A Tube Amp Modeling Project

Helmut Keller

This paper describes the theoretical background and the source-code of a software plugin emulating a tube amplifier for electric guitars. The circuit 5E3 of the Fender Tweed Deluxe amplifier is used as the base for the amp model. Some circuit variants and a universal tone stack extend the tonal range of the original circuit. In contrast to existing comparable solutions, this plugin runs in real-time even on fanless tablets, with very low latency, and with plenty resources left for the obligatory IR-loader and an armada of additional guitar effects, if required. The sound quality and the playing experience is at least on par with existing solutions. The models for the tube stages are based on real tube circuits but are simplified as much as possible without sacrificing aspects relevant to sound and playing experience. The plugin format is JSFX. The JSFX script may be loaded into the VST3 or AU plugin YSFX provided by Joep Vanlier, or it can be directly used in the digital audio workstation REAPER by Cockos Inc. YSFX is available free of charge for Windows, MacOS and Linux.

1. Introduction

The author has been using guitar amp simulations since the advent of Tom Scholz' famous "Rockman" in 1982. This was a very small battery powered device with a headphone output targeting in particular practicing at home and recording. It could produce some impressive sounds, in particular that of a guitar amp with a miked loudspeaker, plus some basic effects. Plugged into a PA even gigging with it was possible, and it rather quickly replaced the author's 32-kg-tube-combo. At the time modeling was still completely done in the analog domain. The first digital amp simulator the author deployed was the software "Guitar Rig" from Native Instruments, followed by the "Helix" hardware floorboard by Line 6. However, the author was still not satisfied with the sound quality and the playing experience, and he moved back to a software solution, namely "Gig Performer®" by Deskew Technologies as a VST host (for live performances), and guitar amp plugins provided by Neural DSP for the guitar amp emulation. The authors expectations were nearly satisfied at this stage. However, the above setup did not run with a reasonable low latency setting on fanless Windows tablets like the author's Microsoft Surface Pro 7. Moreover, the desired sound was somewhere in between the "Tone King Imperial Mark II", and the clean channel of the "Soldano SLO 100". While the compression of the former amp was always perceived as too strong, the second amp always felt too "tight". In the end the author decided to write his own guitar amp plugin. So far, emulating the sound of Neil Young's guitar on the live album "Weld" has always been a most difficult task. Given this background, a realistic model of the 5E3 circuit of the Fender Tweed Deluxe amp seemed to be an appropriate goal for the project. Also, for more tonal flexibility some variants of this amp model were deemed to be purposeful, as well.

Chapters 2 to 8 of this paper describe the theoretical background of the plugin. At least a medium level knowledge in mathematics, analysis of electronic circuits, system theory, and digital signal processing is required here. These chapters may, however, be skipped by those only interested in using the plugin. Chapter 9 deals with the implementation in software and requires some basic software engineering knowledge.

N.B.: The author encourages everybody interested in tube amp modeling to use the free-of-charge source code and the underlying DSP block library as a template for their own non-commercial projects, the only condition being that the author is properly cited in publications. Commercially oriented use, however, needs a written permission from the author.

Chapter 9 may be skipped, as well, by those only interested in using the plugin. Chapter 10 is the user manual of the plugin. It describes the installation, GUI, and usage of the plugins. It is written with every guitar player in mind who seeks to use the plugin (free of charge). Chapter 11 is a summary of the paper. Chapter 12 provides literature references.

2. Static nonlinearities and aliasing

Before we dig into the details of tube amps, we need to overcome a principal problem of digital signal processing of nonlinear systems. If an (analog) continuous time signal has passed through a static nonlinear transfer function it will contain spectral components at frequencies which were not present ahead of the nonlinearity. If a (digital) discrete time signal has passed through a static nonlinear transfer function, the additional spectral components above the Nyquist frequency will be found as aliased spectral components below the Nyquist frequency. These artificial aliased spectral components can be audible and annoying even at sample rates of 48 kHz. There a two well-known ways to reduce the probability of audible aliasing artifacts:

- Increase the sample rate where/when nonlinearities are processed
- Use an antiderivative approach for the processing of nonlinearities.

In the antiderivative approach [1] the antiderivative $F(x_n)$ of the nonlinear transfer function $f(x_n)$ is calculated for each input sample x_n. According to equation (2.1), the function $g(x_n)$ approximates the mean value of the actually present and of the previous value of the nonlinear transfer function, but with reduced amplitudes of the aliased spectral components.

$$g(x_n)=\frac{F(x_n)-F(x_{n-1})}{x_n-x_{n-1}}\approx\frac{f(x_n)+f(x_{n-1})}{2} \quad (2.1)$$

In order to avoid division by very small values, the second part of equation (2.1) is used in the case that the absolute value of the difference of the input sample values is lower than a given threshold. This means that both the nonlinear transfer function and its antiderivative need to be calculated for each input sample – because the input sample difference is not predictable. Thus the required processing power is a bit greater than double when compared to the direct solution. In order to be the preferable solution, the antiderivative approach therefore needs to reduce aliasing more significantly than would doubling the sample rate.

In **Fig. 1** we see the spectra of a distorted discrete time signal with no oversampling, with oversampling by a factor of 2, with oversampling by a factor of 4, and with the antiderivative approach. The input signal is a harmonic signal with a fundamental frequency of 1202.5 Hz. The first three harmonics have equal amplitude, the fourth harmonic is attenuated by 6 dB and the fifth harmonic is attenuated by 12 dB. This signal mimics a very high note ouput by an electric guitar, albeit with a non-dispersive string oscillation. Emulating the dispersion of the string is omitted here because additional spectral components would be produced by the distortion circuit, and these additional components could not be easily distinguished from aliasing products. The distortion circuit is a serial connection of two soft-clipping nonlinear transfer functions. Each transfer function is an inverting generalized logistic function of type A with $k_{bias} = 0.4$ and $b = 0$ (see the next chapter for explanation). It emulates a typical preamp stage of a tube amplifier. Each stage is overdriven by 20 dB. In the analog world the spectrum would not hold any components between DC and the fundamental frequency of the input signal. All spectral components we observe in this region are aliased spectral components. The SNR value (signal to noise ratio in dB) displayed in **Fig. 1** describes the ratio of the power of the spectral component at the fundamental frequency to the power of all aliased components in the mentioned frequency range. The antiderivative approach therefore is 4.6 dB better than factor-2-oversampling, and 2.9 dB worse than factor-4-oversampling. It reduces aliasing by 12.3 dB compared to the direct solution. It is worth to note that the input signal used in this example is quite extreme in terms of its high fundamental frequency. For input signals with less high-frequency content, the aliasing would generally be less, and the antiderivative approach would perform even better when compared to the oversampling solutions. Consequently, we decide to use the antiderivative approach in our project. Note that other modern amp-modeling solutions like the newer plugins provided by Neural DSP (the ones with no oversampling switch, introduced in 2021) use a similar approach [2].

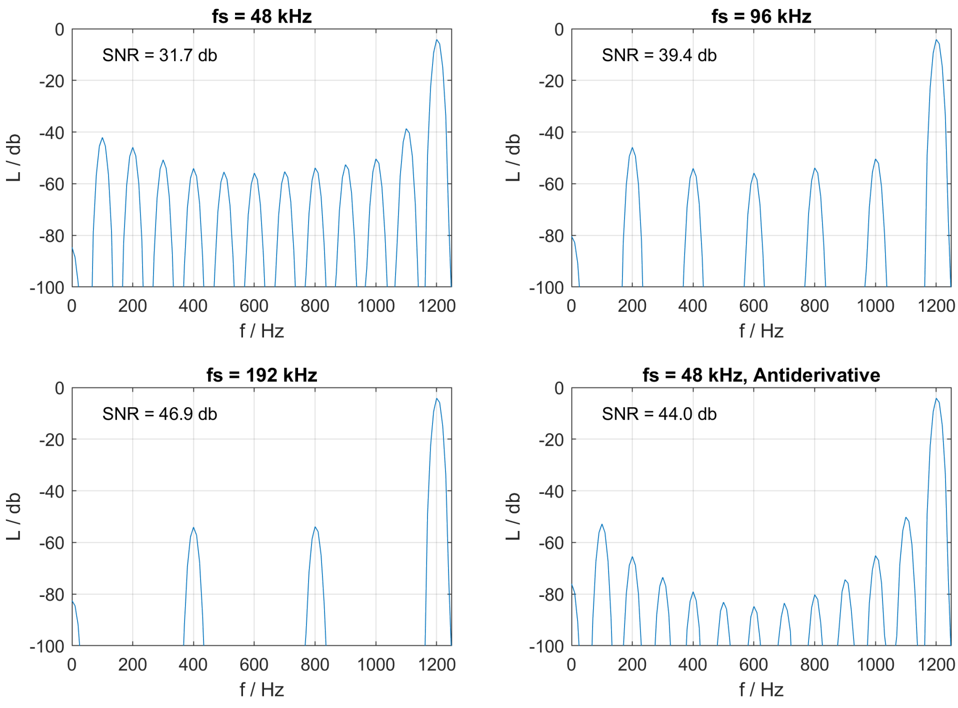


Fig. 1: Spectra of a distorted signal with different sample rates fs

The antiderivative of a nonlinear transfer function is often not available in analytical form. For this reason, and also mainly for efficiency reasons, we use a lookup table of the original nonlinear transfer function and calculate the antiderivative of each lookup table segment. The function is normalized in a first step: For $x = 0$ the function value is set to zero and the gain is set to one. The output swing between positive and negative saturation is set to one. We only use functions which saturate on both sides. The input range of the table is -15 to +15. This assures that using constant output values outside the range of the table still provides a very good approximation even for very softly saturating functions. The input resolution is set to 0.02 which assures that the approximation errors are negligible inside the table range. The table consists of 1500 segments. For each segment we calculate and store 4 polynomial coefficients of the nonlinear transfer function. The polynomial is defined via fitting the original function at the two segment borders and at two equally spaced sample points in between. The five polynomial coefficients of the antiderivative can now be calculated and are stored for each segment too.

From equation (2.1) we can conclude that the antiderivative approach implements not only a desired static nonlinear transfer function. It also acts as low pass filer because two subsequent output samples are averaged. A second order IIR filter can be used to equalize the frequency response up to a cutoff frequency f_c. Frequency response plots (**Fig. 2 to 5**) show the effect of the equalizer with cutoff frequencies of 4, 8, 12 and 16 kHz for a sample rate of 48 kHz. In the figures, the filter coefficients are given, as well. The blue traces show the frequency response of the original lowpass. The red traces show the equalizer, and the black traces show the equalized lowpass.

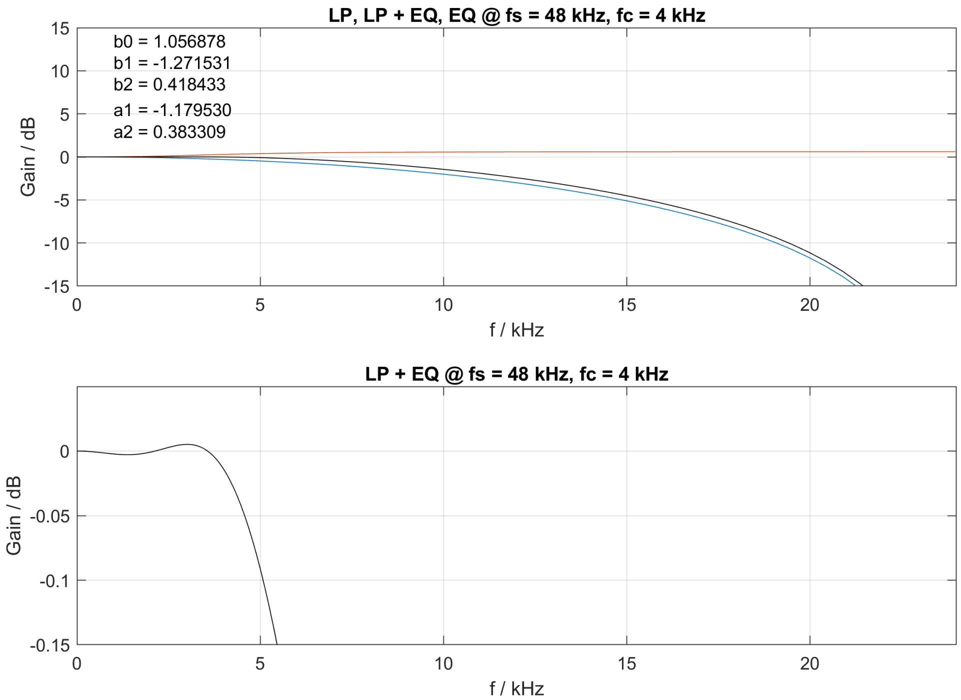


Fig. 2: Equalizer for the antiderivative approach with a cutoff frequency of 4 kHz

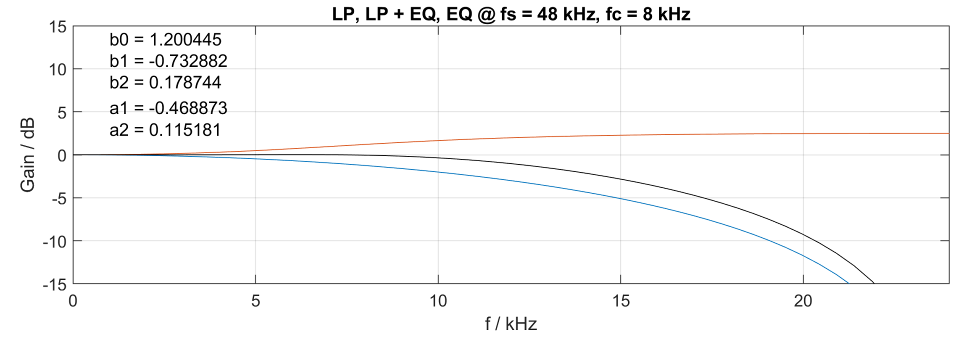


Fig. 3: Equalizer for the antiderivative approach with a cutoff frequency of 8 kHz

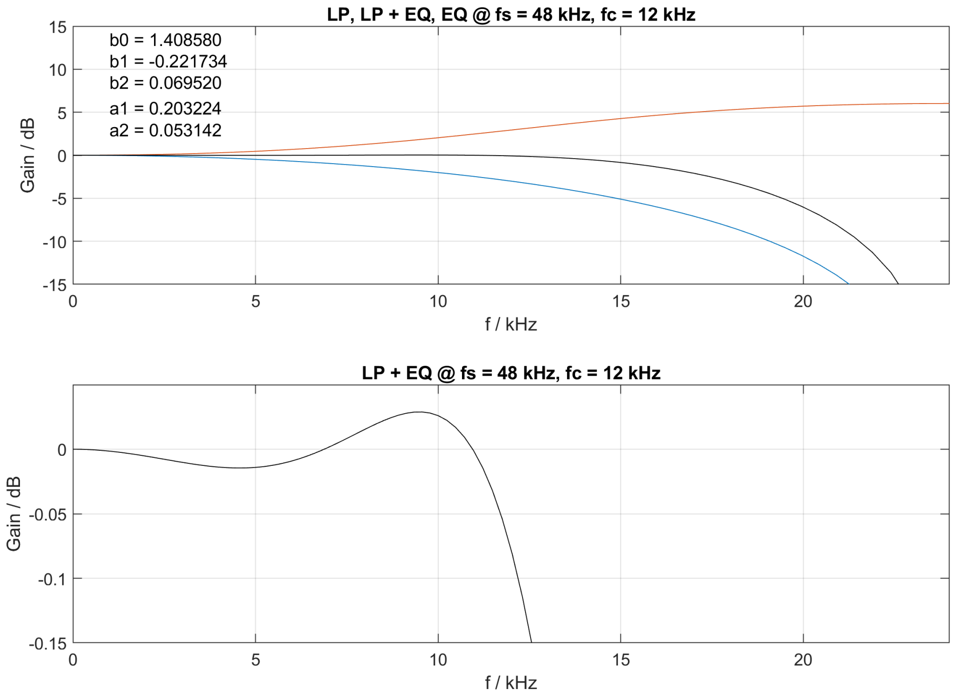


Fig. 4: Equalizer for the antiderivative approach with a cutoff frequency of 12 kHz

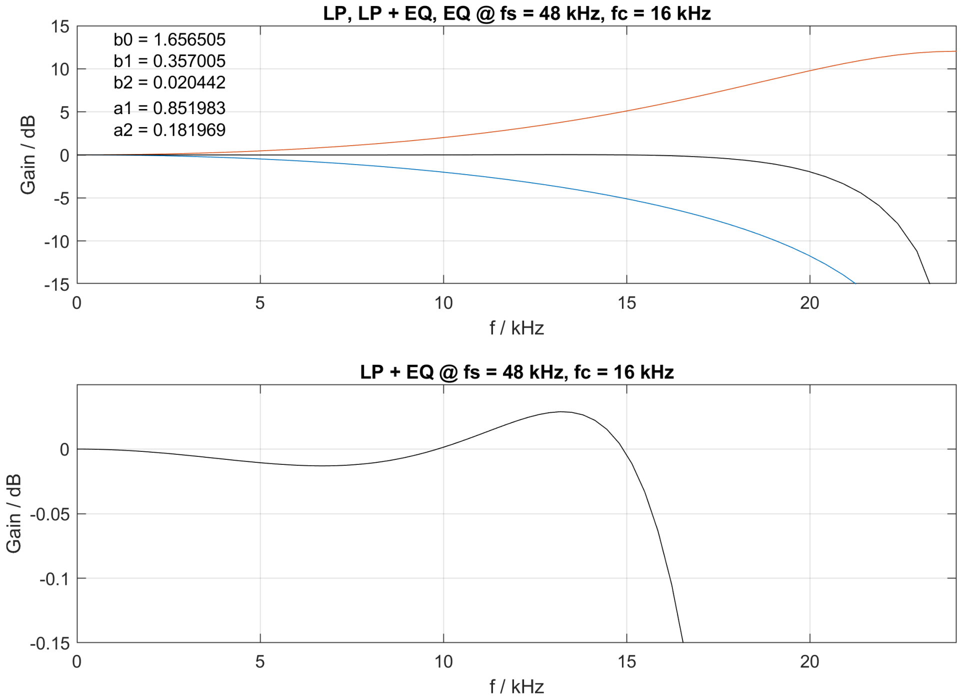


Fig. 5: Equalizer for the antiderivative approach with a cutoff frequency of 16 kHz

3. The generalized logistic function

When modeling tube amps it is necessary to describe the nonlinear transfer functions of different tube stages. For tube stages with local feedback, we can calculate the resulting (harder) clipping transfer function from knowing both the transfer function without local feedback and the closed loop gain. The hyperbolic tangent is a good starting point for a realistic soft clipping transfer function. Real tube stages however have asymmetric transfer functions. The parameter sets of the two types (A and B) of the generalized logistic functions glfa(x, k_{bias}, b) and glfb(x, k_{bias}, b) are identical given a scaled and shifted version of the tanh(x) function for the special case: k_{bias} = 0.5 and b = 0. However, they can also describe asymmetric soft clipping transfer functions of very different shapes. The relation between the two types of the generalized logistic function is:

$$\text{glfb}(x, k_{\text{bias}}, b) = -\text{glfa}(-x, 1-k_{\text{bias}}, -b) \qquad (3.1)$$

Type A is known from literature while type B has been created by the author to obtain additional choices of transfer function shapes. We use a normalized version of the generalized logistic function in this paper: the function value equal zero and the gain equal one at zero input, and the output swing equal one.

The type A of the generalized logistic function is defined in equations (3.2) to (3.4):

$$v = \frac{\ln(k_{\text{bias}})}{\ln(1+e^{b})} \qquad (3.2)$$

$$a = \frac{-\left(1+e^{b}\right)^{(1-v)}}{v \cdot e^{b}} \tag{3.3}$$

$$\text{glfa}\left(x, k_{\text{bias}}, b\right) = \left(1+e^{(b-a\cdot x)}\right)^{v} - k_{\text{bias}} \tag{3.4}$$

The type B of the generalized logistic function is defined in equations (3.5) to (3.7):

$$v = \frac{\ln\left(1-k_{\text{bias}}\right)}{\ln\left(1+e^{-b}\right)} \tag{3.5}$$

$$a = \frac{-\left(1+e^{-b}\right)^{(1-v)}}{v \cdot e^{-b}} \tag{3.6}$$

$$\text{glfb}\left(x, k_{\text{bias}}, b\right) = 1 - \left(1+e^{(-b+a\cdot x)}\right)^{v} - k_{\text{bias}} \tag{3.7}$$

The output value at negative saturation is - k_{bias}. The parameter k_{bias} should be limited to the range from 0.1 to 0.9 in order to avoid numerical problems. The parameter b is proportional to the curvature at the bias point. It should be limited to the range from -4 to +4 to avoid numerical problems.

We can achieve harder-clipping versions of the generalized logistic functions if we embed them in a negative feedback loop (**Fig. 6**).

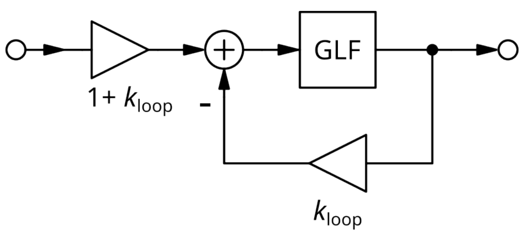


Fig. 6: The generalized logistic function in a negative feedback loop

Note that the small-signal gain of the circuit shown in **Fig. 6** is one. For k_{loop} = 0 we obtain the original soft clipping function. Increasing k_{loop} will provide harder, less “round” clipping. The resulting nonlinear transfer function could be calculated iteratively with the Newton-Raphson-algorithm. Since we work with a lookup table anyway, we can use a more efficient approach. With the input signal of the GLF block in **Fig. 6** being designated x, we can calculate the input variable u of the circuit:

$$u = \frac{x + k_{\text{loop}}\,\text{glf}(x)}{1+k_{\text{loop}}} \tag{3.8}$$

The transfer function glf_fb(u) of the feedback circuit equals glf(x) if u satisfies equation (3.8). We obtain the lookup table of glf_fb(w) for an undistorted input signal w by resampling u and the corresponding function values glf_fb(u) at equidistant values of w. A linear interpolation is sufficient for the resampling process. We use the saturation values for the function values where we need samples of u with corresponding x values outside of the original table range.

Fig. 7 to 15 show the generalized logistic functions of type A and B for nine values of k_{bias} (from 0.1 to 0,9), for five values for b (from -4 to +4), and for the two values of k_{loop} (0 and 100). The working hypothesis of the present project is that this set of functions is sufficient to emulate the static nonlinear transfer functions of real tube stages with very good accuracy in terms of sound perception. The dynamics of the nonlinearities of real tube amp stages will be discussed later in this paper.

With the parameter *type* we can blend between type A and B of the GLF:

$$glf\left(x, k_{\text{bias}}, b, type, k_{\text{loop}}\right) = \left(1 - type\right) \cdot glfa\left(x, k_{\text{bias}}, b, k_{\text{loop}}\right) + type \cdot glfb\left(x, k_{\text{bias}}, b, k_{\text{loop}}\right) \quad (3.9)$$

In **Fig 16.** we observe the effect of *type* for several values of k_{bias} and $b = 0$. For $k_{\text{bias}} = 0.5$ there is no effect at all. The maximum effect is achieved for k_{bias}-values of 0.1. and 0.9.

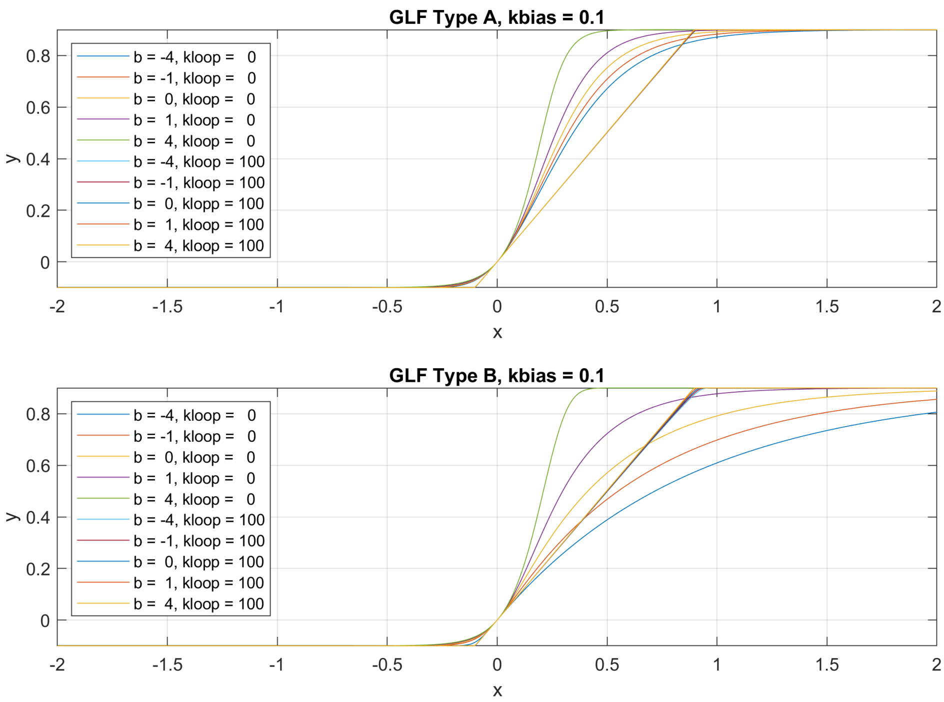


Fig. 7: The generalized logistic functions for $k_{\text{bias}} = 0.1$

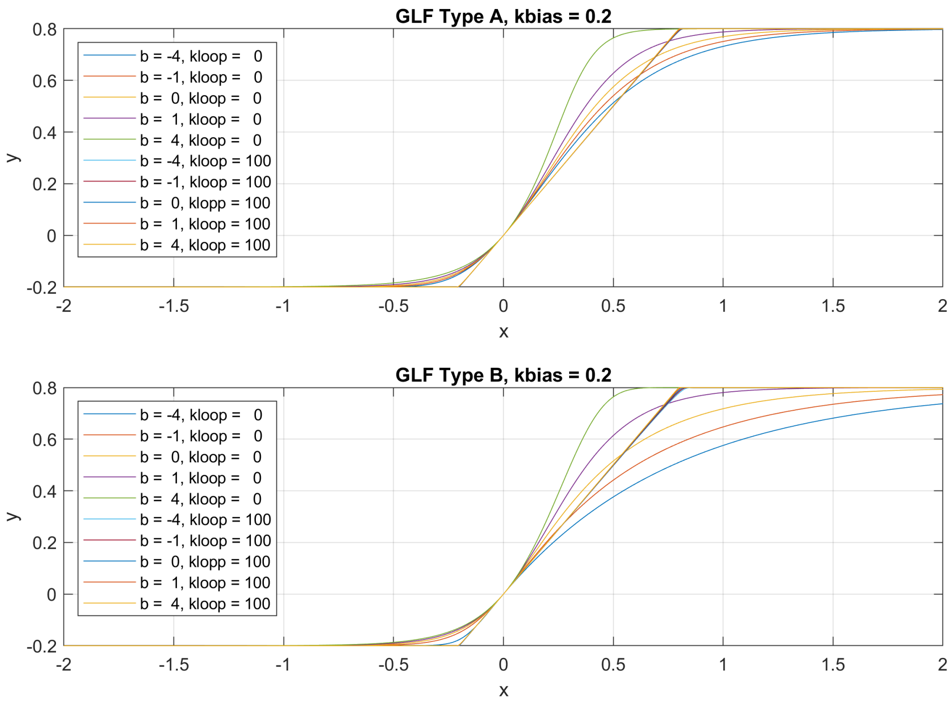


Fig. 8: The generalized logistic functions for k_{bias} = 0.2

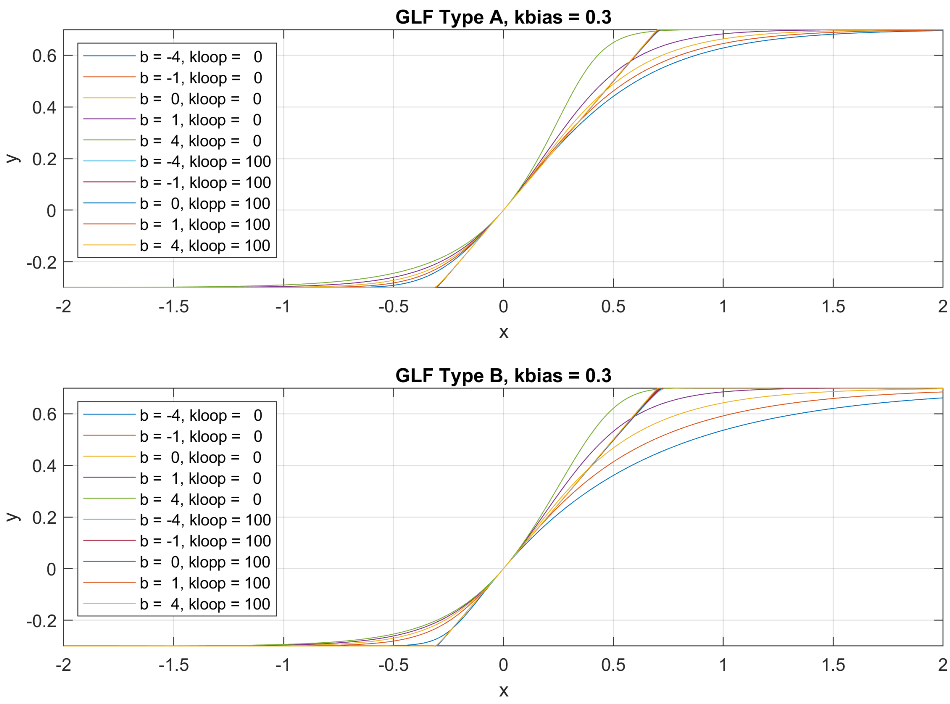


Fig. 9: The generalized logistic functions for k_{bias} = 0.3

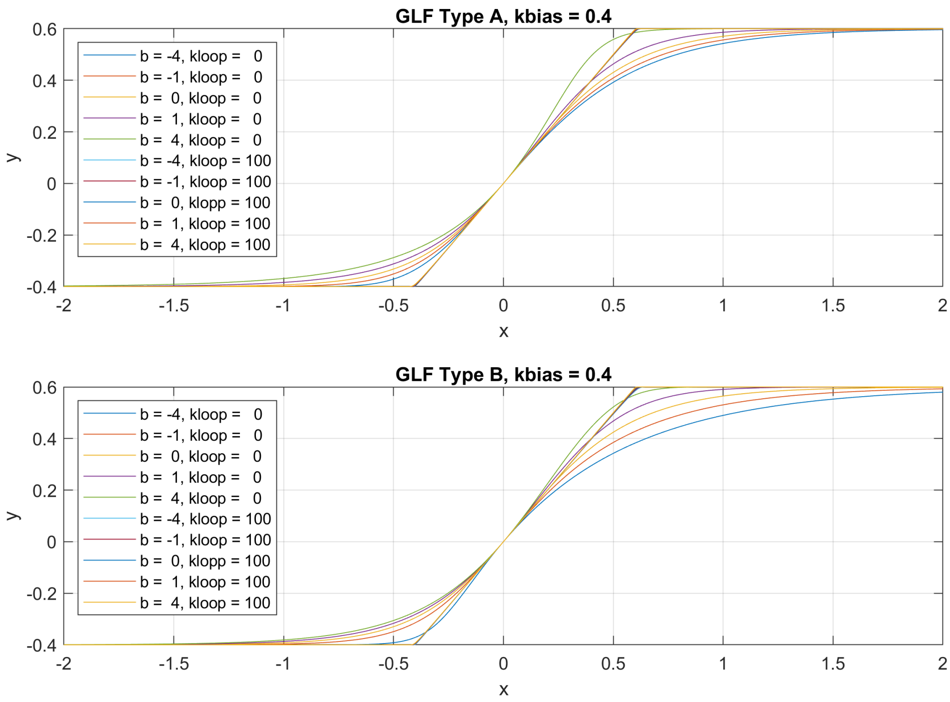


Fig. 10: The generalized logistic functions for $k_{\text{bias}} = 0.4$

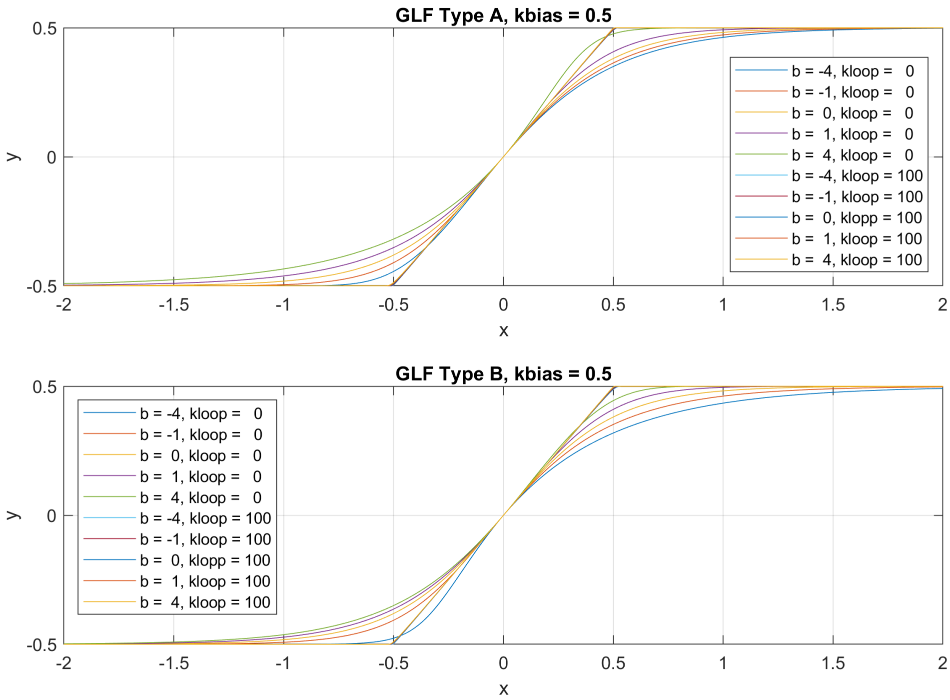


Fig. 11: The generalized logistic functions for $k_{\text{bias}} = 0.5$

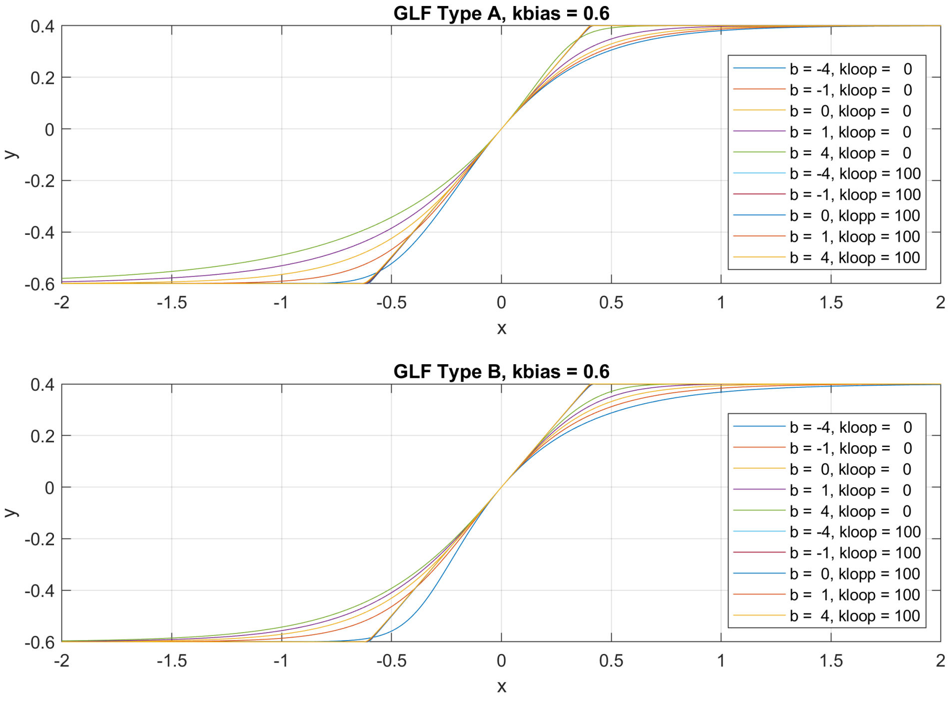


Fig. 12: The generalized logistic functions for $k_{\text{bias}} = 0.6$

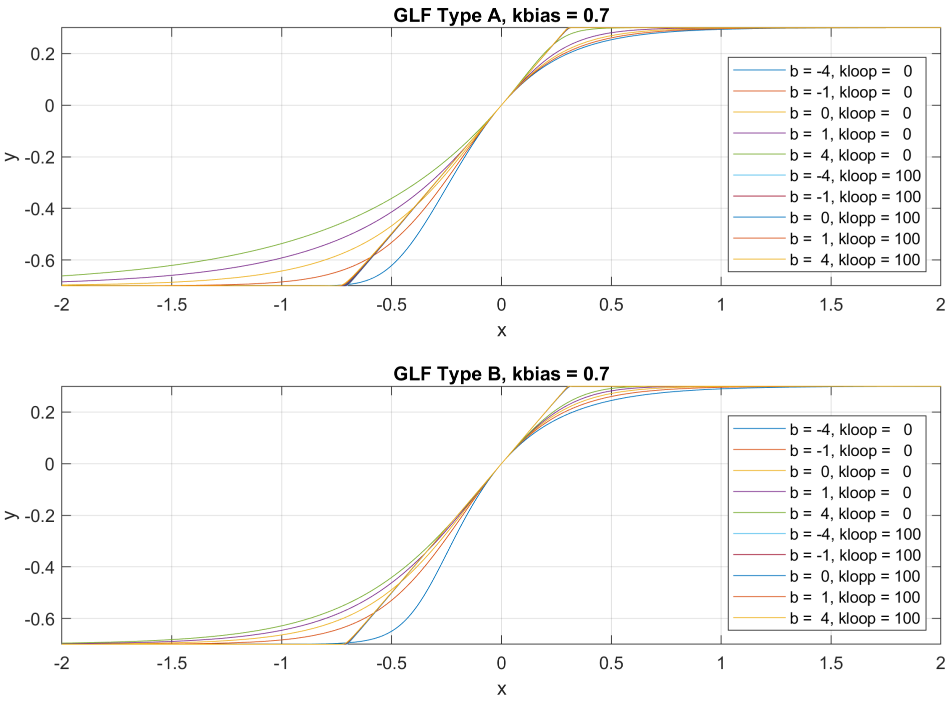


Fig. 13: The generalized logistic functions for $k_{\text{bias}} = 0.7$

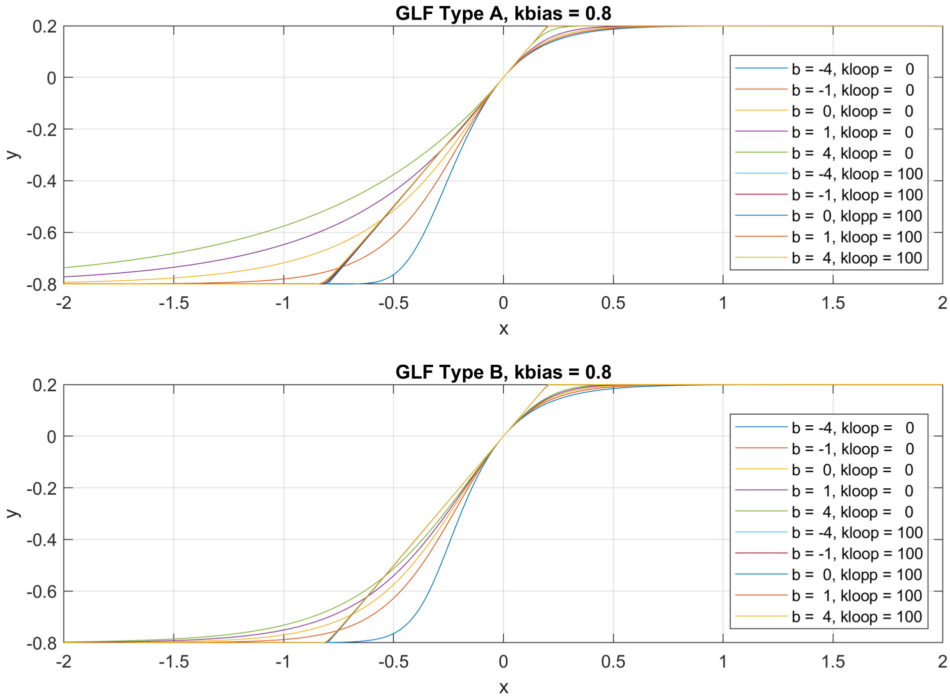


Fig. 14: The generalized logistic functions for $k_{\text{bias}} = 0.8$

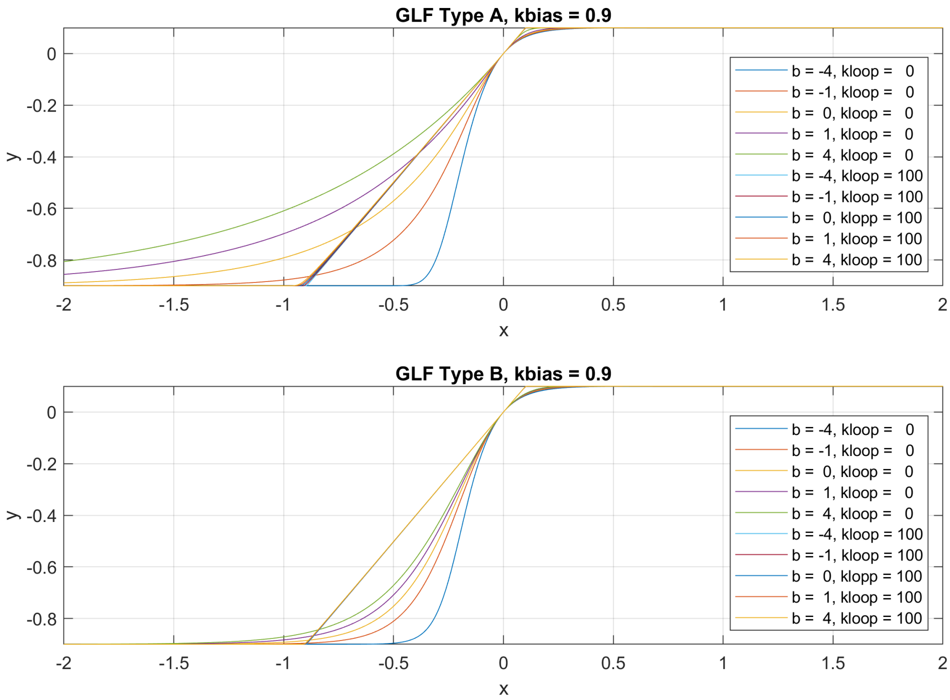


Fig. 15: The generalized logistic functions for $k_{\text{bias}} = 0.9$

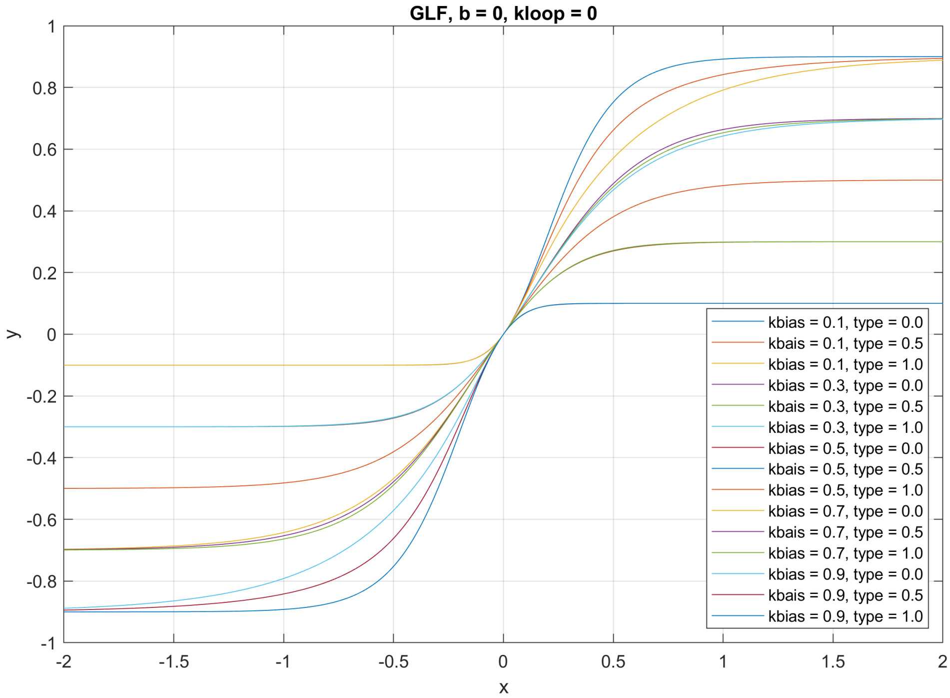


Fig. 16: The generalized logistic functions for some values of k_{bias} and *type*

4. Small-signal models of five basic tube stage circuits

In this chapter we analyze the small-signal behavior of the five commonly used tube stage circuits as shown in **Fig. 17**:

- Common cathode circuit
- Common cathode circuit fitted with a bypass capacitor C_K parallel to the cathode resistor R_K
- Cathode follower circuit
- Cathodyne circuit
- Long tail pair circuit

In a first step we will analyze the following small signal properties

- voltage gain *g* from all inputs to all outputs
- input impedances Z_{in} for the cathodyne and long tail pair circuit (otherwise assumed infinitely high) for all inputs
- output impedances Z_{out} at all outputs
- sensitivity to variations of the power supply voltage, k_{SV}, at all outputs

For small signals the tube can be simplified as a voltage-controlled voltage source with gain $-\mu$ and output impedance R_a. The control voltage is the voltage between grid and cathode. The controlled voltage is the voltage between anode and cathode.

The parasitic capacitances of the tube are disregarded to begin with in a first step. The capacitance between anode and grid multiplied by (1 - g) acts as an input capacitor (Miller-effect) and can be relevant when "high gain" meets "high source impedance". A first-order low pass filter at the input of the stage would have to be included in the model in this case.

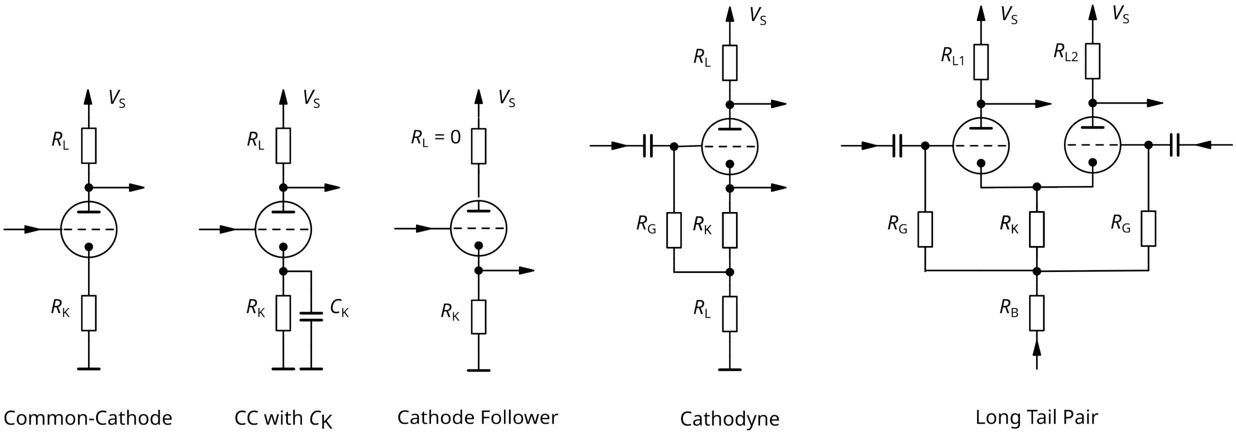


Fig. 17: Five common tube stage circuits

Common cathode – small-signal model

This circuit is used in tube amps where medium gain values are required.

$$g_A = \frac{-\mu R_L}{R_L + R_a + (1+\mu) R_K} \tag{4.1}$$

$$Z_{outA} = \frac{R_L \left(R_a + (1+\mu) R_K\right)}{R_L + R_a + (1+\mu) R_K} \tag{4.2}$$

$$k_{SVA} = \frac{R_a + (1+\mu) R_K}{R_L + R_a + (1+\mu) R_K} \tag{4.3}$$

Common cathode fitted with a cathode capacitor – small-signal model

This circuit is used in tube amps where high gain values are required. Its small signal properties depend on the signal frequency f and can be derived from equations (4.1) to (4.3) by the following substitution:

$$R_K \rightarrow \frac{R_K}{1 + j\,2\pi f R_K C_K} \tag{4.4}$$

The frequencies f_1 and f_2 mark the borders of three regions:

$$f_1 = \frac{1}{2\pi R_K C_K} \tag{4.5}$$

$$f_2 = \frac{R_L + R_a + (1+\mu) R_K}{R_L + R_a} f_1 \tag{4.6}$$

For frequencies much lower than f_1, equations (4.1) to (4.3) are good approximations of the small-signal properties. For frequencies much higher than f_2, the equations (4.7) to (4.9) prove to be good approximations. The small signal properties, however, change rapidly with varying frequency in the range between f_1 and f_2.

$$g_A = \frac{-\mu R_L}{R_L + R_a} \tag{4.7}$$

$$Z_{outA} = \frac{R_L R_a}{R_L + R_a} \tag{4.8}$$

$$k_{SVA} = \frac{R_a}{R_L + R_a} \tag{4.9}$$

Cathode follower – small-signal model

This circuit is often used downstream of a common cathode stage (with or without C_K) to reduce the output impedance. The approximations in the equations (4.10), (4.11) and (4.12) are valid for the normal use case where R_L equals zero, and where R_a and R_K are in the same order of magnitude.

$$g_K = \frac{\mu R_K}{R_L + R_a + (1+\mu) R_K} \approx \frac{\mu}{2+\mu} \tag{4.10}$$

$$Z_{outK} = \frac{(R_L + R_a) R_K}{R_L + R_a + (1+\mu) R_K} \approx \frac{R_a}{2+\mu} \tag{4.11}$$

$$k_{SVK} = \frac{R_K}{R_L + R_a + (1+\mu) R_K} \approx \frac{1}{2+\mu} \tag{4.12}$$

Cathodyne – small-signal model

This circuit is used as a phase splitter. It feeds the two branches of a push-pull power amp with signals of opposite polarity.

$$g_A = \frac{-\mu R_L}{R_L + R_a + (1+\mu)(R_K + R_L)} \approx \frac{R_L}{R_K + R_L} \cdot \frac{\mu}{3+\mu} \tag{4.13}$$

$$g_K = \frac{\mu (R_K + R_L)}{R_L + R_a + (1+\mu)(R_K + R_L)} \approx \frac{\mu}{3+\mu} \tag{4.14}$$

$$Z_{in} = \frac{R_G}{1 - g_K \frac{R_L}{R_K + R_L}} \approx \frac{R_G}{\frac{3 R_L}{(3+\mu)(R_K + R_L)} + \frac{R_K}{R_K + R_L}} \approx \frac{R_G}{\frac{3}{\mu} + \frac{R_K}{R_L}} \tag{4.15}$$

$$Z_{outA} = \frac{R_L \left(R_a + (1+\mu)(R_K + R_L)\right)}{R_L + R_a + (1+\mu)(R_K + R_L)} \approx R_L \tag{4.16}$$

$$Z_{outK} = \frac{(R_L + R_a)(R_K + R_L)}{R_L + R_a + (1+\mu)(R_K + R_L)} \approx \frac{R_L + R_a}{3+\mu} \tag{4.17}$$

$$k_{SVA} = \frac{R_L + R_a + (1+\mu) R_K}{2 R_L + R_a + (1+\mu) R_K} \tag{4.18}$$

$$k_{SVK} = \frac{R_L + R_K}{2 R_L + R_a + (1+\mu) R_K} \tag{4.19}$$

For equations (4.18) and (4.19), we assume that no current is flowing through the coupling capacitor, i.e. these equations are only valid for DC and very low frequencies. For the other equations of the cathodnye circuit, we assume that there is no voltage drop across the coupling capacitor, i.e. they are valid in the audio frequency range only.

Long tail pair – small-signal model

Like the cathodyne arrangement, this circuit is used as a phase splitter. However, it does provide more gain, larger output voltage swing, and a better balance of the two output impedances. The circuit has three inputs for signals, namely (from left to right in **Fig. 17**): v_{in1}, v_{inK} and v_{in2}. Often only v_{in1} is used as a signal input and the other two inputs are connected to ground. For power amps which use global feedback v_{in2}, or both v_{in2} and v_{inK} are used as feedback inputs. The circuit has two voltage outputs (from left to right in **Fig. 17**): v_{out1} and v_{out2}. The first output signal has the opposite polarity of the first input signal. The second output carries a phase inverted version of the first output signal i.e. it is in phase with the first input signal.

The analysis of this circuit is rather complex; its equations may be simplified by using the two effective cathode resistor values R_{K1} and R_{K2}. These are loading the first and second tube, respectively, when the input of the respective other tube and the cathode input are connected to ground.

$$R_{K1} = \frac{(R_K + R_B)(R_a + R_{L2})}{(1+\mu)(R_K + R_B) + (R_a + R_{L2})} \approx \frac{R_a + R_{L2}}{1+\mu} \tag{4.20}$$

$$R_{K2} = \frac{(R_K + R_B)(R_a + R_{L1})}{(1+\mu)(R_K + R_B) + (R_a + R_{L1})} \approx \frac{R_a + R_{L1}}{1+\mu} \tag{4.21}$$

Often the load resistors R_{L1} and R_{L2} are equal. In this case the negative value of the gain from input 1 to output 1 ($-g_{11}$) is a bit greater than the gain from input 1 to output 2 (g_{12}). This can be compensated for if R_{L1} is a bit smaller than R_{L2}. The approximations on the right side of the long-tail-pair equations are right on target if the product (1 + μ) ($R_B + R_K$) is much bigger than ($R_a + R_{L1}$) and ($R_a + R_{L2}$).

$$g_{11}=\frac{-\mu R_{L1}}{R_a+R_{L1}+(1+\mu)R_{K1}}\approx\frac{-\mu R_{L1}}{2R_a+R_{L1}+R_{L2}} \tag{4.22}$$

$$g_{12}=\frac{\mu R_{K1}}{R_a+R_{L1}+(1+\mu)R_{K1}}\cdot\frac{(1+\mu)R_{L2}}{R_a+R_{L2}}\approx\frac{\mu R_{L2}}{2R_a+R_{L1}+R_{L2}} \tag{4.23}$$

$$g_{22}=\frac{-\mu R_{L2}}{R_a+R_{L2}+(1+\mu)R_{K2}}\approx\frac{-\mu R_{L2}}{2R_a+R_{L1}+R_{L2}} \tag{4.24}$$

$$g_{21}=\frac{\mu R_{K2}}{R_a+R_{L2}+(1+\mu)R_{K2}}\cdot\frac{(1+\mu)R_{L1}}{R_a+R_{L1}}\approx\frac{\mu R_{L1}}{2R_a+R_{L1}+R_{L2}} \tag{4.25}$$

$$g_{K1}=\frac{\mu(R_a+R_{L2})R_{L1}}{(1+\mu)(R_K+R_B)(2R_a+R_{L1}+R_{L2})+(R_a+R_{L1})(R_a+R_{L2})}\approx\frac{\mu}{2(1+\mu)}\cdot\frac{R_{L1}}{R_K+R_B} \tag{4.26}$$

$$g_{K2}=\frac{\mu(R_a+R_{L1})R_{L2}}{(1+\mu)(R_K+R_B)(2R_a+R_{L1}+R_{L2})+(R_a+R_{L1})(R_a+R_{L2})}\approx\frac{\mu}{2(1+\mu)}\cdot\frac{R_{L2}}{R_K+R_B} \tag{4.27}$$

$$Z_{in1}=\frac{R_G}{1-\frac{\mu R_{K1}}{R_a+R_{L1}+(1+\mu)R_{K1}}}\approx 2R_G \tag{4.28}$$

$$Z_{in2}=\frac{R_G}{1-\frac{\mu R_{K2}}{R_a+R_{L2}+(1+\mu)R_{K2}}}\approx 2R_G \tag{4.29}$$

$$Z_{inK}=R_K+R_B+\frac{(R_a+R_{L1})(R_a+R_{L2})}{(1+\mu)(2R_a+R_{L1}+R_{L2})}\approx R_K+R_B \tag{4.30}$$

$$Z_{out1}=\frac{R_{L1}(R_a+(1+\mu)R_{K1})}{R_{L1}+R_a+(1+\mu)R_{K1}}\approx\frac{R_{L1}(2R_a+R_{L2})}{R_{L1}+2R_a+R_{L2}} \tag{4.31}$$

$$Z_{out2}=\frac{R_{L2}(R_a+(1+\mu)R_{K2})}{R_{L2}+R_a+(1+\mu)R_{K2}}\approx\frac{R_{L2}(2R_a+R_{L1})}{R_{L2}+2R_a+R_{L1}} \tag{4.32}$$

For better readability of the equations (4.35) and (4.36), we define the resistors R_p and R_t:

$$R_p=\frac{(R_a+R_{L1})(R_a+R_{L2})}{(2R_a+R_{L1}+R_{L2})} \tag{4.33}$$

$$R_t=R_B+(1+\mu)R_K \tag{4.34}$$

The sensitivities of the two outputs against variations of the power supply voltage are:

$$k_{\mathrm{SV1}}=\frac{\frac{R_{\mathrm{a}}}{R_{\mathrm{a}}+R_{\mathrm{L1}}}\cdot R_{\mathrm{p}}+R_{\mathrm{t}}}{R_{\mathrm{p}}+R_{\mathrm{t}}} \tag{4.35}$$

$$k_{\mathrm{SV2}}=\frac{\frac{R_{\mathrm{a}}}{R_{\mathrm{a}}+R_{\mathrm{L2}}}\cdot R_{\mathrm{p}}+R_{\mathrm{t}}}{R_{\mathrm{p}}+R_{\mathrm{t}}} \tag{4.36}$$

For equations (4.33) to (4.36), we assume that no current is flowing through the coupling capacitors. These equations therefore are only valid for DC and very low frequencies. For the other equations of the long tail pair, we assume that there is no voltage drop across the coupling capacitors. These equations consequently are valid in the audio frequency range only.

5. Large-signal models of the five basic tube stage circuits

In this chapter we expand the small signal models into the large-signal models of the five basic tube stage circuits.

Static nonlinear model

In a first step we approximate the static nonlinearities of the tubes with a generalized logistic function glf(x, k_{bias}, b, $type$, k_{loop}) or shortly GLF. In order to achieve a realistic saturation behavior, we need to embed this normalized GLF nonlinearity between a pre- and a post-gain stage. For the common cathode circuit, for example, we obtain:

$$k_{\mathrm{pre}}=\frac{-g_A}{i_{\mathrm{sat}}R_L}=\frac{\mu}{i_{\mathrm{sat}}\left(R_{\mathrm{L}}+R_{\mathrm{a}}+(1+\mu)R_{\mathrm{K}}\right)} \tag{5.1}$$

$$k_{\mathrm{post}}=-i_{\mathrm{sat}}R_{\mathrm{L}} \tag{5.2}$$

The saturation current i_{sat} is the anode current at saturation. It depends not only on the circuit of the tube stage of interest but on the source impedance of the previous stage too. A high source impedance results in a low saturation current, and vice versa. This relation is highly nonlinear and a result of the nonlinear grid current flowing at high grid voltages only. An upper boundary for i_{sat} is the current which would flow through a short between the anode and the cathode. The bias current i_{bias} is the anode current at the bias point; it may be calculated by dividing the bias voltages (often indicated in the circuit diagrams of tube amps) by the corresponding resistor values. The parameter k_{bias} of the GLF can be calculated now:

$$k_{\mathrm{bias}}=\frac{i_{\mathrm{bias}}}{i_{\mathrm{sat}}} \tag{5.3}$$

The parameters *type* and *b* offer two degrees of freedom which can be used to optimize fitting the GLF and the nonlinear transfer function of the real tube stage.

The parameter k_{loop} is the open-loop gain of the circuit. It determines the “hardness” of the GLF clipping. A small-signal analysis of the common cathode circuit results in:

$$k_{\text{loop}} = \frac{(1+\mu) R_{\text{K}}}{R_{\text{L}} + R_{\text{a}}} \tag{5.4}$$

Note that the GLF is zero for zero input. This means that the output voltage of this static nonlinear model emulates the real actual output voltage minus the real output voltage for zero input signal.

Dynamic effects

This simple model is already a good approximation of the static and nonlinear behavior of a real tube stage. However, there are some dynamic effects which need to be emulated as well:

a) **Limited rejection of fluctuations of the power supply voltage**: The sensitivity k_{SV} regarding how fluctuations in the power supply voltage are felt at the tube stage outputs is already known from the small signal model. A portion k_{SV} of the change in the power supply voltage $\mathrm{d}v_s$ is added to the output voltage of the tube stage. Since changes in the power supply voltage occur rather slowly i.e. they are fluctuations at rather low frequencies, they are barely audible as an added signal. The temporary bias-point shifting in the next tube stage (which will occur even if there is a DC-blocking capacitor between the stages) will still affect the nonlinear distortion and the gain of the next tube stages. This will be likely to result in audible effects.

b) **Blocking distortion:** A bias-point shift happens due to the change of the DC voltage across the DC blocking capacitor between two stages. The DC voltage changes because at high input voltages, a nonlinear grid current flows into the grid of the tube stage of interest. This effect can be so strong that the bias voltage itself drives the tube far into the negative saturation region where the tube has zero gain. It takes some time for the bias shift to recover even after the input signal itself does not overdrive the tube anymore. During this time the input signal is “blocked”.

c) **Sagging:** The power supply voltage often decreases or “sags” with increasing input signal level because the tube stages draw more supply current. This causes a reduction of the output voltage swing, of the gain, and of the supply current of the tube stages.

d) **Signal-dependent supply current**: The supply current drawn by a tube stage changes with changing input signal level. It often increases with increasing input level.

We will now look into the block diagrams of the five basic tube stage models which implement all of the above dynamic effects:

Common cathode – large-signal model

The block diagram of the common-cathode circuit is shown in **Fig. 18**. Let us identify the blocks already described for the static nonlinear model first:

The pre-gain stage is the first stage after the input.

The GLF block is shown in the middle of the diagram. It is an adnl-object implementing a nonlinear transfer function utilizing the antiderivative approach for antialiasing as described in chapter 2. It is configured as a GLF. Note that the parameters of an object are shown in the rounded rectangle below the processing block of the object.

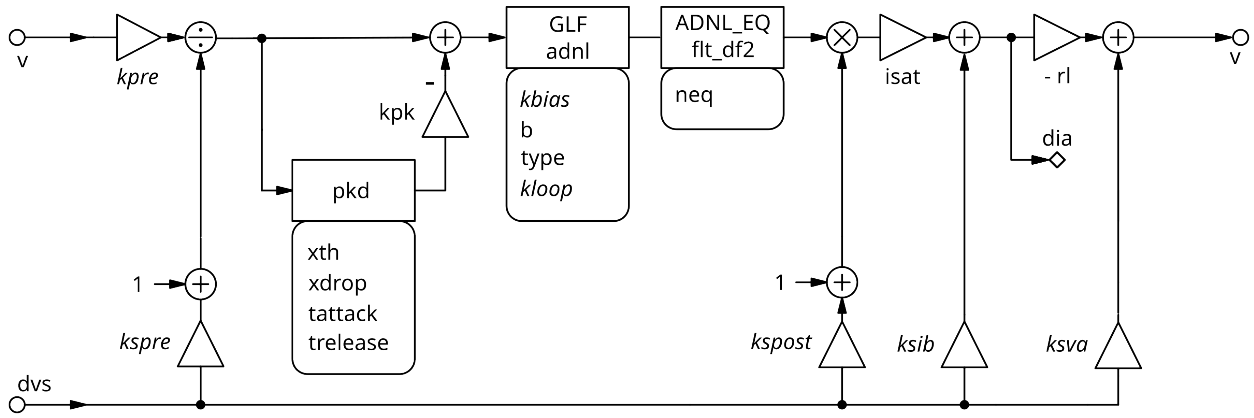


Fig. 18: Block diagram of the common-cathode circuit

Subsequent to the GLF block, the signal enters an flt-df2-object which is a 2nd-order IIR-filter in a direct form 2. It is configured to equalize the frequency response of the antiderivative approach (ADNL_EQ) as described in chapter 2. Its parameter n_{eq} sets the cutoff frequency of the equalizer in integer multiples of the sample rate divided by 12.

The post-gain stage is split into two parts. After the multiplication with i_{sat} and adding the bias-current changes (due to fluctuations in the power supply voltage) the signal d_{ia} emulates the actual anode current minus i_{bias}. This is used in the emulation of the power supply, as well. After multiplication with $-R_L$, the end of post-gain stage is reached.

We now get to the implementation of the dynamic effects:

The **limited rejection of fluctuations in the power supply voltage** is realized by multiplying the voltage difference dv_s (actual supply voltage minus supply voltage given no signal) with k_{SVA}, and adding the product to the output voltage of the stage. Note that dv_s is an additional input signal sourced from a power-supply emulator connected to all tube stages.

The **blocking distortion** effect is realized directly before the GLF block utilizing a coefficient k_{pk} and a pkd-object implementing a peak detector:

The DC-blocking capacitor of the previous stage and the nonlinear grid current of the actual stage indeed act as a positive peak detector. A virtual diode connected in parallel to the grid and cathode terminals of the tube can emulate the nonlinear current. The current only flows for input voltages above a certain threshold and charges the blocking capacitor. The peak detector is realized in the normalized signal domain after the pre-gain stage because its parameters are also normalized this way, and are more easily estimated.

The realistic range of the parameter k_{pk} is about 0.20 to 0.99. It determines the strength of the blocking-distortion effect. A low source impedance results in high parameter values and vice versa. If there is no DC-blocking capacitor ahead of the tube stage, k_{pk} must be set to zero in order to disable the blocking distortion effect.

The parameter x_{th} corresponds with the input voltage threshold. A normalized input value of 1 - k_{bias} corresponds to the input voltage at the positive saturation knee. Half of this value is a good starting point for x_{th}.

The parameter x_{drop} corresponds with the voltage drop across the virtual diode at very high input voltages. A good starting point is x_{th} times the reciprocal of the square root of k_{pk}. Given this relation, x_{drop} increases with increasing source impedance as is the case in the real circuits.

The parameters t_{attack} and $t_{release}$ are the attack and release time constants of the peak detector and can be estimated by the product of the DC-blocking capacitor value and that of the effective source and input resistors.

The peak detector object is implemented as a series connection of 3 blocks:

- A nonlinear transfer function is shown in **Fig. 19**. It emulates the voltage drop across a virtual diode caused by the diode current. It has three regions with the following normalized values:
  $x <= 0$ with: $y = 0$
  $0 < x < 2$ with: $y = 0.25\, x^2$
  $x >= 2$ with: $y = x - 1$
- A first order lowpass filter with the time constant t_{attack}.
- A peak-hold circuit with a release time constant $t_{release}$.

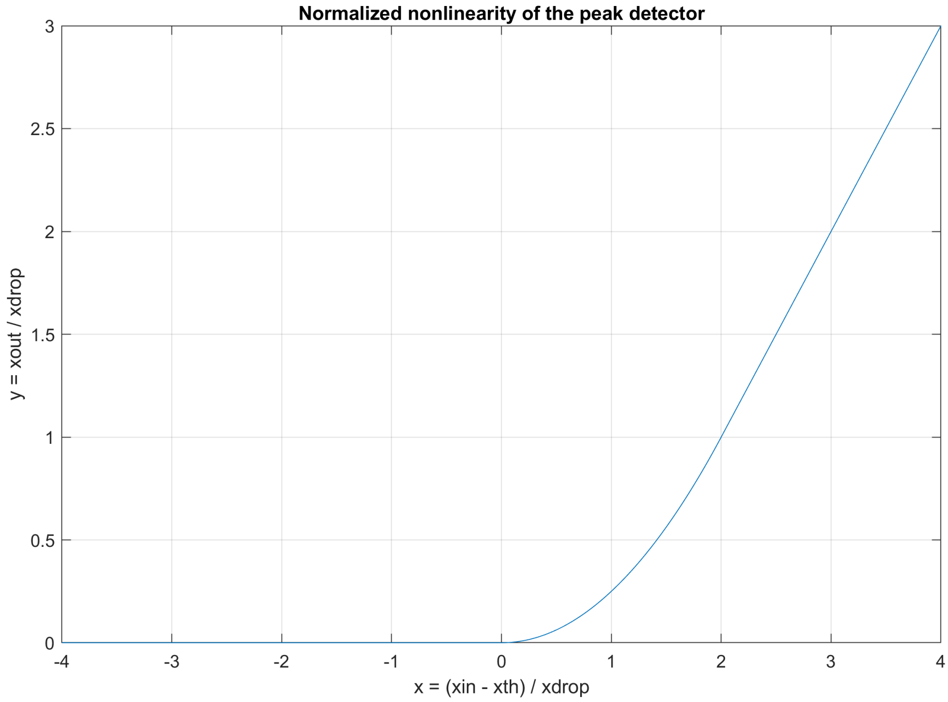


Fig. 19: The normalized nonlinear transfer function of the peak detector

The **sagging** effect is implemented with the help of the coefficients k_{Spre}, k_{Spost} and k_{Sib}:

$$k_{Spre} = \frac{1 - k_{comp}}{v_s} \tag{5.5}$$

$$k_{Spost} = \frac{1}{v_s} \tag{5.6}$$

$$k_{Sib} = \frac{i_{bias}}{v_s} \tag{5.7}$$

In equations (5.5) to (5.7), the parameter v_s is the supply voltage of the tube stage with no input signal. The parameter k_{comp} sets the amount of gain compression. A parameter value of zero results in no gain compression, while a value of one results in maximum gain compression. For triode tube stages k_{comp} should be set to values close to zero because a decreasing supply voltage merely results in lower saturation values while the gain does not change much. For pentode tube stages k_{comp} should be set to values close to one because a decreasing supply voltage also decreases the voltage at the secondary grid – which results in a gain reduction of the tube. The equations (5.5) to (5.7) are valid for all five tube-stage circuits.

The **signal-dependent supply current** minus i_{bias} is available after the multiplication with i_{sat}. Subsequent to adding the sagging-dependent component of the supply current change which is the product of dv_s and k_{Sib}, we obtain di_a which is the actual supply current minus i_{bias}. This signal is an output signal which is processed by a power supply emulator connected to all tube stages.

Some common-cathode circuits are biased by an additional offset voltage at the input rather than by the voltage drop across R_K . This may be emulated by simply setting R_K to zero. There is no need to add the offset voltage to the input because the model works with differences of voltage and current relative to the bias points.

Note that the input parameters which describe a tube-stage model are shown in regular letters in the tube-stage block diagrams. Parameters or coefficients that need to be calculated from one or more input parameters are shown in italics. The 16 input parameters of a common-cathode tube stage model are: μ, R_a, i_{sat}, i_{bias}, b, *type*, v_s, R_L, R_K, k_{comp}, k_{pk}, x_{th}, x_{drop}, t_{attack}, $t_{release}$ and n_{eq}.

Common-cathode fitted with a cathode capacitor - large signal model

The block diagram of the common-cathode circuit with cathode capacitor is shown in **Fig. 20**. In this circuit, the feedback of the anode current to the input voltage is implemented explicitly; this is in contrast to the common-cathode circuit where we used an implicit approach. The parallel circuit of R_K and C_K acts as a lowpass filter for the feedback signal in the real circuit. This lowpass filter is emulated with an flt_ii1-object. The latter implements a very efficient first-order IIR-filter utilizing the impulse-invariant design method. It is configured as a lowpass filter (LP). Its parameter t_{CK} is the time constant of the lowpass filter which is the product of R_K and C_K. After the multiplication with k_{fb} we obtain the feedback signal. In order to arrive at a delay-free feedback loop, we need to delay the signal by one sampling period with the Z^{-1} block. The feedback loop is closed by adding the feedback signal to the input signal. At high frequencies the feedback loop is open and only k_{SVA} from equation (4.9) determines the sensitivity to fluctuations of the power supply voltage. At low frequencies the feedback path is active. It is not only driven by the output voltage but by – d_{vs} as well. This is necessary to arrive at a correct sensitivity re. power supply voltage fluctuations at all frequencies.

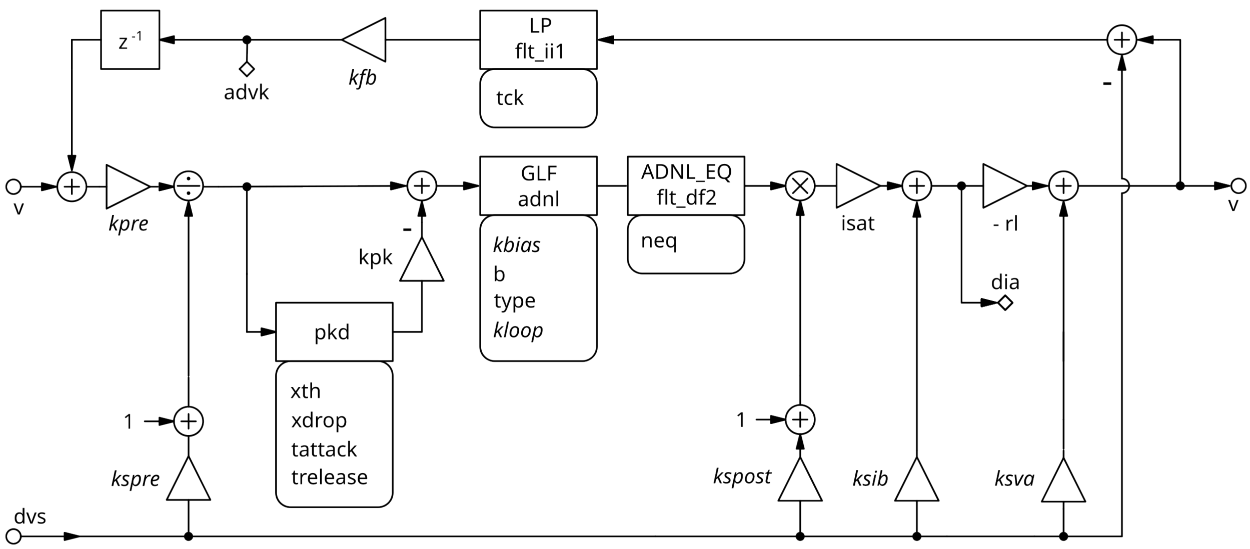


Fig. 20: Block diagram of the common-cathode circuit fitted with a cathode capacitor

The explicit feedback loop reduces the gain of the circuit at low frequencies. It is also shifting the bias point of the circuit if the average output voltage changes due to variations in the input signal level, or due to sagging. A “harder” clipping behavior at low frequencies is a consequence of the explicit feedback loop, as well.

Note: the explicit feedback solution used here would produce an unrealistic frequency response peak at high frequencies if t_{CK} is smaller than about 12 sampling periods! This is the reason why we use an implicit solution for the broadband feedback loops of the other circuits. Fortunately, the time constants of real tube stages fitted with a cathode capacitor are higher than 12 sampling periods for most guitar amplifiers.

The remaining parameters of the common-cathode circuit fitted with a cathode capacitor are:

$$k_{\text{pre}} = \frac{\mu}{i_{\text{sat}}(R_L + R_a)} \tag{5.8}$$

$$k_{\text{loop}} = 0 \tag{5.9}$$

$$k_{\text{fb}} = \frac{(1+\mu) R_K}{\mu R_L} \tag{5.10}$$

The difference voltage adv_K is the difference between the actual cathode voltage and the voltage drop caused by i_{bias}. This signal is also available outside the tube-stage object and may be used to emulate shared cathode resistors and capacitors as they are often used in double-triode arrangements such as with the 12AX7, or for pairs of power amp tubes. Replacing the adv_K values of both tubes with their mean value before the tube models are processed will do the job in this case.

The input-parameter set of the model is identical to the input-parameter set of the common-cathode tube stage model while t_{CK} is an additional parameter.

Cathode follower – large-signal model

The block diagram of the cathode-follower circuit is shown in **Fig. 21**. This block diagram is nearly identical to the block diagram of the common-cathode stage but implements a cathode output instead of an anode output.

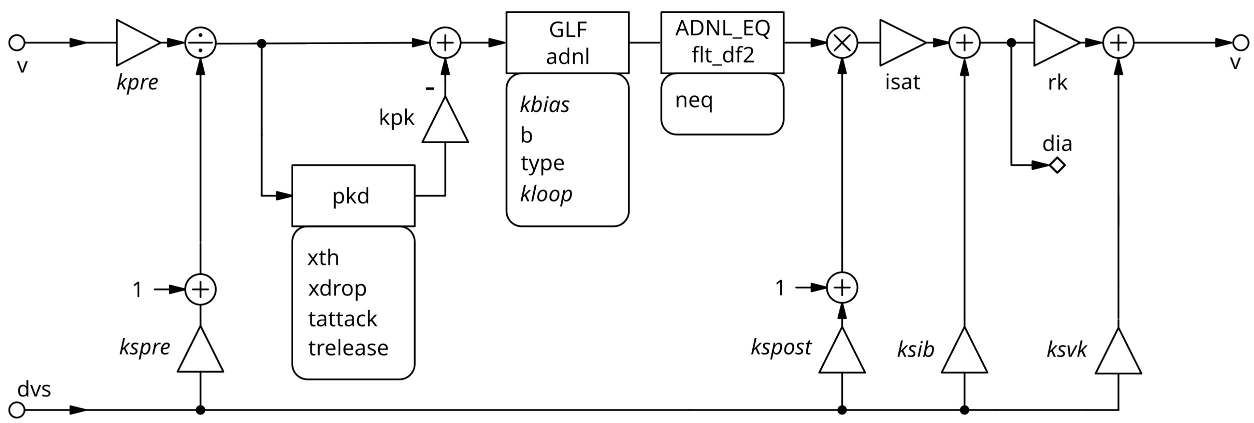


Fig. 21: Block diagram of the cathode-follower circuit

The remaining parameters of the cathode-follower circuit are:

$$k_{\mathrm{pre}} = \frac{-\mu}{i_{\mathrm{sat}}\left(R_{\mathrm{L}} + R_{\mathrm{a}} + (1+\mu) R_{\mathrm{K}}\right)} \tag{5.11}$$

$$k_{\mathrm{loop}} = \frac{(1+\mu) R_{\mathrm{K}}}{R_{\mathrm{L}} + R_{\mathrm{a}}} \tag{5.12}$$

The input-parameter set is identical to the input-parameter set of the common-cathode tube stage model. However, the parameter R_{L} is zero in most cases of use of this circuit.

Cathodyne – large-signal model

The block diagram of the cathodyne circuit is shown in **Fig.22**. This block diagram is nearly identical with the block diagram of the common-cathode stage but adds a cathode output signal v_{K}.

The remaining parameters of the cathodyne circuit are:

$$k_{\mathrm{pre}} = \frac{-\mu}{i_{\mathrm{sat}}\left(R_{\mathrm{L}} + R_{\mathrm{a}} + (1+\mu)(R_{\mathrm{K}} + R_{\mathrm{L}})\right)} \tag{5.13}$$

$$k_{\mathrm{loop}} = \frac{(1+\mu)(R_{\mathrm{K}} + R_{\mathrm{L}})}{R_{\mathrm{L}} + R_{\mathrm{a}}} \tag{5.14}$$

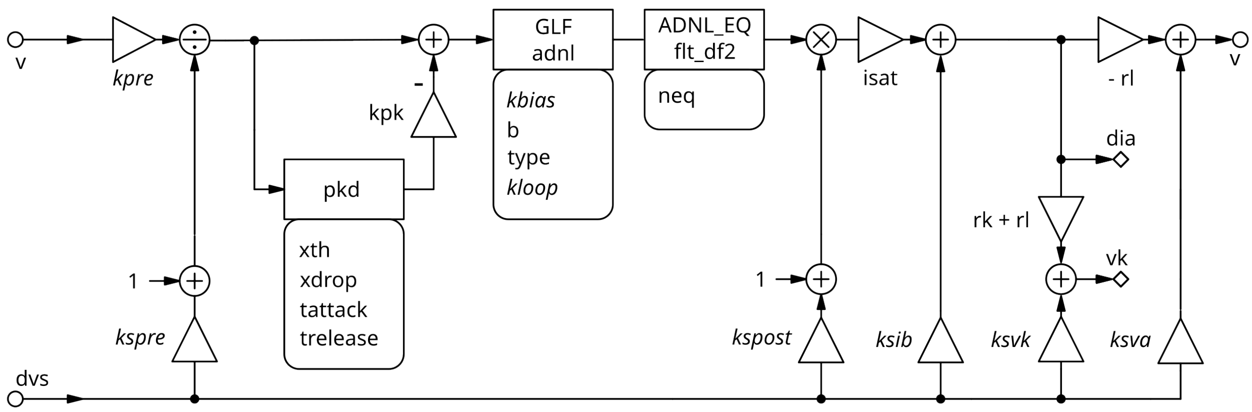


Fig. 22: Block diagram of the cathodyne circuit

The input-parameter set of the model is identical to the input-parameter set of the common-cathode tube stage model.

Long tail pair – large-signal model

The block diagram of the long-tail-pair circuit is shown in **Fig.23**. It is similar to a pair of common-cathode circuits with an additional cathode input.

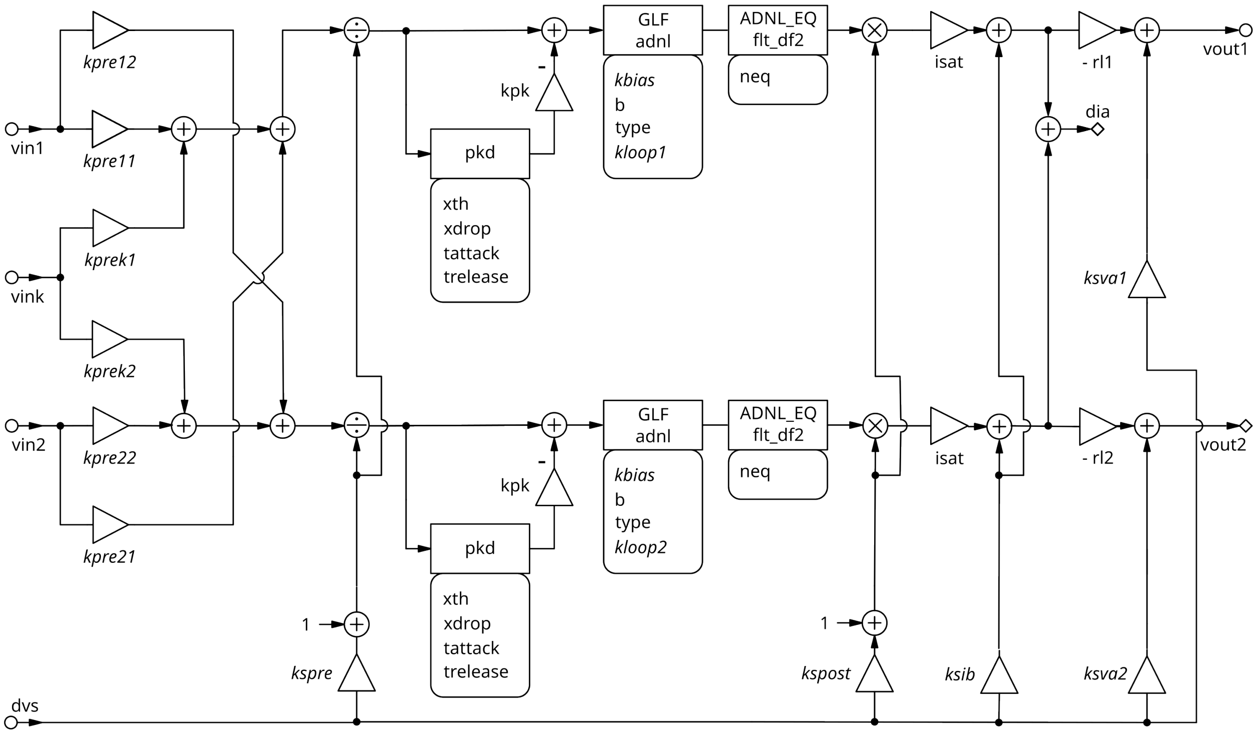


Fig. 23: Block diagram of the long-tail-pair circuit

With the resistors R_{K1} and R_{K2} already defined in equations (4.20) and (4.21), we can calculate the remaining parameters of the long tail pair circuit:

$$k_{pre11}=\frac{\mu}{i_{sat}\left(R_a+R_{L1}+(1+\mu)R_{K1}\right)}\approx\frac{\mu}{i_{sat}\left(2R_a+R_{L1}+R_{L2}\right)} \tag{5.15}$$

$$k_{pre12}=\frac{-\mu R_{K1}}{R_a+R_{L1}+(1+\mu)R_{K1}}\cdot\frac{(1+\mu)}{i_{sat}\left(R_a+R_{L2}\right)}\approx\frac{-\mu}{i_{sat}\left(2R_a+R_{L1}+R_{L2}\right)} \tag{5.16}$$

$$k_{pre22}=\frac{\mu}{i_{sat}\left(R_a+R_{L2}+(1+\mu)R_{K2}\right)}\approx\frac{\mu}{i_{sat}\left(2R_a+R_{L1}+R_{L2}\right)} \tag{5.17}$$

$$k_{pre21}=\frac{-\mu R_{K2}}{R_a+R_{L2}+(1+\mu)R_{K2}}\cdot\frac{(1+\mu)}{i_{sat}\left(R_a+R_{L1}\right)}\approx\frac{-\mu}{i_{sat}\left(2R_a+R_{L1}+R_{L2}\right)} \tag{5.18}$$

$$k_{preK1}=\frac{-\mu\left(R_a+R_{L2}\right)}{i_{sat}\left((1+\mu)\left(R_K+R_B\right)\left(2R_a+R_{L1}+R_{L2}\right)+\left(R_a+R_{L1}\right)\left(R_a+R_{L2}\right)\right)}\approx\frac{-\mu}{2(1+\mu)}\cdot\frac{1}{i_{sat}\left(R_K+R_B\right)} \tag{5.19}$$

$$k_{preK2}=\frac{-\mu\left(R_a+R_{L1}\right)}{i_{sat}\left((1+\mu)\left(R_K+R_B\right)\left(2R_a+R_{L1}+R_{L2}\right)+\left(R_a+R_{L1}\right)\left(R_a+R_{L2}\right)\right)}\approx\frac{-\mu}{2(1+\mu)}\cdot\frac{1}{i_{sat}\left(R_K+R_B\right)} \tag{5.20}$$

$$k_{\text{loop1}} = \frac{(1+\mu) R_{\text{K1}}}{R_{\text{a}} + R_{\text{L1}}} \approx \frac{R_{\text{a}} + R_{\text{L2}}}{R_{\text{a}} + R_{\text{L1}}} \quad (5.21)$$

$$k_{\text{loop2}} = \frac{(1+\mu) R_{\text{K2}}}{R_{\text{a}} + R_{\text{L2}}} \approx \frac{R_{\text{a}} + R_{\text{L1}}}{R_{\text{a}} + R_{\text{L2}}} \quad (5.22)$$

The input-parameter set of the model is mostly identical to the parameter set of the common-cathode tube-stage model. Instead of R_{L} we have R_{L1}. R_{L2} and R_{B} are new parameters.

6. Model of the power supply stage

For the already mentioned power supply emulator we need a model of a single power supply stage. The model emulates the RC-lowpass of a single filter stage in a power supply chain. An input voltage feeds the signal to a series resistor *R*. Subsequent to the resistor, a capacitor *C* is connected to ground. The voltage following the resistor supplies one or more tube stages, and also the next filter stage. The block diagram of this model is shown in **Fig. 24**.

We are only interested in differences between actual voltages in operation and the voltages which occur with no input signal. The same is true for the currents. We therefore only use differences in voltages and currents rather than the absolute values.

The input signal $\mathrm{d}i_{\text{a}}$ is the sum of the currents drawn by all tubes supplied by the power supply stage. The input signal s_{next} is the current drawn by the respective next power supply stage. The sum of $\mathrm{d}i_{\text{a}}$ and s_{next} passes the lowpass filter LP1 which is a first-order lowpass filter. The time constant *tau* of the lowpass filter is the product of *R* and *C*. The output signal of LP1 is the current *s* which is the total current drawn from the power supply stage. It is used to calculate the voltage drop across *R* and is send to the s_{next} input of the previous power supply stage.

The voltage input of the power supply stage is $\mathrm{d}v_{\text{s}}$. It passes through the lowpass filter LP2 which is a clone of LP1. After the subtraction of the voltage drop across *R* we obtain the output voltage of the power supply stage. The voltage drop is caused by the current *s* which is already described in the previous paragraph.

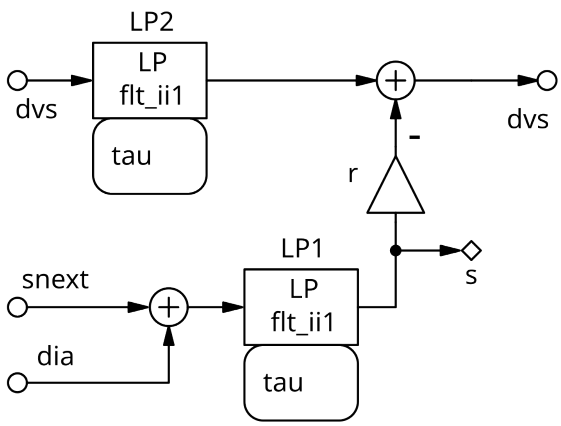


Fig. 24: Model of a power supply stage

Following are the rules for putting together a power supply chain: For the first stage the input voltage is zero. For the last stage s_{next} is zero. The voltage output of the previous stage is connected to the voltage input of the actual stage. The current output s of the next stage is connected to the s_{next} input of the actual stage.

The first stage of real circuits does not show a resistor but the diode pair of the rectifier. The value of resistor R of the first stage must be set to the value of the estimated differential resistor of the rectifier.

In the second stage, often a choke is used instead of a resistor. In this case the resistor R can be estimated as the DC-resistance of the choke. In order to emulate the additional filter effect of the inductance of the choke, we can use higher values for C compared to the capacitor value shown in the circuit diagram.

We could feed a periodic saw-tooth-signal with a frequency of 100 or 120 Hz into the voltage input of the first power supply stage. This would emulate the effects caused by the ripple in the power-supply voltage: a hum signal even present with no guitar signal, and a periodic modulation of the guitar signal and the distortion products. However, the author does not desire to reproduce these mainly annoying effects of real amps, and therefore did not implement any power-supply voltage ripple in the present model.

7. Model of the Tweed Deluxe amp

We are now ready to construct a model of a complete tube amp. Our objective is to model the 5E3 circuit of the Fender Tweed Deluxe amp. The 5E3 diagram shown in **Fig. 25** is copied from the documentation of the “Tweed One-Twelve-16 Amp Kit” from Tube Amp Doctor [3]. The original hand drawn 5E3 diagram from Fender is not shown here because it is hard to read. A lot of thanks to Tube Amp Doctor for the very clean redrawing.

The derived block diagram of the audio-signal path of the amp model is shown in **Fig. 26**:

Object parameters are only shown if they can be influenced by the GUI of the plugin. The objects that show such parameters implement a coefficient smoothing to prevent audible transients caused by a sudden parameter change. The object parameters not shown in the block diagram, and the coefficients k_1 and k_2 may only be changed in the source code of the plugin.

The first block is the gain stage G1 which is not a part of the 5E3 circuit. Its main purpose is to match the sensitivity of the audio interface to the sensitivity of the real amp. It can also be used to compensate different output levels of guitars, or just to get additional (or less) input gain. The associated GUI parameter $p.g_{in}$ can be set between -12 dB and +12 dB. Its default value is 0 dB. The relation between g_{in} and $p.g_{in}$ is:

$$g_{in} = \sqrt{1.2} \cdot 10^{\left(0.05\left(p.g_{in}+12\right)\right)} \tag{7.1}$$

The full-scale input level L_{FS} of an audio interface is specified in its data sheet using the unit dBu. A realistic input sensitivity of the simulated amp requires $p.g_{in} = L_{FS}$ - 12 dBu. In this case the voltage levels at the input of the audio interface and the virtual amp are the same.

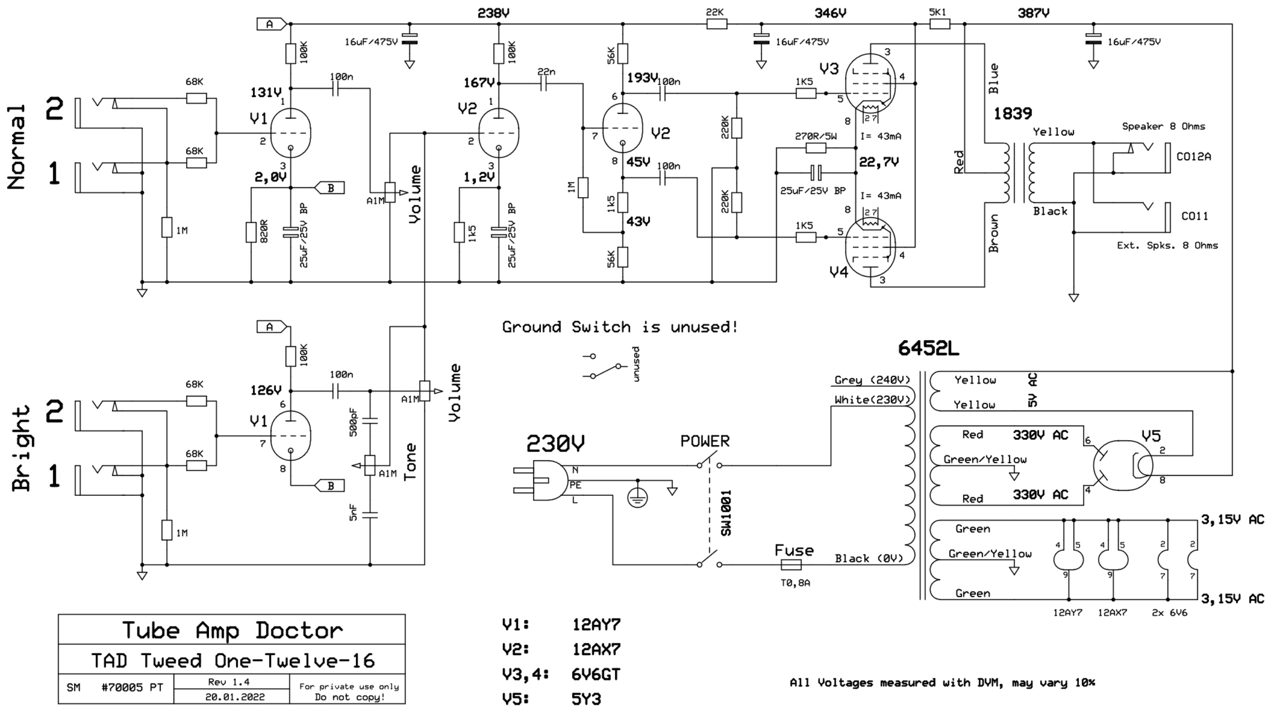


Fig. 25: Circuit diagram of the "Tweed One-Twelve-16 Amp Kit" from Tube Amp Doctor

The next block is the input tube of the 5E3 circuit. The real Tweed Deluxe has two inputs designated "Normal" and "Bright", respectively. We may use just one of the two inputs, or tie them together using a jumper cable. An analysis of the three possible connections showed that they can result in very different sounds if we stick to the original mixing circuit and tone-control following the two input tubes. If we, however, use just a single input tube followed by a single volume control plus a more versatile tone stack, we can obtain all of the sounds possible with the original amp design plus many more useful additional sounds. In fact, the behavior of the original circuit is very unpredictable because the two input tubes, the two volume controls, and the single tone control interact in a very complicated way. It makes no sense to emulate this behavior if we can get the same (and even more) sounds in a much more predictable way. A first step using this approach is to feed only the "Bright"-channel with the guitar signal and set the volume control of the "Normal"-channel to zero. We can emulate this setting via the first tube stage T1. The tube stage T1 emulates a common-cathode circuit with cathode capacitor. The emulated tube is a 12AY7 double triode. The cathodes of the two input tubes share the same 820 Ω resistor and 25 µF capacitor. This results in R_K= 820 Ω and t_{CK} = 20.5 ms. Since we do not feed a signal to the second input tube, its cathode current will not change. Therefore the second tube does not influence the first tube and there is no need to implement cathode coupling as shown in **Fig. 27** for the T4 and T5 tube stages.

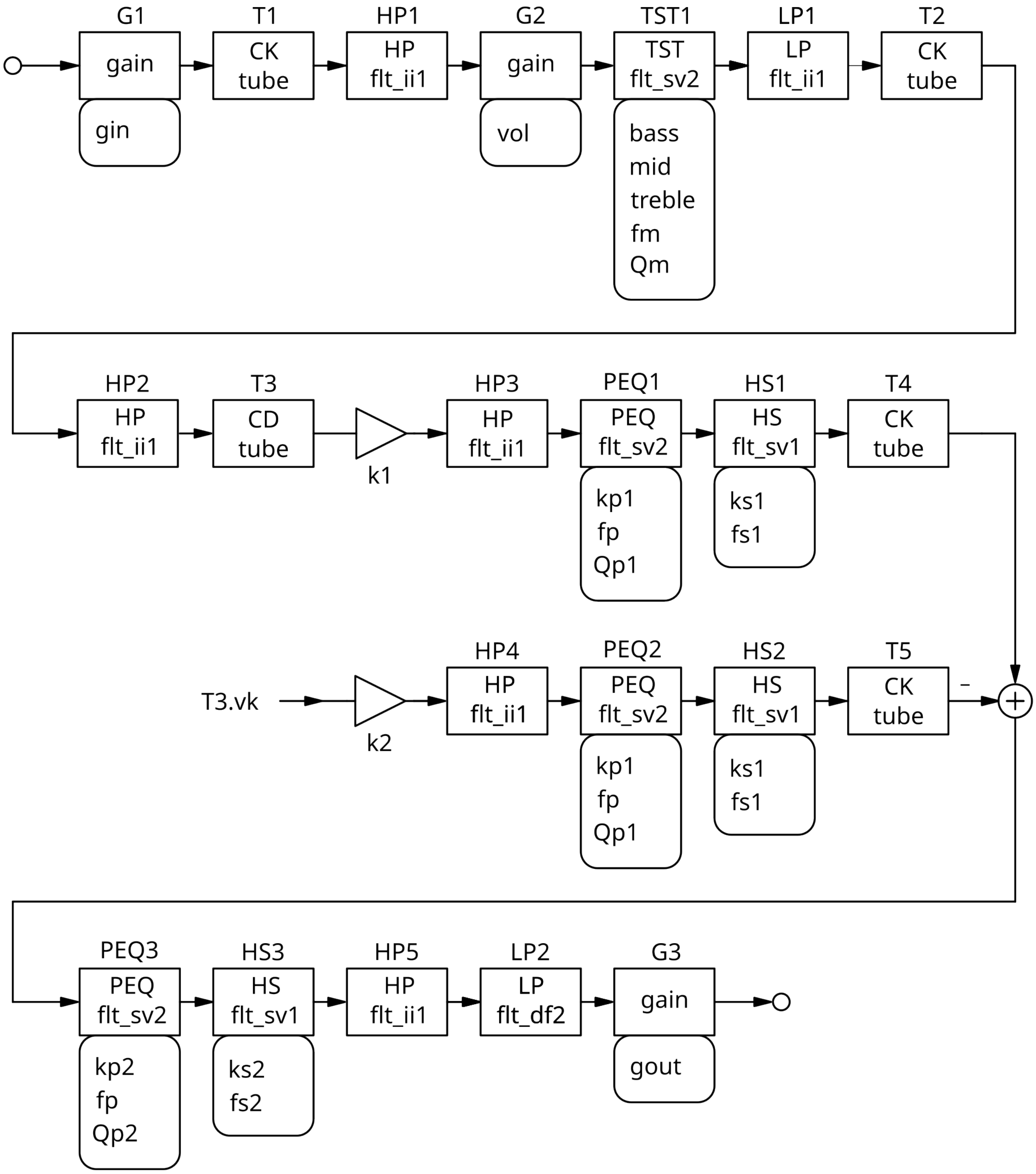


Fig. 26: Model of the 5E3 signal path

We could also emulate jumpered inputs and outputs with a single tube stage. The tube outputs would be jumpered in the 5E3 circuit if both volume controls would be set to 100 %. We simply need to choose R_K= 1.64 kΩ for a realistic emulation of this kind of connection. The result would be lower values of f_1 and f_2, and a stronger signal-dependent bias-point shifting. The output impedance is half that of a single tube. The gain for frequencies much higher than f_2 is the same as for a single tube. We do not offer this option in the GUI of the plugin because k_{bias} of T1 is very close to 0.5. This means that we do not expect a lot of bias point shifting at all, and thus not a big difference in sound between the two connection versions. Those interested in the corresponding small sound differences can easily change the R_K-value of T1 in the source code of the plugin.

The parameter k_{peak} is set to zero for the first tube stage. This disables the peak detector circuit of T1 because we do not assume a DC-blocking capacitor at the input.

The same n_{eq} is applied for all tube models. We select n_{eq} = 2 for sample rates lower than 88.2 kHz, n_{eq} = 0 for sample rates equal or higher than 176.4 kHz, and n_{eq} = 1 for other sample rates. This guarantees a flat frequency response up to a signal frequency of 7.35 kHz for all samples rates higher or equal to 44.1kHz. It also prevents “over equalizing” at higher signal frequencies. Sample rates below 44.1 kHz are not recommended because aliasing might get annoying.

The values of all parameters of all tube stages, and the values of k_1 and k_2, are shown in **Table 1**. Some of them can be directly derived from the given circuit diagram or from tube data sheets. The remaining parameters can at the very least be roughly estimated. Since the author does not own a Tweed Deluxe amp, he fine-tuned these latter parameters based on his own expectations as to sound and feel, comparing the results to recordings for which a Tweed Deluxe has probably been used.

The next block is the first-order highpass filter HP1. Its cutoff frequency is determined by the 100 nF-DC-blocking capacitor, and the effective input impedance of the following volume- and tone-control circuit. In the real circuit its value varies with the control settings. For maximum volume in the “Bright”-channel we see a 500 kΩ load, resulting in a cutoff frequency of 3.2 Hz. For lower volume settings the load resistor can be much lower. We therefore use a value of 10 Hz for the cutoff frequency of HP1.

The next block is the volume control G2. The GUI parameter *p.vol* can be set between 0 and 100 %; the default value is of 50 %. The relation between *vol* and *p.vol* approximates an audio-taper or “logarithmic” potentiometer with 12 dB attenuation in the middle position; it is:

$$vol = \left(\frac{p.vol}{100} \right)^2 \tag{7.2}$$

We now get to the tone stack TST1 as the next block. It is a second order state variable filter configured as a universal tone stack. The three parameters *bass*, *mid* and *treble* determine the gain of the three frequency bands. If all three parameters have the same value, the frequency response of the tone stack is flat. This is true for every value! The three-band filter arrangement uses second order filters for lowpass, bandpass, and highpass. All three filters have the same pole frequency f_{mid} and quality factor q_{mid}. Typical values for the emulation of real tone stacks are: f_{mid} = 630 Hz and q_{mid} = 0.355. These are also the default values of the two parameters. The three GUI parameter *p.bass*, *p.mid* and *p.treble* have the same range, default values, and scaling as the GUI parameter *p.vol*. The GUI parameters $p.f_{mid}$ and $p\ q_{mid}$ cover a range from -12 to +12 dB starting from their default values but are displayed as linear values in the GUI. With this range of parameters, every passive RC tone stack of real tube amps can be emulated. However, we can even enter the territory of active or LC tone stacks if we set *q.mid* > 0.5. In contrast to the situation in real tube amps, the frequency response of this tone stack is easily predictable from the tone stack settings. A detailed description of the state variable filter used for this universal tone stack is available in [4] and [5].

Some frequency response plots of the universal tone stack can be found in chapter 10. There you will find some hints to emulate the original 5E3 tone control settings as well.

The next block is the first-order lowpass filter LP1. Its cutoff frequency is determined by the input capacitance of T2 and the source impedance of the previous stage. The input capacitance of T2 is about 120 pF. Such high values are typical for high-gain tube stages. The effective source impedance value varies with the volume and tone-control settings and will be between 17.6 kΩ and 333.3 kΩ. We assume a typical value of 150 kΩ resulting in a cutoff frequency of 8.8 kHz. We will not insert lowpass filters between the following tube stages because the source impedances and input capacitances are much lower there. This results in much higher cutoff frequencies and negligible attenuation at audio frequencies.

The following block T2 is again a common cathode circuit with a cathode capacitor and a 12AX7 triode.

As the next block, we arrive at the first-order highpass filter HP2. Its cutoff frequency is determined by the 22-nF-DC-blocking capacitor and the effective input impedance of the following cathodyne tube stage. The grid resistor of the cathodyne circuit is 1 MΩ resulting in an input impedance of 17.6 MΩ according to equation (4.15). This results in a cutoff frequency of 0.41 Hz.

Following is the block with tube stage T3. It is a cathodyne circuit deploying a 12AX7 triode.

The two outputs of the cathodyne circuit pass 100-nF-DC-blocking capacitors, and then are subject to the load due to the 220-kΩ resistors. After passing the 1.5-kΩ series resistors, the signals at last reach the grids of the power-amp tubes.

The next block in the anode path is a static gain block with gain k_1. It emulates the attenuation of the anode output due to the 220-kΩ load. On the cathode path we have k_2 for the same purpose.

The next blocks are the first-order highpass filters HP3 and HP4. Their cutoff frequencies are 5.8 Hz and 6.4 Hz.

The power amp tubes of the 5E3 circuit have a very high output impedance compared to the frequency dependent impedance of the loudspeaker. This is especially true for the small-signal range. If the power tubes are driven into saturation the output impedance becomes much smaller. However, it is still not small enough to prevent the occurrence of a significant effect in the frequency response due to the frequency response of the loudspeaker impedance. We can emulate these frequency-dependent effects via two filter stages. The second stage acts subsequent to the power amp tubes; it emulates the influence in the frequency response due to the output impedance at saturation. The first stage is located ahead of the power amp tubes. The product of the transfer functions of the first and second stage emulates the frequency response given due to the small-signal output impedance of the power amp tubes.

We will now address developing the appropriate filter stages:

We use equation (7.3) as a good approximation of a real loudspeaker impedance $Z_s(f)$:

$$Z_s(f)=R_s\left(\frac{1+j\frac{f}{Q_{ts}f_{res}}-\frac{f^2}{f_{res}^2}}{1+j\frac{f}{Q_{ms}f_{res}}-\frac{f^2}{f_{res}^2}}+j\frac{f}{f_{ind}}\right) \tag{7.3}$$

The relevant Thiele-Small parameters of the loudspeaker are: resonance frequency f_{res}, total quality factor Q_{ts}, and mechanical quality factor Q_{ms}. At the frequency f_{ind}, the magnitude of the impedance of the loudspeaker inductance L_s equals its DC-resistance R_s:

$$f_{ind}=\frac{R_s}{2\pi L_s} \tag{7.4}$$

$Z_s(f)$ puts a load onto a voltage source with an output impedance R_x, and we obtain $H_x(f)$ as the transfer function from the source to the load voltage:

$$H_x(f)=\frac{Z_s(f)}{R_x+Z_s(f)}\approx\frac{R_s}{R_x+R_s}\cdot\frac{1+j\frac{f}{Q_{ts}f_{res}}-\frac{f^2}{f_{res}^2}}{1+j\frac{f}{Q_x f_{res}}-\frac{f^2}{f_{res}^2}}\cdot\frac{1+j\frac{f}{f_{ind}}}{1+j\frac{R_s}{R_x+R_s}\frac{f}{f_{ind}}} \tag{7.5}$$

The quality factor $Q_{x:}$ is defined in equation (7.6):

$$Q_x=\frac{R_x+R_s}{\frac{R_x}{Q_{ms}}+\frac{R_s}{Q_{ts}}} \tag{7.6}$$

The approximation of $H_x(f)$ in equation (7.5) is the product of three transfer functions: a gain stage, a second-order peak-equalizer filter, and a first-order high-shelf filter. The gain stage emulates the attenuation that would occur if the loudspeaker impedance would be equal to R_s at all frequencies. The emulation of the power-amp tube stages T4 and T5 already includes this gain stage. The peak-equalizer filter emulates the peak appearing due to the peaking of the loudspeaker impedance at the resonance frequency of the speaker. The high-shelf filter emulates the additional gain occurring at high frequencies due to the loudspeaker inductance.

Typical loudspeakers found in Tweed Deluxe amps are P12 and C12 speakers sourced from Jensen. The typical data sheet values of Jensen loudspeakers for the “Vintage Alnico” and “Vintage Ceramic” series are: R_s = 6 Ω, f_s = 90 Hz, Q_{ts}= 2, Q_{ms} = 12.5, and f_{ind} = 1.4 kHz.

In the figure 10.5.20 of [6] the measured output impedance of a Tweed Deluxe amp is shown. From this figure we can estimate that R_x is about 80 Ω for small signals, this yielding a treble gain of 23 dB from the high shelf filter, and resonance gain of 13 dB from the peak-equalizer filter.

Chapter 10.5.12 of [6] gives the measured frequency responses of three tube power amp stages driven into saturation and loaded with typical guitar loudspeakers. We can generalize this measurement results by assuming that a treble gain of about 3 dB, and a resonance gain of about 2.5 dB would be typical values for tube power amps driven into saturation and loaded with typical guitar loudspeakers. This corresponds to an effective R_x of about 2.5 Ω for the “saturated” 5E3 power amp.

The gain values for the saturated scenario directly apply to the second filter stage implemented via PEQ3 and HS3. The gain values of the first filter stage are the gain values for small signals minus the gain values for saturation. The first filter stage is implemented as two identical filter stages ahead of each of the two power tubes, respectively. Ahead of T4, we have PEQ1 and HS1, and ahead of T5 we have PEQ2 and HS2.

The influence of the loudspeaker load on the perceived sound is very strong, which needs to be taken into account given that in real life Tweed Deluxe amps are often (with very attractive results) fitted with replacement speakers of different types and makes. Also, the resonance gain of a realistic model can be of a rather annoying effect. For these reasons all relevant load parameters for the loudspeaker are available in the GUI of the plugin.

For the “Speaker Resonance” (or the peak-equalizer filter) we have the following GUI parameters:

- Gain 1 = $p.g_{p\_pre}$: 0 to 24 dB, default value is 3 dB.
- Gain 2 = $p.g_{p\_post}$: 0 to 24 dB, default value is 1 dB.
- Fres = $p.f_p$: -12 dB to +12 dB, around a default value of 80 Hz
- Qts = $p.q_p$: -12 dB to +12 dB, around a default value of 2.0

For the “Speaker Inductance” (or the high-shelf filter), the following GUI parameters are available:

- Gain 1 = $p.g_{s\_pre}$: 0 to 24 dB, default value is 20 dB.
- Gain 2 = $p.g_{s\_post}$: 0 to 24 dB, default value is 3 dB.
- Find = $p.f_s$: -12 dB to +12 dB, around a default value of 1.25 kHz

The default values of the GUI parameters are not the most realistic values but they are recommended as starting points for dialing in an attractive sound. The relations between the GUI parameters and the filter stage parameters are:

$$k_{p1}=10^{(0.05\,p.g_{p\_pre})} \tag{7.7}$$

$$k_{p2}=10^{(0.05\,p.g_{p\_post})} \tag{7.8}$$

$$f_p=p.f_p \tag{7.9}$$

$$q_{p2}=p.q_p\sqrt{k_{p2}} \tag{7.10}$$

$$q_{p1}=q_{p2}\sqrt{k_{p2}k_{p1}} \tag{7.11}$$

$$k_{s1}=10^{(0.05\,p.g_{s\_pre})} \tag{7.12}$$

$$k_{s2}=10^{(0.05\,p.g_{s\_post})} \tag{7.13}$$

$$f_{s2}=p.f_s\sqrt{k_{s2}} \tag{7.14}$$

$$f_{s1} = f_{s2} \sqrt{k_{s2} k_{s1}} \tag{7.15}$$

A description of the peak-equalizer and high-shelf filters, and the definitions of their parameters can be found in [4].

The blocks T4 and T5 emulate 6V6GT tubes in a common-cathode circuit fitted with a cathode capacitor. The additional signal processing necessary to emulate the cathode coupling is shown in **Fig. 27**.

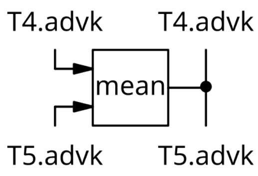


Fig. 27: Model of the 5e3 cathode coupling

The output voltages of T4 and T5 are combined by calculating their difference.

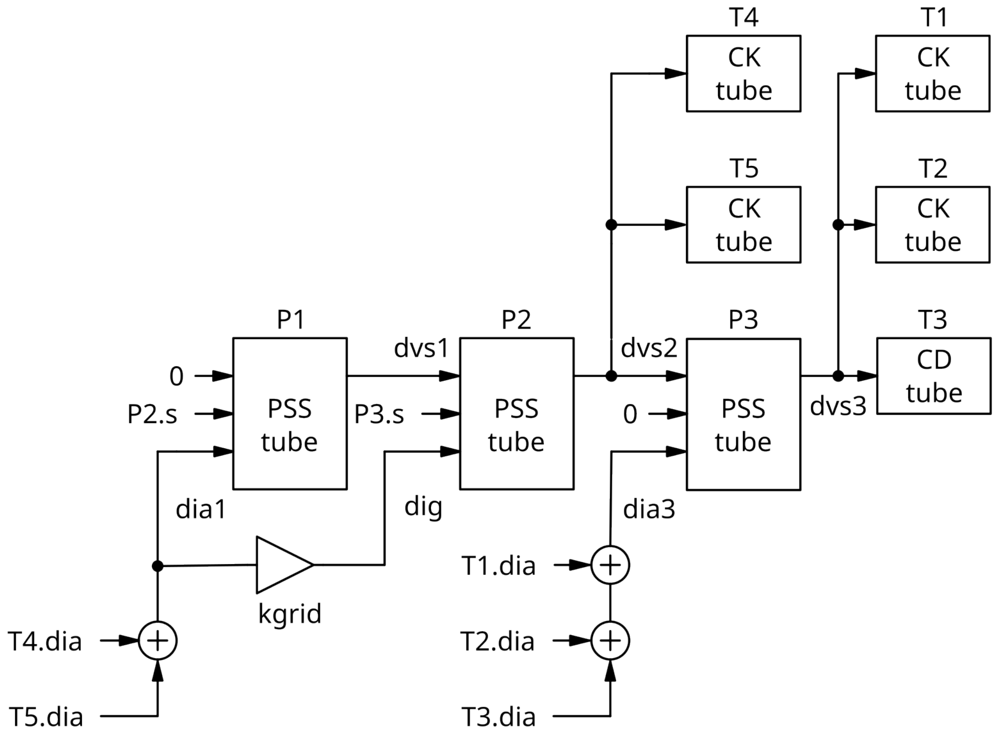
The output transformer in the 5E3 circuit translates a load of 8 Ω at the loudspeaker terminals to a load of 8 kΩ between the two anodes of the power-amp tubes. The first-order highpass filter HP5 emulates the low-end cutoff caused by the inductance of the transformer. A typical cut off frequency for tube-powered guitar amps is 40 Hz. The second-order lowpass filter LP2 emulates the high-end cutoff caused by the stray inductance and the stray capacitances of the transformer. A typical cutoff frequency for tube-based guitar amps is 10 kHz, and a quality factor of 0.707 insures a flat frequency response without a resonance peak.

The last block in the signal path of the 5E3 is the output gain block G3. Ahead of G3, the signal represents the difference voltage between the anodes of the power amp tubes. It is already filtered as the output transformer would filter it, but it is not transformed down to the loudspeaker voltage level. We use G3 to convert this voltage into normalized values. The normalization converts the nominal output voltage at saturation to a normalized value of 0.25 if the GUI parameter $p.g_{\mathrm{out}}$ is set to 0 dB. This parameter has a range of -12 dB to +12 dB, with its default value at 0 dB. It can be used to adjust the output level as desired by the user.

The relation between g_{out} and $p.g_{\mathrm{out}}$ is:

$$g_{\mathrm{in}} = \frac{0.5}{\mathrm{T4.}i_{\mathrm{sat}} \cdot \mathrm{T4.}R_{\mathrm{L}} + \mathrm{T5.}i_{\mathrm{sat}} \cdot \mathrm{T5.}R_{\mathrm{L}}} \cdot 10^{(0.05\, p \,.\, g_{\mathrm{in}})} \tag{7.16}$$

The model of the 5E3 power supply is shown in **Fig. 28**. There are three stages in the power supply: P1, P2 and P3.



.

Fig. 28: Model of the 5E3 power supply

For P1 we estimate R = 500 Ω. With C = 16 µF from the circuit diagram, we obtain tau = 8 ms.

For P2 we have R =5.1 kΩ and C = 16 µF in the circuit diagram, resulting in tau = 81.6 ms.

For P3 we have R =22 kΩ and C = 16 µF in the circuit diagram, resulting in tau = 352 ms.

The current into the secondary grid di_g is calculated as the product of di_{a1}.and k_{grid}. We estimate k_{grid} = 0.1.

The tube stages T4 and T5 are supplied with the difference voltage dv_{s2}. This is done because we assume that the grid supply is dominating the sagging effects.

A summary of the parameters and some derived properties of all tube stages is given in **Table 1**. The input voltages at the saturation knees are v_{inmax} and v_{inmin}. The output voltages in saturation-mode are v_{outmax} and v_{outmin}. The normalized input values of the tube stage GLF for the case of a saturated previous tube stage are x_{max} and x_{min}. The maximum value of the peak detector multiplied with k_{pk} is x_{pkmax}. This value determines the maximum strength of the blocking-distortion effect. The overdrive factors are k_{ODmax} and k_{ODmin}. They determine the maximum degree of distortion of the tube stage and are defined by the ratios x_{max} / (1 - k_{bias}) and - x_{min} / k_{bias}.

The anode-to-anode output voltage v_{aa} is the difference of the output voltages of T4 and T5. Its v_{outmax} and v_{outmin} values are defined by the saturation of the power amp tubes for the case of a flat loudspeaker impedance ($p.g_{p\_post}$ = 0 and $p.g_{s\_post}$ = 0). All values in **Table 1** imply SI units.

Table 1: Parameters and derived properties of all tube stages of the “CD 5E3”- variant

	T1	**T2**	**T3**	**k1**	**T4**	**T3.vk**	**k2**	**T5**	**Vaa**
Circuit	CK	CK	CD		CK	CD		CK	
µ	44	100	100		125			125	
Ra	25,000	62,500	62,500		40,000			40,000	
isat	0.00220	0.00155	0.00160		0.11000			0.12000	
ibias	0.00120	0.00076	0.00073		0.04200			0.04200	
b	0	0	0		2			2.5	
type	0.5	0.5	0.5		0.5			0.5	
vs	238	238	238		346			346	
RL	100,000	100,000	56,000		3,000			3,000	
RL2									
RK	820	1500	1500		540			540	
RB									
kcomp	0	0	0		1			1	
kpk	0	0.2	0.5		0.5			0.7	
xth	*0.250*	0.255	0.272		0.309			0.325	
xdrop	*0.25*	0.570	0.384		0.437			0.388	
tattack	*0.01*	0.015	0.00085		0.00575			0.00155	
trelease	*0.05*	0.05	0.3872		0.0276			0.0234	
tck	0.0205	0.0375			0.00675			0.00675	
gain	-35.200	-61.538	-0.945	0.797	-8.721	0.970	0.940	-8.721	
Zout	20,000	38,462	56,000		2,791	1,150		2,791	
vinmax	2.84	1.28	51.56		23.39			26.83	
vinmin	-3.41	-1.24	-43.26		-14.45			-14.45	
voutmax	120.0	76.0	40.9	32.6	126.0	50.0	47.0	126.0	360.0
voutmin	-100.0	-79.0	-48.7	-38.8	-204.0	-42.0	-39.5	-234.0	-330.0
1 - kbias	0.455	0.510	0.544		0.618			0.650	
- kbias	-0.545	-0.490	-0.456		-0.382			-0.350	
xmax		47.64	0.80		0.86			1.21	
xmin		-39.70	-0.83		-1.03			-1.02	
xpkmax		9.364	0.091		0.087			0.349	
kODmax		93.5	1.5		1.4			1.9	
kODmin		81.0	1.8		2.7			2.9	

It is worth mentioning that T1 is very difficult to push into overdrive without an additional input booster. T2 can be overdriven to an extreme degree if the volume control and the tone stack are set for minimum attenuation. The phase splitter T3 can only be slightly overdriven. However, the clipping of T3 is of a very “hard” character due to the strong local feedback. The power amp tubes can be overdriven a bit more in comparison with T3. However, due to the lack of local feedback, the clipping of a much softer kind. When increasing the volume control, distortion occurs first in T4 and T5, and then in T3. The distortion in T2 comes in last but will then quickly dominate the overall distortion level.

T4 and T5 have very different parameters due to the different source impedances provided by T3. For that reason, the power amp operates very asymmetrically, producing a lot of even-order harmonics. On the other hand, typical power amps driven by a long tail phase inverter show “symmetric” behavior and would suppress even-order harmonics to a high degree.

8. Variants of the 5E3 circuit

Above, we modeling a 5E3 circuit that is – safe for the expanded tone control circuit – true to the original “tweed” circuit. We will now look into some further modified variants of this circuit. The first mod deals with the unbalanced output impedance of the cathodyne stage. It is easy to balance the output impedance perfectly by inserting a series resistor of 54.85 kΩ at the cathode output in the “CD BAL”-variant. Even for the “CD 5E3”-variant we already inserted a virtual 12.85 kΩ resistor at this point to prevent a too-agressive blocking distortion caused by a very short attack time. See **Table 2** for all tube stage parameters of the “CD BAL”- variant. The cutoff frequency of HP2 is 0.41 Hz. The cutoff frequencies of HP3 and HP4 are 5.8 Hz and 5.8 Hz, respectively.

The cathodyne phase splitter clips in a very hard fashion due to its high open-loop gain. The “LTP”-variants use a long tail pair instead of the cathodyne phase splitter. The clipping is much softer – however, the circuit has 29.2 dB more gain. Therefore, we insert a static gain stage with gain k_4 ahead of the long tail pair. We provide three “LTP” variants: For “LTP 1” we reduce the gain by 29.2 dB and obtain the same total gain as given with the cathodyne circuit. For “LTP 2” we implement 14.6 dB, and for “LTP 3” we have 29.2 dB additional gain. For a more easily accessible sound comparison of the variants we can use a gain compensation which compensates the additional gain of the phase splitter ahead of the second tube stage. By default, this gain compensation is active, but it can be switched off by the user for the case that very pronounced high-gain sounds are required. See **Table 3** for all tube stage parameters of the “LTP”-variants. The cutoff frequency of HP2 is 10 Hz. The cutoff frequencies of HP3 and HP4 are 5.7 Hz and 5.6 Hz.

So far, we have seen five variants of phase-splitter circuits. We can introduce a variant of the first tube stage, as well. Instead of the original 12AY7 tube, we can select a 12AX7 tube with an additional gain of 4.9 dB. This additional gain is compensated for after the first tube stage in the case that the gain compensation is switched on. See **Table 4** for all tube stage parameters of the “CD 5E3”-variant with a 12AX7 input tube.

Table 2: Parameters and derived properties of all tube stages of the "CD BAL"-variant

	T1	**T2**	**T3**	**k1**	**T4**	**T3.vk**	**k2**	**T5**	**Vaa**
Circuit	CK	CK	CD		CK	CD		CK	
µ	44	100	100		125			125	
Ra	25,000	62,500	62,500		40,000			40,000	
isat	0.00220	0.00155	0.00160		0.11000			0.11000	
ibias	0.00120	0.00076	0.00073		0.04200			0.04200	
b	0	0	0		2			2	
type	0.5	0.5	0.5		0.5			0.5	
vs	238	238	238		346			346	
RL	100,000	100,000	56,000		3,000			3,000	
RL2									
RK	820	1500	1500		540			540	
RB									
kcomp	0	0	0		1			1	
kpk	0	0.2	0.5		0.5			0.5	
xth	*0.250*	0.255	0.272		0.309			0.309	
xdrop	*0.25*	0.570	0.384		0.437			0.437	
tattack	*0.01*	0.015	0.00085		0.00575			0.00575	
trelease	*0.05*	0.05	0.3872		0.0276			0.0276	
tck	0.0205	0.0375			0.00675			0.00675	
gain	-35.200	-61.538	-0.945	0.797	-8.721	0.970	0.797	-8.721	
Zout	20,000	38,462	56,000		2,791	1,150		2,791	
vinmax	2.84	1.28	51.56		23.39			23.39	
vinmin	-3.41	-1.24	-43.26		-14.45			-14.45	
voutmax	120.0	76.0	40.9	32.6	126.0	50.0	39.9	126.0	330.0
voutmin	-100.0	-79.0	-48.7	-38.8	-204.0	-42.0	-33.5	-204.0	-330.0
1 - kbias	0.455	0.510	0.544		0.618			0.618	
- kbias	-0.545	-0.490	-0.456		-0.382			-0.382	
xmax		47.64	0.80		0.86			1.32	
xmin		-39.70	-0.83		-1.03			-1.11	
xpkmax		9.364	0.091		0.087			0.288	
kODmax		93.5	1.5		1.4			2.1	
kODmin		81.0	1.8		2.7			2.9	

Table 3: Parameters and derived properties of all tube stages of the “LTP”-variants

	T1	**T2**	**T6**	**k1n**	**T4**	**T6.vout2**	**k2**	**T5**	**Vaa**
Circuit	CK	CK	LTP		CK	LTP		CK	
µ	44	100	100		125			125	
Ra	25,000	62,500	62,500		40,000			40,000	
isat	0.00220	0.00155	0.00160		0.11000			0.11000	
ibias	0.00120	0.00076	0.00074		0.04200			0.04200	
b	0	0	0		2			2	
type	0.5	0.5	0.5		0.5			0.5	
vs	238	238	238		346			346	
RL	100,000	100,000	82,000		3,000			3,000	
RL2			100,000						
RK	820	1500	820		540			540	
RB			6,800						
kcomp	0	0	0		1			1	
kpk	0	0.2	0.2		0.495			0.49	
xth	*0.250*	0.255	0.269		0.309			0.309	
xdrop	*0.25*	0.570	0.601		0.439			0.442	
tattack	*0.01*	0.015	0.015		0.00594			0.00663	
trelease	*0.05*	0.05	0.0500		0.0278			0.0285	
tck	0.0205	0.0375			0.00675			0.00675	
gain	-35.200	-61.538	-29.425	0.792	-8.721	29.629	0.772	-8.721	
Zout	20,000	38,462	57,871			64,808			
vinmax	2.84	1.28	2.40		23.39			23.39	
vinmin	-3.41	-1.24	-2.06		-14.45			-14.45	
voutmax	120.0	76.0	60.7	48.1	126.0	74.0	57.1	126.0	330.0
voutmin	-100.0	-79.0	-70.5	-55.9	-204.0	-86.0	-66.4	-204.0	-330.0
1 - kbias	0.455	0.510	0.538		0.618			0.618	
- kbias	-0.545	-0.490	-0.463		-0.382			-0.382	
xmax		47.64	26.91		1.27			1.51	
xmin		-39.70	-22.43		-1.48			-1.75	
xpkmax		9.364	5.209		0.258			0.372	
kODmax		93.5	50.07		2.05			2.44	
kODmin		81.0	48.49		3.87			4.60	

Table 4: Parameters and derived properties of all tube stages of the "CD 5E3"-variant with a 12AX7 input tube

	T1	T2	T3	k1n	T4	T3.vk	k2	T5	Vaa
Circuit	CK	CK	CD		CK	CD		CK	
µ	100	100	100		125			125	
Ra	62,500	62,500	62,500		40,000			40,000	
isat	0.00165	0.00155	0.00160		0.11000			0.12000	
ibias	0.00076	0.00076	0.00073		0.04200			0.04200	
b	0	0	0		2			2.5	
type	0.5	0.5	0.5		0.5			0.5	
vs	238	238	238		346			346	
RL	100,000	100,000	56,000		3,000			3,000	
RL2									
RK	1500	1500	1500		540			540	
RB									
kcomp	0	0	0		1			1	
kpk	0	0.2	0.5		0.5			0.7	
xth	*0.250*	0.255	0.272		0.309			0.325	
xdrop	*0.25*	0.570	0.384		0.437			0.388	
tattack	*0.01*	0.015	0.00085		0.00575			0.00155	
trelease	*0.05*	0.05	0.3872		0.0276			0.0234	
tck	0.0375	0.0375			0.00675			0.00675	
gain	-61.538	-61.538	-0.945	0.797	-8.721	0.970	0.940	-8.721	
Zout	38,462	38,462	56,000		2,791	1,150		2,791	
vinmax	1.45	1.28	51.56		23.39			26.83	
vinmin	-1.24	-1.24	-43.26		-14.45			-14.45	
voutmax	76.0	76.0	40.9	32.6	126.0	50.0	47.0	126.0	360.0
voutmin	-89.0	-79.0	-48.7	-38.8	-204.0	-42.0	-39.5	-234.0	-330.0
1 - kbias	0.539	0.510	0.544		0.618			0.650	
- kbias	-0.461	-0.490	-0.456		-0.382			-0.350	
xmax		30.17	0.80		0.86			1.21	
xmin		-35.33	-0.83		-1.03			-1.02	
xpkmax		5.870	0.091		0.087			0.349	
kODmax		59.2	1.5		1.4			1.9	
kODmin		72.1	1.8		2.7			2.9	

9. Software implementation

The plugin format is JSFX (Jesusonic effect). JSFX plugins are scripts written in the EEL2 language; they can be loaded directly into the DAW REAPER or into the VST3 or AU Plugin YSFX. The scripts are compiled by its host immediately after loading them. A programming reference for JSFX can be found in [7]. The binaries of the plugin YSFX can be downloaded from

[8] and are available for Windows, MacOS and Linux. A detailed description of a simple JSFX plugin can be found in [5].

The JSFX format has been chosen because it is orders of magnitudes easier to program a simple plugin compared to a plugin in the VST or AU format written in C++ using an SDK like JUCE. The disadvantage is of course that a VST or AU bridge like YSFX is required if we want to use the plugin outside of REAPER. Another disadvantage of the JSFX format can be seen in some given limitations. However, we do not push these limits in the present project. For plugins with a very complex GUI, and a comfortable preset handling the author would still recommend to program them directly as VST or AU plugins in C++ with the aid of JUCE.

The EEL2 language is not an object orientated language. However, a DSP block needs to hold its own states and properties, and fortunately, EEL2 allows pseudo-objects. A function may be called up within a namespace and will use global variables of that same namespace if they are marked as “instance”-variables in the function definition. A typical DSP block class consists of three functions: An initalize-function which initializes the states, a set-function which sets the properties, and, finally, a process-function which takes care of the signal processing for each sample. Different instances of a DSP block simply use different namespaces.

The author has developed the following DSP libraries for JSFX:

- HK_LIB_TOOLS.jsfx-inc: some useful tools
- HK_LIB_SMOOTH.jsfx-inc: parameter smoothing
- HK_LIB_GAIN.jsfc-inc a gain block with parameter smoothing
- HK_LIB_FLT_II.jsfx-inc: very efficient impulse-invariant filters
- HK_LIB_FLT_DF.jsfx-inc: filters in direct form 2
- HK_LIB_FLT_SV.jsfx-inc: state-variable filters
- HK_LIB_MOD.jsfx-inc: tools for modulation effects
- HK_LIB_ADNL.jsfx-inc: antiderivative nonlinearties
- HK_LIB_PKD.jsfx-inc: a peak detector
- HK_LIB_TUBE.jsfx-inc: tube stage circuits

The DSP blocks of the last three libraries are described in detail in chapters 2 to 6 of this paper. State-variable filters are described in [4] and [5] while the other libraries implement common DSP blocks or some general tools.

The EEL2 language does not come with a GUI library but supports parameter input with a standard slider interface. In order to program a more comfortable GUI, we can use the library “ui-lib.jsfx-inc” by Geraint Luff [9]. The present author modified that library in its look and feel, and renamed it “hk-ui-lib.jsfs-inc”. The library “HK_LIB_GUI.jsfx-inc” is using this modified library and extends it via some high-level control elements. The script of the plugin is called “TWD DLX.jsfx”.

All libraries and the script of the plugin are stored in the folder “HK” of the zip file accompanying this paper.

At the beginning of the script “TWD DLX.jsfx” we include the necessary libraries via “import” statements. A “desc” statement describes the search name of the plugin in REAPER. The “in_pin” and “out_pin” statements describe the function of the input and output signals. With the “options” statement of “no_meter” we tell the host not to show input and output meters. The “slider” statements define all those plugin parameters that can be changed by the user via the standard slider interface, in the programmed GUI, or via host automation.

The “@init”-section of the source code is called after the plugin is loaded. In a first step we initialize all strings arrays which are necessary for the programmed GUI. Then we define all control elements of the programmed GUI, and initialize flags for parameter changes for all plugin parameters. We initialize all DSP blocks and variables, and set the parameters of all DSP blocks which cannot be controlled by plugin parameters. Then some strings that appear in the programmed GUI are defined. Last but not least we define the function “parameter_update()”. This function handles all plugin parameter updates and is designated the “@slider”-section. It is called up in the “@ gfx”-section, as well, if there has been any user interaction with the programmed GUI.

The “@slider”-section is called when slider values have changed by user input or host automation. We simply call the function “parameter update()” here.

The “@buffer”-section is called before the samples of a block are processed. We use this section to initiate unmuting of the plugins output shortly after muting has been initiated by a tube circuit variant change in the function “parameter-update()”.

The “@sample”-section is called for each signal sample. Here we do all the DSP of the plugin. Since we use libraries for the DSP blocks, the code is still rather simple and hopefully self-explaining, too.

Since we implement a mono tube amp, we create a mono signal if there is more than one input channel. Before we process the input signals, we implement the cathode coupling of T4 and T5 according to **Fig. 27**. In the next step we implement the power supply emulation according to **Fig.28**. Afterwards we implement the signal path according to **Fig. 26**. For the case of a “LTP”-variant we slightly modify **Fig. 26** and **Fig. 28** as described in chapter 8. At the end of the signal path, we copy the signal sample of channel 0 to channel 1. Finally we mute both channels for a short time if a variant change has been initiated in the function “parameter_update()”.

The “@gfx”-section is called whenever the GUI needs to be redrawn. See [9] for a quick-start tour and [10] for the full documentation of “ui-lib.jsfx-inc”. See “HK-LIB-GUI.jsfc-inc” for the source code of the additional high level GUI control elements.

With the instruction “control_start("main","tron")”, we initialize the GUI using the “main” screen for startup, and we select the “tron” look and feel mode. The instruction “gui_scroll_scale_set(48)” is defined in “HK_LIB-GUI.jsfx-inc”, it says that not more than about 48 mouse wheel steps are necessary to cover the entire range of a control element. Beside the “main” screen which contains the most important control elements, we have an “options” screen which contains optional control elements. Moreover, there is the “about” screen which informs the user about the plugin version and author. For the case that the user has interacted with the GUI, the function “parameter_update()” is called.

10. User manual of the plugin

The file “TXD DLX-jsfx” is a script of an JSFX plugin which emulates a Fender Tweed Deluxe guitar amplifier. The emulation is based on the famous 5E3 circuit, but in addition some circuit variants are also emulated. This script may directly be run in the DAW REAPER, or it may be loaded with the VST3 or AU plugin YSFX and used in any host which supports VST3 or AU plugins. Please note that only the amp is emulated. An impulse response loader needs to be placed after the amp emulation. The file “Tweed Deluxe Loudspeaker.wav” comes with the plugin and is a good choice for the impulse response of a typical close-miked Tweed Deluxe amp speaker. It is recorded with a sample rate of 48 kHz. A good reverb plugin subsequent to the IR-loader is all that is needed to create a typical high-quality electric guitar sound.

Installing the plugin requires some actions on your (the user’s) part:

If you are not a REAPER user you will need to download the plugin YSFX from [8]. The plugin is available for Windows, MacOS and Linux. Download and unzip the desired zip file, and copy the extracted VST3 or AU file into your VST3 or AU folder. In case the link in [8] would not work anymore, you can find a copy of the YSFX binaries in the zip file accompanying this paper.

Open the YSFX plugin in your host. Press the “Load” button of the YSFX plugin and watch in which default folder YSFX looks for JSFX files. Then copy the folder “HK” with its complete content from the content of the zip file (that accompanies the present publication) into that folder. You could also copy the folder “HK“ to any other place in your computer memory, but it will then be harder to find it via YSFX. Now press the “Load“ button of YSFX again. Open the folder “HK” and open the file “TXD DLX.jsfx”. Now you will see the “Main” screen of the TWD DLX plugin as shown in **Fig. 29**.

Please note that there is a bug in YSFX! After loading a script, it defaults to a sample rate of 44.1 kHz. You will need to bypasss and then un-bypass the YSFX plugin after loading a new script in order to ensure that the actual sample rate of your host is used by YSFX. Now you are ready to use the TWD DLX plugin. First, however, we will describe the rest of the YSFX GUI, and only then concentrate on the TWD DLX plugin.

The “Recent” button of YSFX gives you quick access to the recently loaded JSFX plugins. Don’t forget to bypass and then again un-bypass the plugin if you use this button.

The “Edit” button opens the script of the currently loaded plugin. Here you can edit the script and see all global variables and their current values. Only programmers should use this button, though!

The “Preset” button could load manufacturer presets for the plugin. Note, though, that the present author does not provide any presets for the TWD DLX plugin. User presets are not supported at all in YSFX.

The “Graphics” button toggles between the programmed GUI of the plugin, and the standard slider interface.

Between the “Recent” and “Edit” buttons, the file name of the loaded script and the number of supported input and output audio channels are displayed.

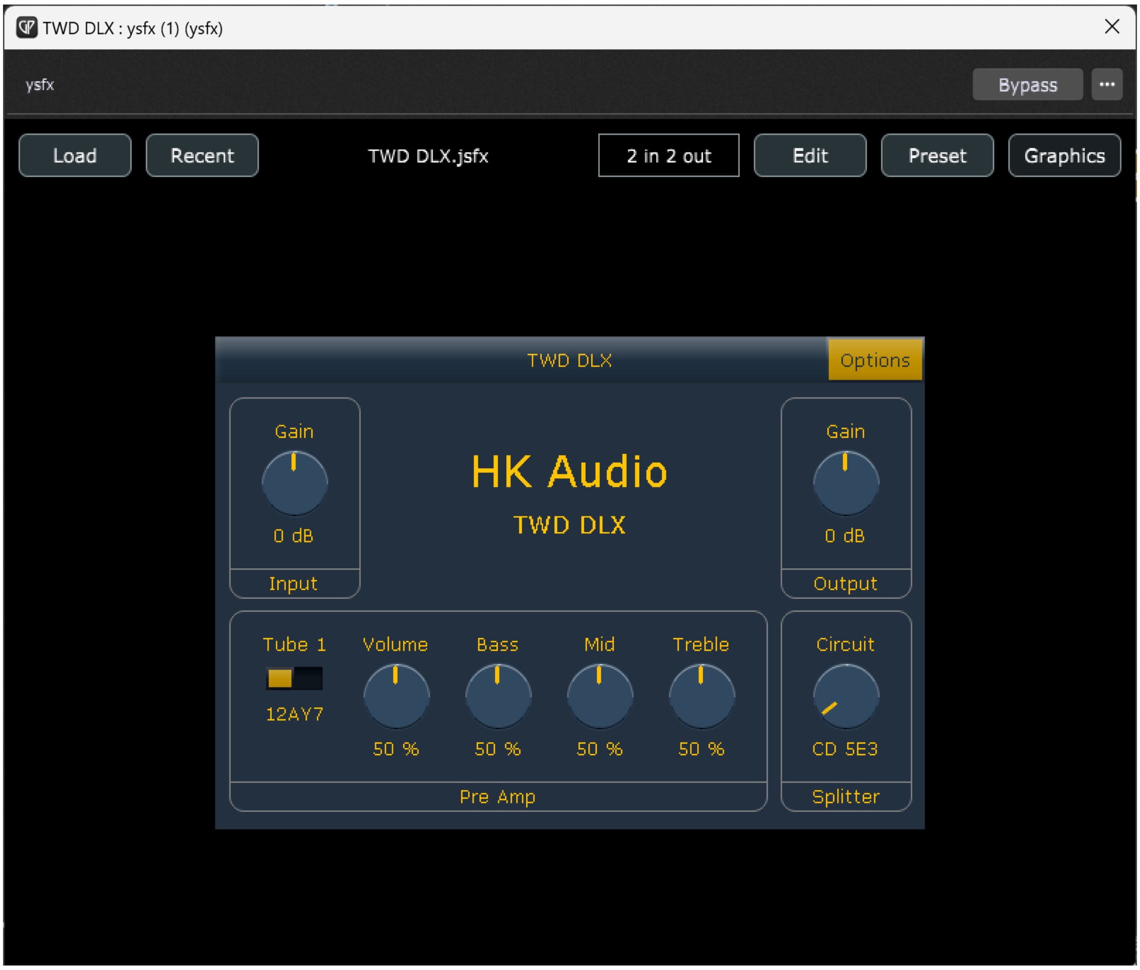


Fig. 29: Main screen of the TWD DLX plugin when loaded in YSFX

If you already are a REAPER user, you can copy the “HK” folder with its content into the “Effects” folder of the REAPER resource path. This path is easily opened in the explorer or finder via the options menu of REAPER. You will find the TWD DLX plugin under the JS effects with the Name “HK TWD DLX”.

Fig. 29 depicts the “Main” screen of the TWD DLX plugin. A double click on a dial will restore its default value. The mouse wheel changes the value of a dial if the mouse pointer is moved over that dial. Dragging the mouse pointer upward and downward over a dial will change that dial value, too. Displayed toggle switches change their states by clicking on them.

The input gain value should be set to 12 dBu below the full-scale input level of the deployed audio interface if an authentic distortion behavior is desired. For example, if your audio interface has a full-scale input level of +13 dBu, you should set the input gain to 1 dB. You can of course also use variable input gain to compensate for different output levels of guitars, or just bring in some additional gain or attenuation.

The 5E3 circuit of the Tweed Deluxe amp uses a 12AY7 triode in its first tube stage. However, some amps of this type – such as Neil Young’s specimen – are modified via the installation of a 12AX7 triode instead. This leads to an additional input gain of 4.9 dB. However by default we compensate the additional gains of variants with an additional attenuation at the volume control position. This way it is much more easy to compare the effect of the variants on the distortion behavior because the overall gain stays constant. If you do not like this compensation you can turn it off in the “Options” screen. If you want to access the maximum over all gain of some circuit variants you need to switch off the gain compensation too. The “12AY7” is hard to overdrive and produces only negligible distortion at normal input levels where the “12AX7” already brings in some “excitement”.

The volume control reduces the gain after the first tube stage. We emulate the “Bright” channel of the original amp and assume that its second volume control is turned off. With the volume control settings of 0, 25, 50, 75 and 100 % we obtain infinite, 24 , 12, 5 and 0 dB of attenuation.

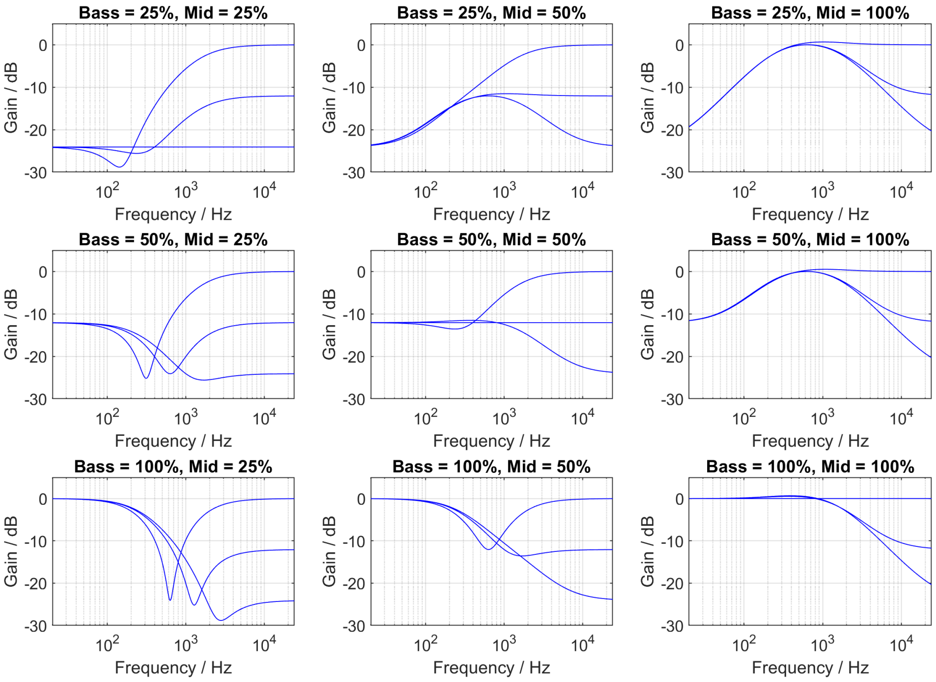


Fig. 30: Frequency responses of the tone stack for Fmid = 630 Hz and Qmid = 0.355

The original 5E3 circuit has two input channels with a separate volume controls and a common tone control. In Manfred Zollner’s video “T99 Fender Deluxe” [11] you will find an excellent description of the functions and the complicated interactions of these controls. However we emulate the “Bright” channel only and replace the tone control with a “universal tone stack”. We do not emulate any interaction between the volume and the tone stack controls as well.

The “univeral tone stack” of our emulation is much more versatile than the original tone control. It can emulate the three band tone stacks e.g. found in later Fender or Marshall amps and their derivatives very well. However the interactions between the bass, mid and treble controls are eliminated and the different scalings of the controls are omitted. This results in a more predictable frequency response. This tone stack does also enable us to obtain all the tones the 5E3 “Bright” channel, and any combination of the “Bright” and “Normal“ channel will give. Moreover, many more tonal settings are possible, as well. The frequency response of the tone stack is always completely flat if the bass, mid, and treble controls are set to the same value. The attenuation applied to the three bands of the tone stack is the same as for the volume control if the same values are set. In **Fig. 30** we see all 27 frequency responses of the tone stack for all 27 control value combinations of 25, 50 and 100%. Note the small negative peak in the setting: bass = 50% , mid = 50 % and treble =100 %. It is typical for three band tone stacks but can not occur in the 5E3 circuit. This negative peak will disappear if the mid-value is increased a bit. Note that the most useful 5E3 tone settings can be achieved by reducing the treble-value in respect to the bass-value a bit for high volume settings. But at low volume settings the treble-value should be higher. than the bass-value. The mid control should always be set somewhere between the bass and treble control for Tweed Deluxe sounds. However the “scooped” tones of later Fender amps may be achieved with mid-values lower than the bass- and treble-values. In the “Options” screen we can further modify the tone stack.

The original 5E3 circuit uses a cathodyne phase splitter. The output impedances of its two output paths are extremely different, the consequence being a very unbalanced power amp that generates a lot of even-order distortion. We call this variant “CD 5E3”. The phase splitter circuit variant “CD BAL” uses a series resistor at the cathode output to match both output impedances. This leads to less even order distortion, and to less compression due to blocking distortion of the second power amp tube. The sound of this variant reminds the author of the sound of Marshall Plexi amps. Both cathodyne variants are clipping very harshly due to the very strong local feedback. Since the power amp tubes clip first and in a much softer fashion, this effect is not as strong as it could be – but is still perceivable as an “aggressive” distortion. We therefore introduce variants using the long tail pair phase splitter of the 6G3 circuit of the Fender Brownface Deluxe amp. However, at this point we still do not use the feedback loop from the loudspeaker output to the phase splitter input found in the 6G3 circuit. We simply replace the phase splitter. From this, we can expect a sound closer to a VOX AC 15 than to a Fender Brownface Deluxe. The clipping of the long tail pair circuit is much softer while the gain is about 29.2 dB higher compared to the cathodyne circuit. The output impedances and the gains are reasonably balanced. The output voltage swing is higher compared to the cathodyne circuit. This results in a fairly well-balanced power amp with more compression due to blocking distortion of both power amp tubes. Three “LTP” variants are offered here. In “LTP 3” we use the additional gain of 29.2 dB to its full extent. With the help of an attenuation of 14.6 dB just ahead of the phase splitter, we use only 14.6 dB additional gain in “LTP 2”. In “LTP 1”, we use no additional gain at all due to an attenuation of 29.2 dB just ahead of the phase splitter. All “LTP” variants sound softer and more compressed than the “CD” variants. To the authors ears, the “LTP-3” variant is the best variant for clean and edge-of-breakup sounds. As with the tube 1 variants, we compensate the additional gain at the position of the volume control if gain compensation is switched on.

The output gain can vary the output level of the plugin. If it is set to 0 dB the amp will saturate at about -12 dBFS.

We reach the “Options” screen shown in **Fig. 31** by clicking the “Options” button of the “Main” screen.

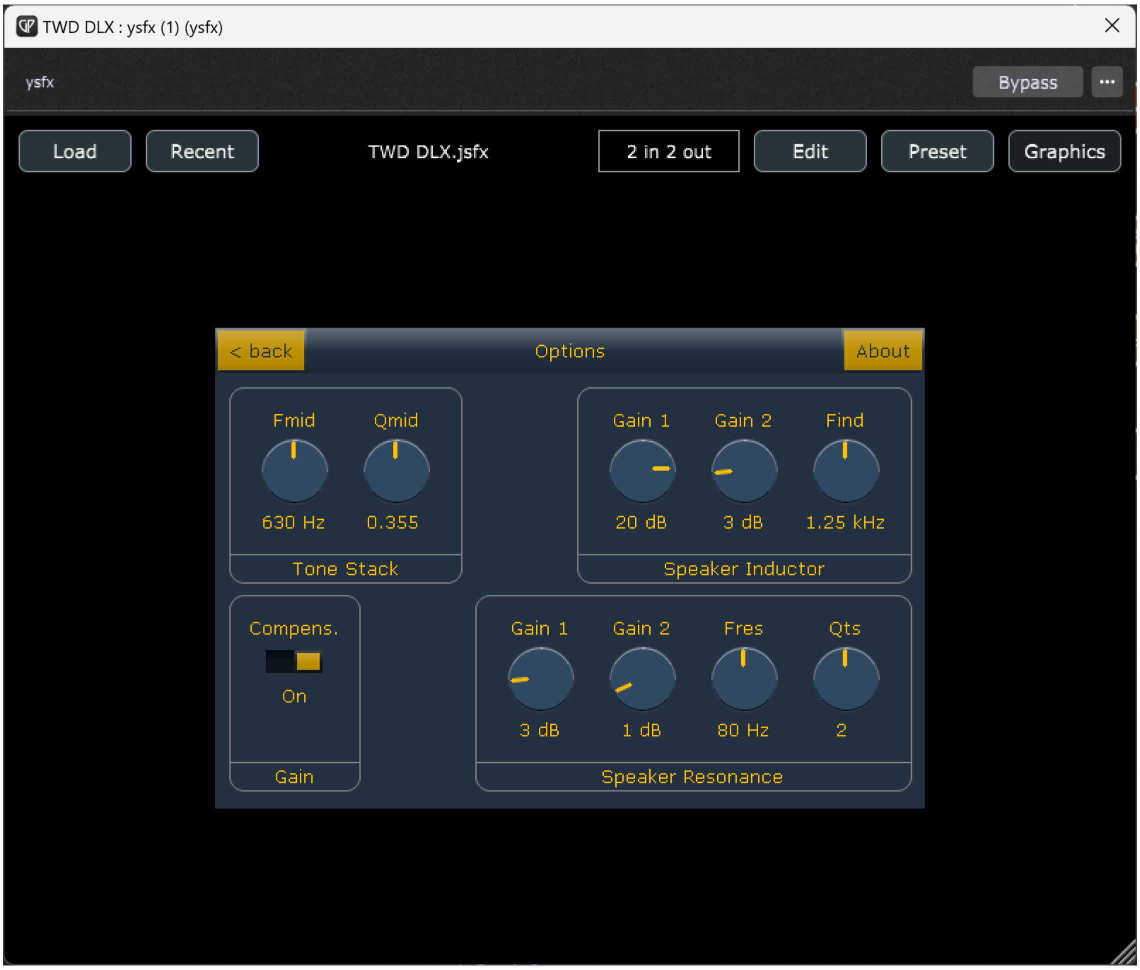


Fig. 31: Options screen of the TWD DLX plugin when loaded in YSFX

The effect of the gain compensation is already described above with the circuit variant controls. It may be switched off here.

The control “Fmid” can shift the center frequency of the middle-frequency band of the tone stack. The control “Qmid” changes the quality factor and thus the bandwidth of the middle band of the tone stack. For values above 0.5 we enter the territory of active or LC tone stacks. For example, the sound of the “Mid-Booster” in Eric Claptons Stratocaster could be obtained this way without the need of an extra equalizer plugin in front. In **Fig. 32** and Fig. **33** you can observe the effect of the two optional tone stack controls.

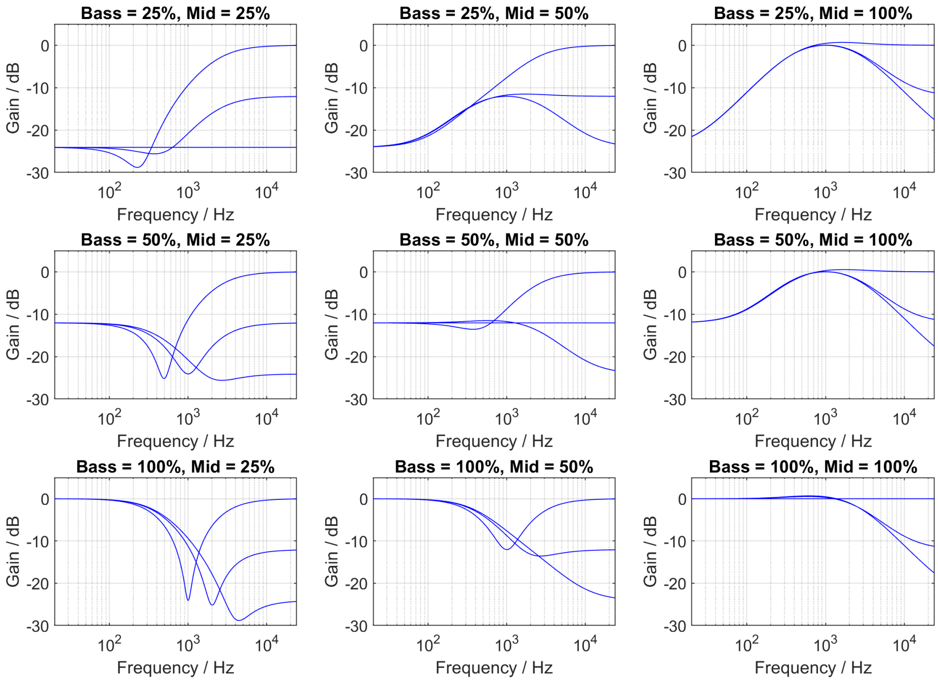


Fig. 32: Frequency responses of the tone stack for Fmid = 1000 Hz and Qmid = 0.355

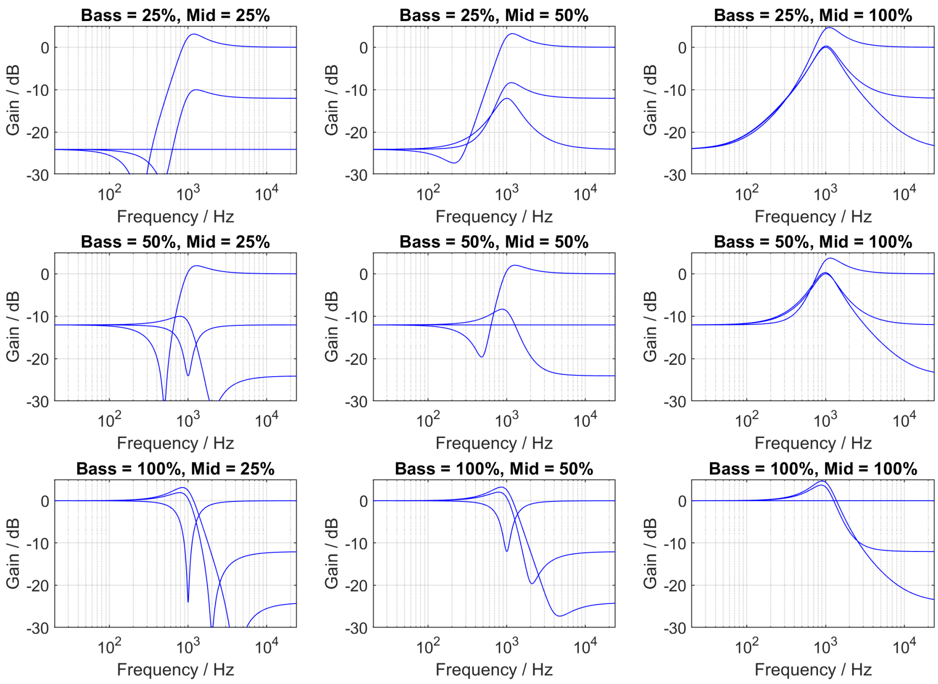


Fig. 33: Frequency responses of the tone stack for Fmid = 1000 Hz and Qmid = 1.4

The seven controls related to “speaker” allow for modifications of the interaction between the amp and the connected loudspeaker.

“Speaker inductor” with its three controls takes care of modifying two high shelf filters, one located ahead and the other after the power amp tubes. The treble gain is controlled with “Gain 1” and “Gain 2”. “Find” sets the frequency where the loudspeaker inductor begins to dominate the loudspeaker impedance. The default values for these three controls are realistic and yield a good sound. However, you are encouraged to modify the settings since they do have a huge impact on the sound. The frequency response of power amps including a global feedback loop could be emulated by reducing “Gain 1” to values close to 0 dB. The frequency response of amps with a presence control could be emulated with moderate values of “Gain 1”. Emulating the harder clipping of power amps with a global feedback loop this way is, however, not possible.

The four controls related to “speaker resonance” modify two peak equalizer filters, the first located ahead, and the second after the power amp tubes. The gain at the resonance frequency of the loudspeaker is controlled with “Gain 1” and “Gain 2”. The resonance frequency of the loudspeaker is set using the control “Fres”. The total system-quality factor of the loudspeaker is set with the control “Qts”. Realistic gain values would be 10.5 dB for “Gain 1” and 2.5 dB for “Gain 2”. Most guitarist would not like the resulting “boomy” sound, though; they probably would try to counteract with an external equalizer. The default values therefore are much lower. As already mentioned for the “inductor” controls, we can emulate the frequency response of power amps with a global feedback loop by using “Gain 1”-values close to 0 dB.

Please note that we usually work with impulse responses of very closely-miked loudspeakers. The “in the room”-sound of a Tweed Deluxe combo does however have much less bass due to the “acoustic short cut” of its open back enclosure. For this reason, a strong resonance peak could be beneficial for its sound “in the room”.

We reach the “About” screen shown in **Fig. 34** by clicking the “About” button of the “Options” screen. Returning to the “Options” screen is done by clicking the “< back” button of the “About” screen. We arrive back at the “Main” screen by clicking the “< back” button of the “Options” screen.

Please note that changes of circuit variants do not use parameter smoothing. Thus you might hear a clicking sound, and a short signal pause during the change. It is consequently not recommended to use different circuit variants in the middle of a running performance. All other parameter changes of the plugin are smooth and generate no gap.

It is worth to mention that the TWD DLX plugin needs only less than half of the computer resources if we compare it with modern amp plugins from Neural DSP or with the NAM plugin with standard quality profiles. However the sound quality and playing experience is at least on par with these competitors. It is worth to mention too that Neural DSP plugins introduce a latency of more than 1 ms compared to only 2 sample periods of the TWD DLX. Thus TWD DLX is a very good choice if you are after a high quality live setup with very low latency and running on a fanless tablet like the authors Microsoft Surface Pro 7.

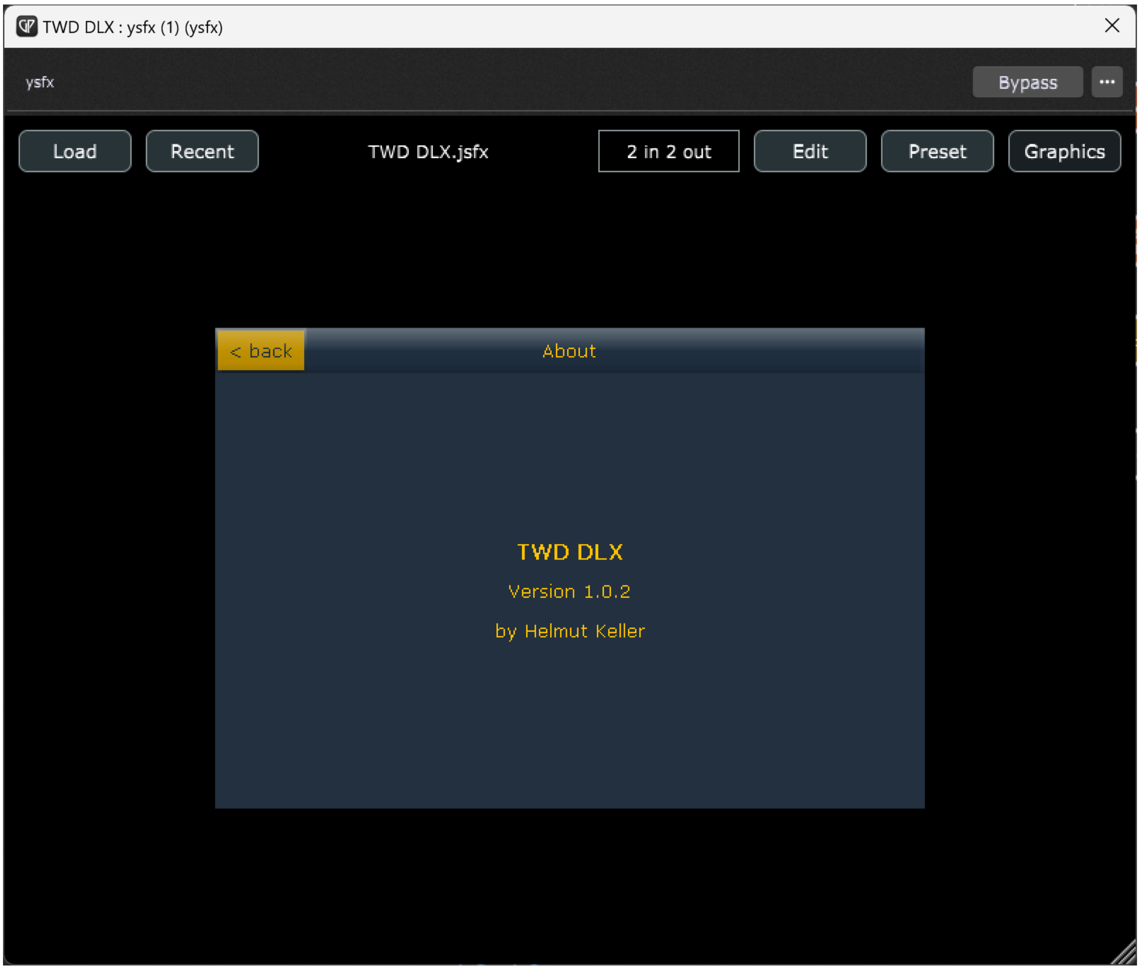


Fig. 34: “About” screen of the TWD DLX plugin when loaded in YSFX

You will find three further JSFX plugin of the author in the folder “HK”:

The plugin “Volume Wah” is an improved version of the plugin described in [5]: A programmed GUI has been added. The signal processing is modular now by using the authors DSP block libraries. The parameter smoothing is improved. The frequency scaling is even more flexible. With its default settings the plugin emulates the famous VOX V847 Wah-Wah pedal. The plugin can function as volume pedal as well.

The plugin “360 ° Panner” implements a vector rotation based panorama control. When panning hard right for example the original mono signal will appear on the right channel and the original side signal will appear on the left channel. The pan position can be modulated too. This way it is possible to create the illusion that a stereo signal is rotating 360 ° around the listener. You can hear this effect in the middle part of the Pink Floyd song “Poles Apart” where the orchestra seems to rotate slowly around you.

The plugin “Blender” can blend or switch between two audio channels A and B. It is very useful to blend between two IR-loader plugins or to switch between two guitar amp plugins. The plugin has four input channels for two stereo inputs. You may restrict the number of input channels of the plugin to two input channels in your DAW. In this case the plugin will blend between the two mono inputs or between the left and right channel of a stereo input.

11. Summary

State-of-the-art models for tube amplification stages have been developed. The models were used in an emulation of a Fender Tweed Deluxe guitar amp. Various additional circuit variants and a universal tone stack allow for a large variety of “vintage” guitar sounds. The amp model is available at no charge as a JSFX plugin. With a bridge plugin like YSFY it can be run in every VST3 or AU host. The present plugin is much more effective compared to other currently available modern solutions; at the same time it is at least on par in terms of sound quality and playing experience.

A tube amplification stage includes only about 17 relevant parameters that need to be defined. Some of them can be directly derived from the given circuit diagram or from tube data sheets. The remaining parameters can at the very least be roughly estimated. Since the author does not own a Tweed Deluxe amp, he fine-tuned these latter parameters based on his own expectations as to sound and feel, comparing the results to recordings for which a Tweed Deluxe has probably been used. If measurement results of signals from inside a reference Tweed Deluxe were available, a more accurate emulation could be probably achieved. The question remains whether this is actually desirable. For example, more or less annoying side-effects found in real amps (such as hum, hiss, ripple in the power supply voltage, etc.) and huge production tolerances are not really attractive to be emulated – and in the present amp model, they were not included.

Currently the author has no effective solution to implement global feedback loops as they are often found in higher-power amp circuits. This may be part of further tube amp modeling projects.

The author wants to thank Mr. Tilmann Zwicker who massively improved the english language of this paper.

12. Literature

[1] J. D. Parker et al., “Reducing the Aliasing of Nonlinerar Waveshaping Using Continuous-Time Convolution”, http://dafx16.vutbr.cz/dafxpapers/20-DAFx-16_paper_41-PN.pdf

[2] Neural DSP News, “A new audio engine powering Neural DSP plugins”, January 22, 2021, https://neuraldsp.com/news/a-new-audio-engine-powering-neural-dsp-plugins

[3] Tube Amp Doctor, “Documentation of the 5E3-kit”, https://www.tubeampdoctor.com/media/pdf/17/ab/8a/full_documentation_one-twelve-20-01-2022.pdf

[4] H. Keller, “Digital State-Variable Filters”, GITEC, 2023, https://gitec-forum-eng.de/filter-design-digital-state-variable-filters/

[5] H. Keller, “Wah-Wah Plugin”, GITEC, 2023, https://gitec-forum-eng.de/write-your-own-wah-wah-plug-in-filter-design-continued/

[6] M. Zollner, "Physik der Elektrogitarre", https://www.gitarrenphysik.de, translated into english by T. Zwicker: https://gitec-forum-eng.de/the-book/

[7] Cockos Inc., “JSFX Programming”, https://www.reaper.fm/sdk/js/js.php

[8] J. Vanlier, “YSFX binaries”,https://github.com/JoepVanlier/ysfx/releases/tag/v0.0.2

[9] G. Luff, “ui-lib.jsfx-inc”, https://github.com/geraintluff/jsfx-ui-lib

[10] G. Luff, “API Documentation of ui-lib!”, https://geraintluff.github.io/jsfx-ui-lib/doc/html/

[11] M. Zollner, “Tutorials”, https://www.gitarrenphysik.de/tutorials