 75%, and 100% positions to extract the output audio files.

The audio files extracted from the actual pedal and the MATLAB script were normalized and then assessed qualitatively for their conformity using the Pearson correlation coefficient. The Pearson correlation coefficient quantifies the strength of the linear relationship between two signals with values ranging from -1 to 1, indicating a strong correlation when exceeding 0.5 and approaching perfect conformity as it nears 1.

4.4.2. Evaluation with Pearson correlation coefficient

Distortion knob position	Pearson correlation coefficient
25%	0.6775
50%	0.6442
75%	0.4649
100%	0.2853

Table 4.1 – Pearson correlation coefficient

Table 4.1 presents the Pearson correlation coefficients calculated between audio files extracted via the MATLAB script and actual pedal measurements based on the settings of the distortion knob. While a strong correlation is evident when the distortion knob is set to 25% and 50%, the conformity between the original and the replicated system signals tends to decrease when the knob exceeds 75%.

Reference lists

Tarr, E. (2022) AET 5895 FA22 [Private Video playlist]. YouTube.

Chowdhury, J. (2019). *Complex Nonlinearities Episode 0: Why*. Available at: https://jatinchowdhury18.medium.com/complex-nonlinearities-episode-0-why-4ad9b3eed60f [Accessed 14February 2024].

Holmes, B. (2019). *Guitar Effects-Pedal Emulation and Identification.* Queen's University Belfast. Available at:

https://pureadmin.qub.ac.uk/ws/portalfiles/portal/169350312/main.pdf [Accessed: 13 February 2024]

Benois, P. (2013) *Guitar Effects-Pedal Emulation and* Identification. Master thesis. Helmut Schmidt Universität. Available at:

https://www.researchgate.net/profile/Piero-Rivera-Benois/publication/263013830_Simulation_Framework_for_Analog_Audio_Circuits_based_on_Nodal_DK_Method/links/00b4953996520b377a000000/Simulation-Framework-

G. Borin, G. De Poli, and D. Rocchesso*, "Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems*," Speech and Audio Processing, IEEE Transactions on, vol. 8, no. 5, pp. 597–605, 2000.

Eichas, F., Zölzer, U. *Gray-box Modelling of Guitar Amplifiers.* AES. Available at:

https://www.aes.org/tmpFiles/elib/20240216/19874.pdf [Accessed: 12 February 2024]

Chowdhury, J. (2020). *A Comparison of Virtual Analog Modelling Techniques for Desktop and Embedded Implementations.* arXiv. Available at:

https://arxiv.org/pdf/2009.02833.pdf [Accessed: 14 February 2024]

Miklanek, S., Wright, A. (2023). *Neural Grey-Box Guitar Amplifier Modelling with Limited Data.* DAFx23. Available at:

https://acris.aalto.fi/ws/portalfiles/portal/123416084/Miklanek2023_Neural_grey_box_guitar_amp.pdf [Accessed: 12 February 2024]

ElectroSmash. (n.d.). MXR Distortion Plus Analysis. Retrieved from https://www.electrosmash.com/mxr-distortion-plus-analysis

Reiss, J. and McPherson, A. (2014) Audio Effects. 1st edn. CRC Press. Available at: https://www.perlego.com/book/1604499 (Accessed: 16 May 2024).

Self, D. (2020) Small Signal Audio Design. 3rd edn. Focal Press. Available at: https://www.perlego.com/book/1524218 (Accessed: 27 May 2024).

Signalsmith Audio (2022) Warm Distortion. Available at:

https://signalsmith-audio.co.uk/writing/2022/warm-distortion/ (Accessed: 27 May 2024).

Electronics Tutorials (no date) DC Circuits. Available at:

https://www.electronics-tutorials.ws/dccircuits/dcp_6.html (Accessed: 27 May 2024).

Matrix Technology Solutions Ltd (no date) Diode Characteristics. Available at: https://www.matrixtsl.com/courses/ecc/index.php?n=Diodes.DiodeCharacteristics (Accessed: 27 May 2024).

Appendix

MATLAB Script

```
clear; clc;
% This script is emulating the signal processing of
% MXR Distortion+ guitar pedal in digital


%sample rate and period
Fs = 44100;
Ts = 1/Fs;

% Op-amp stage

% component values of the DK substitution circuit
C1 = 10e-9;
R1 = Ts/(2*C1);

C2 = 47e-9;
R2 = Ts/(2*C2);

R3 = 10e3;
R4 = 1e6;
R5 = 4.7e3;
R8 = 1e6;

% "Distortion" RV-POT position from 0 to 1
% 1-Mega Ohm reverse-log pot
pot = 1; % from 0 to 1
k = 8; % sensitivity factor of the reverse-log pot
R6 = (exp(-k * pot) - exp(-k)) / (1 - exp(-k)) * 1000000;
Rn = R5 + R6; % from 4.7k + 0  to  4.7k + 1Meg

% transfer function's G-values
Ga = 1 + (R3/R1);
Gb = 1 + (R2/Rn);
Gh = 1 + (R4/(Gb*Rn));
Gx = (1/R8) + (1/(Ga*R1));

% transfer function's coefficients
b0 = Gh/(Ga*R1*Gx);
b1 = ((R3/(Ga*R1))-1)*(Gh/Gx);
b2 = (-R2*R4)/(Gb*Rn);


% input signal: audio file
[input, Fs] = audioread('testaudio.wav');
N = length(input);

% output signal setting
y = zeros(N,1);

% Initial state value for DK substitution circuit of capacitors
x1 = 0; x2 = 0;

% opamp-stage sample-by-sample processing
for n = 1:N
    Vin = input(n,1); % discrete input signal
```

```
    Vout = b0 * Vin + b1 * x1 + b2 * x2;

    % some values included in state-update equations
    Vx = (1/Gh)*Vout + ((R2*R4)/(Gb*Gh*Rn))*x2;
    VR1 = (Vin-Vx+(R3*x1))/Ga;

    VRn = (Vx-(R2*x2))/Gb;
    VR2 = (R2/Rn)*VRn + (R2*x2);

    % state-update equations
    x1 = (2/R1)*VR1 - x1;
    x2 = (2/R2)*VR2 - x2;

    % create a vector 'y' for discrete output signals
    y(n,1) = Vout;
end


% clipping stage

% Germanium Diode parameters
Is = 100e-9; % saturation current
Vt = 0.026; % thermal voltage
eta = 2;% emission coefficient

% clipping-stage component values w/ DK substitution
Rb = 10000;
Ca = 1e-9;
Ra = Ts/(2*Ca);

% "OUTPUT" POT position from 0 to 0.99
outputpot = 0.76;

Re = 10000 * (1 - log10(1 + 9 * (1 - outputpot)));
Rd = 10000 - Re;

Gg = (1/Rd) + (1/Re);


Vd = 0; % initial guess of Vd
Vout2 = Vd/(Rd*Gg); % initial Vout2 val.

TOL = 1e-10; % a very small value close enough to zero
xa=0; % initial state value for DK substitution circuit of a capacitor

% clipping-stage sample-by-sample processing
for n = 1:N

    Vin2 = y(n,1); % discrete input signal

    fVd = 2*Is*sinh(Vd/(eta*Vt))+(Vd/Rb)+(-Vin2/Rb)+(Vd/Ra)-xa +
(Vd/Rd)+(-Vout2/Rd);

    count=0;

    % a nested loop to find the Vd value for each input
    while((abs(fVd) > TOL) && (count<10))
```

```
        der = ((2*Is/(eta*Vt)) * cosh(Vd/(eta*Vt))) + (1/Ra) + (1/Rb) +
(1/Rd);

        Vd = Vd - fVd/der; % N-R update equation

        fVd = 2*Is*sinh(Vd/(eta*Vt))+(Vd/Rb)+(-Vin2/Rb)+(Vd/Ra)-xa +
(Vd/Rd)+(-Vout2/Rd);
        count = count+1;

    end

    Vout2 = Vd/(Rd*Gg);
    xa = ((2/Ra) * Vd) - xa; % state-update equation

    % create a vector 'z' for the final discrete output signals
    z(n,1) = Vout2;

end


audiowrite('input.wav', input, Fs);
[input, Fs] = audioread('input.wav');

audiowrite('output.wav', z, Fs);
[output,Fs] = audioread('output.wav');

% audioplayer for output
output=audioplayer(output,Fs);
input=audioplayer(input,Fs);
```