

Low Level RF Modeling for the MEBT Rebunchers

Lawrence R. Doolittle, LBNL

This technical note is intended exclusively for internal use inside the SNS Front-End Systems Group. The primary purpose of the technical note series is to promote rapid dissemination of results for work in progress. No formal review has taken place to ascertain the accuracy of its contents, or the consistency/compatibility of the information with other SNS work.

1 INTRODUCTION

The four MEBT rebuncher cavities need control electronics to set their amplitude and phase to the values needed for proper beam propagation. The raw power to drive them (20kW to 40 kW at 402.5 MHz) comes from commercially packaged grounded-grid triode amplifiers. The “Low Level RF” subsystem discussed in this note needs to:

- send an RF waveform to the triode assembly
- measure the cavity detune angle (tuner error signal)
- use EPICS for command and readback.

An architecture for this subsystem was reviewed on May 10, 2001[1]. A digital signal processing section (based on an FPGA) is connected to a low latency measurement of the cavity field vector, and controls the RF drive vector sent to the triode amplifier.

At its simplest, the digital control unit could turn the RF output on for 1 ms after a 60 Hz trigger, and provide a readback of the cavity voltage waveform to the operator. Previous experience with the RFQ shows that the decay segment of this waveform can be analyzed to determine the cavity detune angle. The benefit of having a low-latency measurement/control loop is that digital feedback can be included with no additional hardware.

The body of this paper explains a numerical model of the RF system, and shows the expected behavior of the system. It shows how feedback and feedforward can be used to keep the beam loading transient under control.

I also provide documentation for how to work with the numerical model, which is provided in the form of a c program. The program, along with this tech note, are available for download[2].

2 MODEL SETUP

The block diagram of the RF system model is shown in figure 1. An explanation of the feedforward terms is included in figure 2. The whole loop is modeled at baseband, leaving out all the mixers and frequency conversion. This means

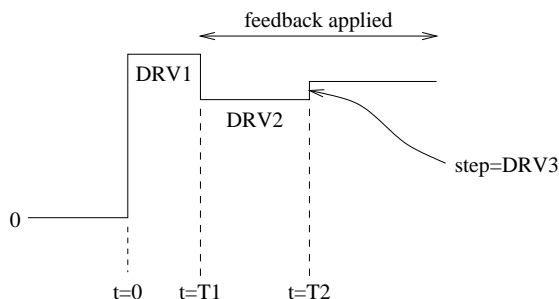


Figure 2: Schematic of feedforward waveform structure. T2 is adjusted to match the arrival of the first beam.

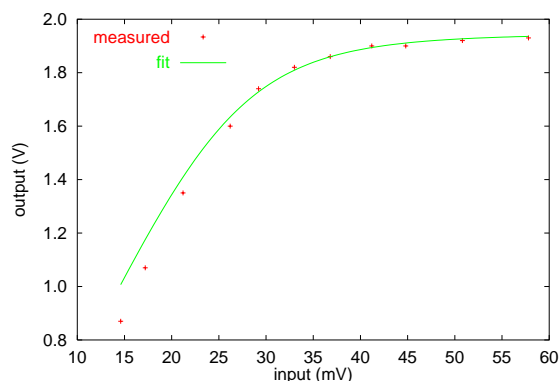


Figure 3: Triode amplifier saturation characteristics.

that all signals are represented in vector form, and that filter poles are not restricted to negative real and conjugate pairs. In particular, detuning a cavity takes its pole position from a nominal $-1/\omega_0$ slightly into the imaginary direction. This shift properly introduces a coupling between the otherwise linearly independent vector components of the drive signal.

All the low-pass filters are set up to have low frequency gain of 1. Thus, the numerical amplitudes used are given in suitably normalized units.

A unity signal (1,0) coming from the amplifier corresponds to full output, (20kW for cavities 1-3, 40 kW for cavity 4). I define the small signal amplifier gain as unity, which implicitly defines the input voltage scaling.

Triode clipping is described with a harshness parameter c , such that the output signal v_o from the tube based on its input v_i varies according to $v_o = v_i \cdot (1 + |v_i|^c)^{-1/c}$. The saturated output amplitude from this equation is 1. While some phase shift with drive level is observed, this effect is

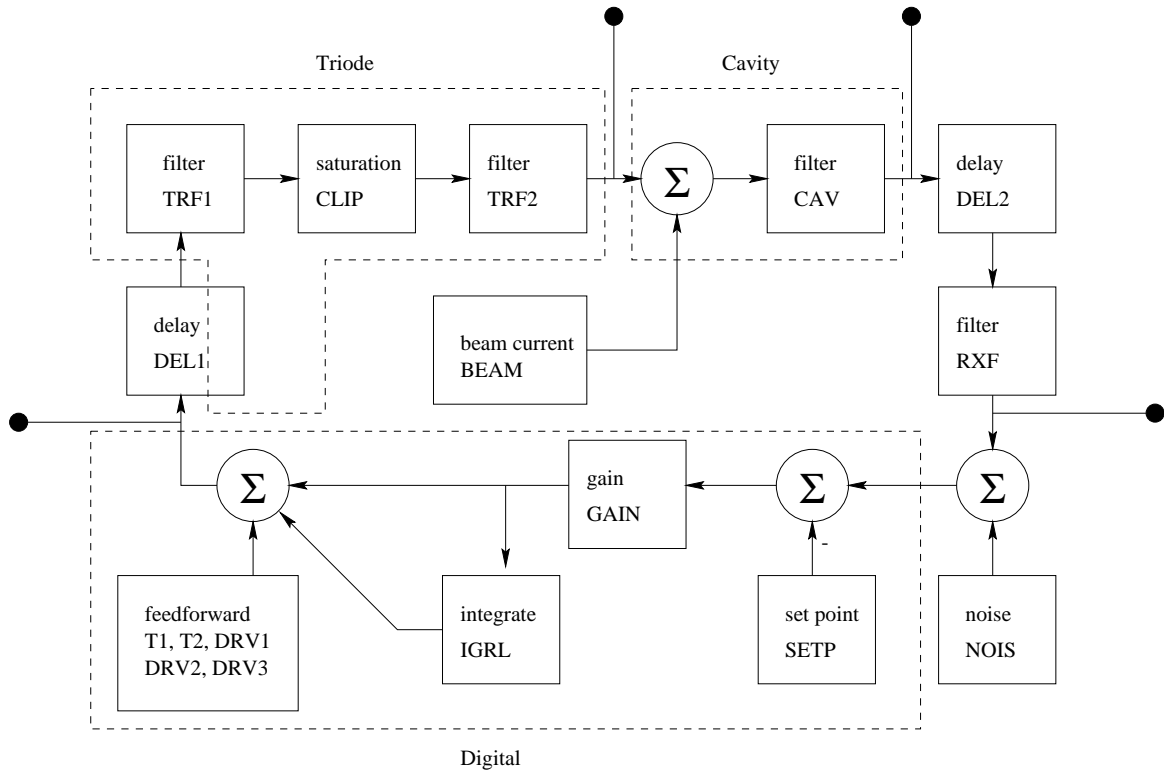


Figure 1: Block diagram of the system as modeled.

not yet included in the model. When $c = 5$, as provides a decent fit to measurements shown in figure 3, it takes about 1.5 (unitless) drive level from the control system to reach 97.5% of this amplitude, 95% of the power. This level should correspond to full scale digital output.

For the cavity, unity is the equilibrium voltage on resonance given the rated amplifier power. Rebuncher cavity $Z_{shunt}T^2/L$ from the Superfish runs[3] is based on a 2.72 cm length, which is the half-length of the cell used in the Superfish analysis. The scaling from current to beam-referred cavity voltage is given by $Z_{shunt}T^2$ for the whole cavity (not the half cell), which involves another multiplication by two. Furthermore, the computed value of Z_{shunt} is based on a cavity Q of 21700. This needs to be scaled to the loaded cavity Q , estimated (80% theoretical Q_0 , critical coupling) as 8700. Apply this impedance to unity cavity voltage to get unity beam current.

The following table shows the key cavity properties that are used to set up and interpret the model:

Cavity	1	2	3	4	
Bore	30	36	36	36	mm
$Z_{shunt}T^2/L$	11.75	6.85	6.85	11.75	MΩ/m for Q_0
$Z_{shunt}T^2$	256	149	149	256	kΩ for Q_L
Unity Power	20	20	20	40	kW
Design Power	11.0	6.8	8.1	28.2	kW
Unity Voltage	101	77	77	143	keV
Design Voltage	75	45	49.3	120	keV
Setpoint	0.74	0.58	0.64	0.84	
Unity Current	395	517	517	559	mA

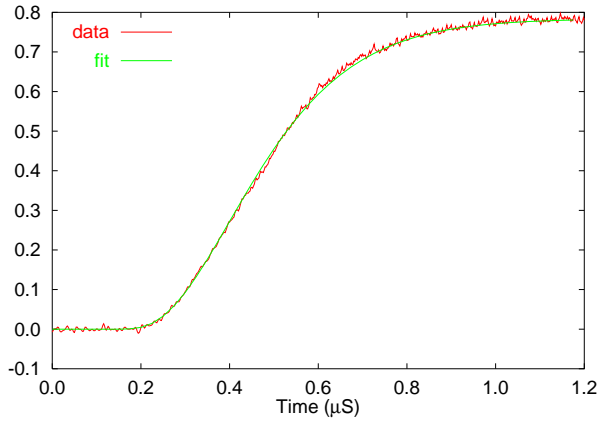


Figure 4: Triode amplifier risetime.

Beam current *du jour* within the minipulse is 38 mA[4], which converts to normalized values between 0.07 and 0.10, depending on the cavity.

Figure 4 shows a measured step response of the amplifier (at 60% of full power), along with a calculated curve based on a filter with poles at -9 , $-7.5+3j$, and $-7.5-3j$ (units are $(\mu s)^{-1}$), plus 175 ns additional delay.

We will use three pole, 2.5 MHz bandwidth (at 50 MHz carrier) receive (RX) filters to keep spurious signals out of the receiver. The manufacturer provided magnitude and delay plots for the customized units. I get good fits to these plots when I use pole locations -18 , $-9.5-15j$, and $-9.5+15j$,

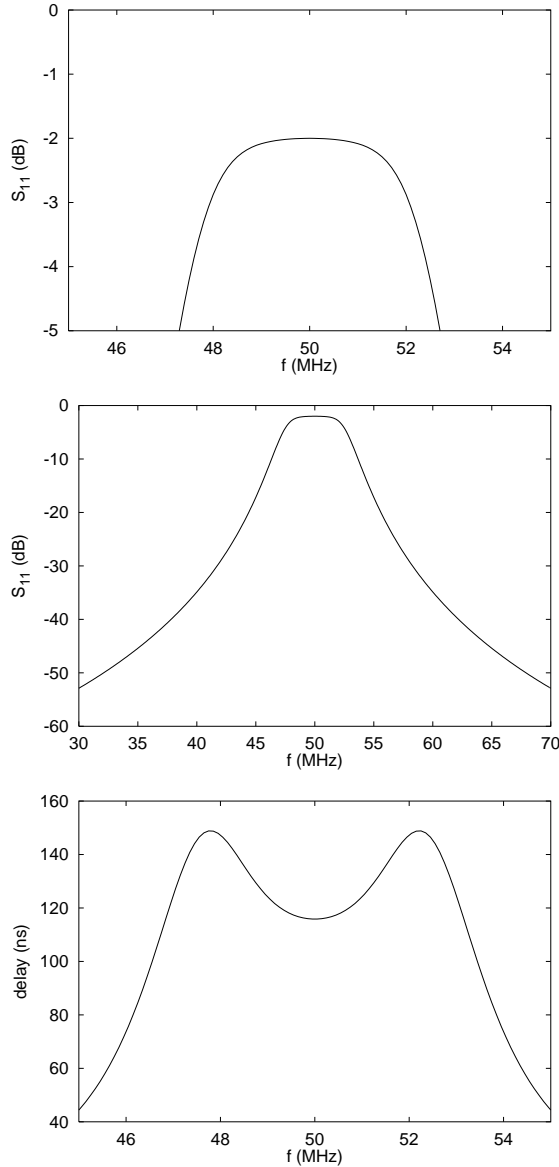


Figure 5: RX Filter amplitude and delay characteristics.

in units of $(\mu s)^{-1}$. The manufacturer's plots were faxed in, so I can't show those here. I do show the results of my three-pole model in figure 5.

So far, the digital section is handled in the same manner as the analog. The floating point numbers used for the amplitudes are effectively not quantized. The integrator accumulates IGRL times the error signal at each time step. No clipping as applied; amplifier saturation is already included as discussed earlier, and the examples shown here do not send the digital section beyond its planned output range.

The delays around the loop, grouped by physical component, are summarized here:

component	delay, ns	DEL1	DEL2
Cavity	-		
Cable (32 ft.)	35		35
RX Filter	120		
ADC (5 cycles)	125		125
Digital (4 cycles)	100		100
DAC (1 cycle)	25		25
TX filter	50	50	
Cable (100 ft.)	110	110	
Amplifier	450	175	
Cable (82 ft.)	90	90	
Cavity	-		
Total	1105	425	285

The DEL1 and DEL2 columns show the contribution to those model elements. All of the RX filter delay and most of the amplifier delay is generated in the filter elements, not the explicit delay elements. Cable delays are based on a velocity factor of 0.9, a reasonable estimate for low loss foam insulated coaxial cable.

3 ODE INTEGRATION

While there are many existing solvers for the numeric Ordinary Differential Equation (ODE) problem, they tend to be more general than needed, and don't fit well into the simple architecture chosen for this software. This architecture is purposely simple for two reasons: speed (1 ms simulations at 40 MHz sample rate involve 40000 sample steps) and embeddability. Chances are this first round of simulations will be grafted on to a VHDL or Verilog synthesizable model of the FPGA. While packaged mixed-signal simulation software is all the rage these days, I chose a known-good and *libre* framework from which to start, so that the results can be reproduced without recourse to proprietary black boxes. Besides, the mathematics of ODE integration are well understood.

Start with the first order differential equation for a single-pole low pass filter. This is expressed in Laplace form as

$$\frac{v_o}{v_i} = \frac{1}{s - p},$$

where v_i and v_o are the input and output signals, and p is the pole location. Make the differentiation explicit, and rearrange to get a form consistent with state-variable numerical ODE integration,

$$\frac{dv_o}{dt} = v_i + pv_o.$$

The simplest expression for a 'next' value v'_o in a discrete time *ansatz* is

$$v'_o = (1 + \Delta t \cdot p)v_o + \Delta t \cdot v_i.$$

To improve convergence properties in the case where $\Delta t \cdot p$ is not tiny, approximate the trajectory of v_i and v_o as linear within a single time step. Specifically, assume that v_o changes from v_o to v'_o , and v_i changes from v_i to v'_i .

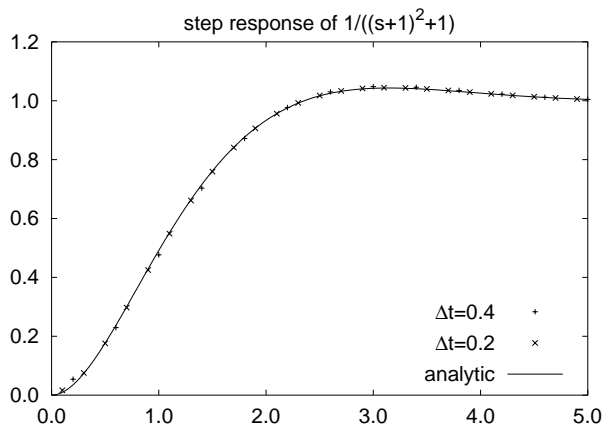


Figure 6: Cross-check of ODE code with analytic solution.

Using what is essentially the Trapezoidal Formula[5], our rendition of the discrete time approximation to the above differential equation becomes

$$\frac{v'_o - v_o}{\Delta t} = \frac{1}{2}(v_i + v'_i) + p\frac{1}{2}(v_o + v'_o) .$$

Solving for v'_o , we get

$$v'_o = av_o + \frac{1}{2}b(v_i + v'_i)$$

$$a = \frac{1 + \frac{1}{2}\Delta t \cdot p}{1 - \frac{1}{2}\Delta t \cdot p}, \quad b = \frac{\Delta t}{1 - \frac{1}{2}\Delta t \cdot p} .$$

This process is coded in c, and tested using a two-pole low pass Butterworth filter. Given the transfer function $1/((s + 1)^2 + 1)$, which has poles at $(-1,1)$ and $(-1,-1)$, the step response is $1 - e^{-t}(\sin x + \cos x)$, for $t > 0$. This analytically known response is plotted in figure 6 against two runs of the numeric ODE solver with different values for the time step. The first point after $t = 0$ has a relatively large error, effectively due to differing representations for the input signal near $t = 0$. After this point, the error is bounded by 1.6% for $|\Delta t \cdot p| = 0.57$, and by 0.4% for $|\Delta t \cdot p| = 0.28$. The largest $|\Delta t \cdot p|$ used in the real model is 0.45.

4 OPERATING THE CODE

The `step` program takes its numerical configuration from environment variables. The names below have a “OP_” prepended (compile-time adjustable) in an attempt to avoid namespace collisions. The table below summarizes the parameters the user has control of (without recompiling), along with their default values.

<i>name</i>	<i>default</i>	<i>short description</i>
DT	0.025	digital sample rate
NPT	2900	# of sample points
DEL1	17	cables plus triode
TRF1	-7.5,2.7 -7.5,-2.7	triode driver filter
TRF2	-9,0	triode output filter

CAV	-.075,+010	cavity filter
DEL2	12	cables plus ADC
RXF	-18,0 -9.5,-15 -9.5,15	receive 50 MHz filter
NOIS	0.003	rms ADC noise
BEAM	0,0.10	beam current vector
CLIP	5.0	triode clip harshness
T1	750	length of initial ramp
T2	1410	time of feedforward
DRV1	1.50,0.00	initial drive vector
DRV2	0.80,0.00	second drive vector
DRV3	0.00,-0.05	feedforward vector
SETP	0.74,0.0	feedback setpoint
GAIN	-6.93,-1.0	feedback prop. gain
IGRL	0.002	feedback integral term

Notes:

Delays must be integer, and are expressed in Δt units.

Vector quantities consist of the real and imaginary part, separated by a comma (but not spaces).

Filters are specified by listing their complex pole locations, in units of 1/time. The default setup uses μs for this and Δt . Multiple poles must be separated by spaces.

Input and output vectors always use scaled units, as discussed in the previous section.

When the triode clipping harshness parameter c is 0, no saturation curve is applied.

The beam current is configurable, but the timing model is not (yet). The timing model includes the 1 MHz chopper, and a ramp from 31% to 63% duty factor over 15 μs .

The program output (written to `stdout`) is nine columns wide. The first column is the time, the others come in pairs, for real and imaginary part. The four vectors, in order, are: FPGA output, triode output, cavity voltage, and receiver input.

5 PERFORMANCE

The amplifier and cavity output for default parameters is shown in figure 7. The analysis runs at the rate of about 16000 time steps per second on a 400 MHz Pentium-II computer. The transient in the imaginary component of the drive signal at $t=20 \mu s$ comes from the feedback loop correcting for a small amount (8°) of cavity detuning. A more complex digital section could reduce this transient by applying some form of feedback during the initial ramp. Such changes will only involve firmware.

The beam current will induce a voltage on the cavity, an effect known as beam loading. This voltage change will not (to first order) effect the bunching behavior of the cavity, but is phased to decelerate the beam. For this reason, I show the absolute voltage, instead of the angle relative to the bunching voltage. Figure 8 shows the scale of the effect for cavity 1, the “worst” of the four (higher TTF than cavities 2 and 3, lower stored energy than cavity 4). The normalized output of the `step` program has been multiplied by 101 to get actual, accelerating mode kV for the vertical scale, which is numerically the same as the beam energy change in keV. The sawtooth structure of the cav-

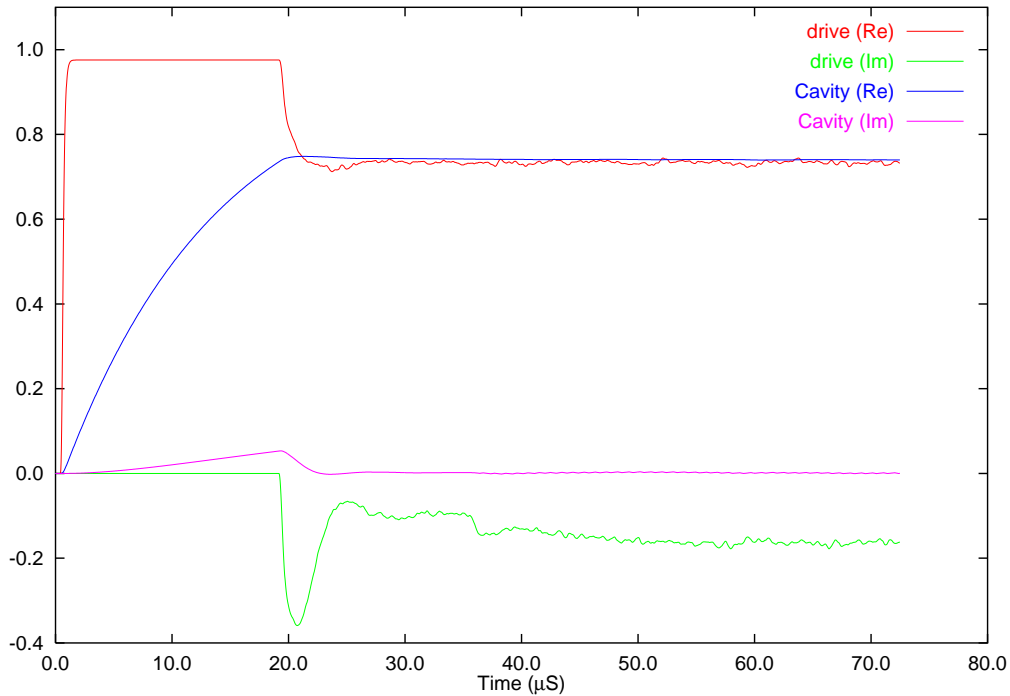


Figure 7: Beam pulse transient as modeled by default parameters.

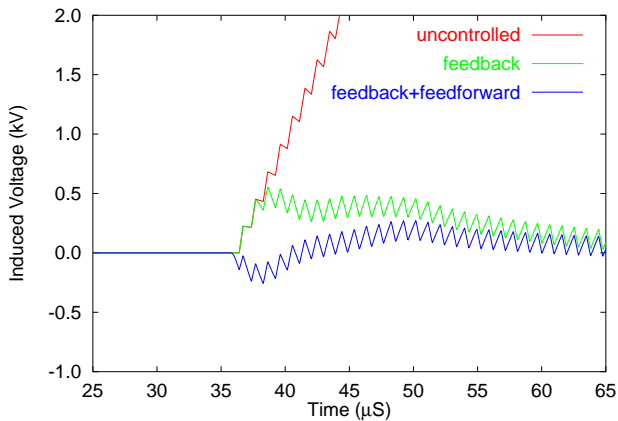


Figure 8: Beam pulse transient for cavity 1.

ity voltage is a result of the 1 MHz modulation of the beam current. Beam modeling results[6] suggest that SNS will run properly with induced energy changes in this cavity as large as about 1.3 keV (1° relative to the 75 keV bunching voltage).

6 CONCLUSIONS AND FUTURE WORK

The digital LLRF control architecture has the flexibility to meet the needs of the rebuncher cavities. Including simple proportional/integral feedback, the transient from beam loading can be kept smaller than the design tolerance of 1.3 keV. Feedforward of a single step can cut the peak of

that transient in half.

This model is not truly done, in that more detail can be added: quantization and digital clipping would probably come next. As the digital system is virtually assembled in VHDL or Verilog, this model can be attached so as to provide a complete system simulation.

The feedback parameters for this model were adjusted by hand. It is of considerable interest to write software that will adjust the control loop without human intervention.

7 REFERENCES

- [1] LLRF Review, A. Ratti, FE-EE-020, June 2001.
- [2] <http://recycle.lbl.gov/~ldoolitt/llrf-model/>
- [3] Rebuncher Cavity Design Update, Lawrence R. Doolittle, FE-EE-014, March 2000.
- [4] SNS Parameter List, SNS 100000000-PL0001-R05, May 2001
- [5] Formula 25.5.3 of Handbook of Mathematical Functions, Abramowitz and Stegun, 1964.
- [6] Front End Systems Beam Dynamics, J. Staples, LBNL-45289, January 2000.