Simulation and Characterization of an Exact Solvable Chaotic Oscillator and Matched Filter

by

John Phillip Bailey III

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 4, 2012

Keywords: chaotic oscillator, matched filter detection, SPICE

Approved by

Michael Hamilton, Chair, Assistant Professor of Electrical and Computer Engineering
Robert Dean, Associate Professor of Electrical and Computer Engineering
Thaddeus Roppel, Associate Professor of Electrical and Computer Engineering

Abstract

The simulation and characterization of a low frequency exact solvable chaotic oscillator and a corresponding matched filter is presented with discussion. The work builds on an existing design for the oscillator by realizing a version of the oscillator which performs analogously in both SPICE simulation and hardware for both shift-band and folded-band operation. Results from simulation and hardware are presented and compared to verify simulation accuracy.

SPICE simulation of the matched filter is also presented; its accuracy is verified by comparison to published theoretical behavior. This simulation is used to evaluate and characterize the frequency requirements of the matched filter's input. Use of the chaotic oscillator/matched filter pair in communication applications is supported by the results of this characterization.

Acknowledgments

The author would like to express great appreciation and thanks to Dr. Michael Hamilton for serving as his advisor and providing significant guidance and assistance throughout the process of performing this work. Dr. Robert Dean and Dr. Thaddeus Roppel are also thanked for their review and commentary.

Appreciation is expressed to Dr. Ned Corron and Dr. Daniel Hahs for their presentation on the mathematics behind the oscillator and matched filter and assistance with tuning of the oscillator. The author would also like to thank Aubrey Beal for his assistance in many aspects of this work.

A personal note of appreciation goes to John Bailey Jr. and Dawn Bailey for their support throughout the author's academic endeavors.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AC	Alternating Current
DC	Direct Current
FFT	Fast Fourier Transform
GIC	Generalized Impedance Converter
LF	Low Frequency
NIC	Negative Impedance Converter
op-amp	Operational Amplifier
PWL	Piece Wise Linear
SPICE	Simulation Program with Integrated Circuit Emphasis

Chapter 1

Introduction

Chaos has been long been regarded as an intriguing topic in the world of academia, but, traditionally, it has been concluded that its value was, at best, the satisfaction of academic curiosity. The power of modern computing has allowed chaos to develop from this relatively irrelevant status into a continuously growing field of its own. Recently proposed uses for chaos encompass many areas, including communication [3], radar [4] and even weather modeling [5].

To understand chaotic systems, it must first be understood that for a system to be considered chaotic, it must exhibit two characteristics: first, the system must have a positive Lyapunov exponent, i.e., trajectories the system visits which are near to each other must be exponentially divergent in value, and, second, these trajectories must be bounded in some manner. These characteristics often cause a chaotic system to appear as if its actions are random, but the utility of chaos comes from the fact that a chaotic system's behavior is in fact not random. It must be understood that this is due to the fact that the behavior of a chaotic system is highly dependent on its initial condition (the initial condition of a truly random process offers no insight into that process's behavior).

A mathematical description of a chaotic system consisting of a chaotic oscillator and a matched filter with potential practical application in both the fields of communications and radar is presented in Chapter 2. This thesis expands on the previously published work by introducing a SPICE simulation that is shown to be accurate when compared to actual hardware and theoretical results. Separate SPICE models have

been developed for the oscillator and matched filter; each model is thoroughly detailed through the presentation and analysis of simulation results.

Simulation was performed using LTSpiceIV, a SPICE simulator developed by Linear Technology. This simulator was chosen due to its ongoing developer support, its readily approachable schematic entry capabilities, and its continued free availability at `http://www.linear.com/designtools/software/`. The resulting netlists included in this work have also been found to function correctly using Ngspice release 24, which allows for Linux-based simulation. Other SPICE simulators, including the latest releases of PSPICE and HSPICE, were evaluated but found to be unsuitable for this work.

Hardware was constructed using standard components to match the oscillator model developed in SPICE. Characterization of the hardware's performance operating in the relevant bands was performed and is discussed. This hardware's performance reveals not only that its behavior matches theoretical expectation, but also that its behavior is closely modeled by the SPICE simulation.

The matched filter SPICE model was also used to evaluate the suitability of this chaotic oscillator and matched filter pair for communication applications. The output of the oscillator, which is also the input to the matched filter, was subjected to low-pass, high-pass, and band-pass filtering. Results from each of these cases are presented.

To expand on the results found through the use of band-pass-filtered input, which is expected to be the input necessary for communication, a simple circuit was added to the simulation to recover the desired waveform from the matched filter's output. This circuit is detailed and it is demonstrated that the circuit is able to correctly recover the desired waveform with band-pass-filtered input.

Chapter 2

Previously Published Work

2.1 Exact Solvable Chaotic Waveform Generation

Because they are often associated with complex behavior, chaotic systems are typically assumed to lack exact analytic solutions; this assumption has, however, been disproven [6] [7] [2]. Building on this work, a differential equation describing an exactly solvable chaotic system was recently presented [2]. The chaotic nature of this system is verified by the attractor visited by the system's resemblance to the well-known strange attractor first described by Lorenz [8]. Figure 2.1 provides an example of this system's attractor (right) as well as a folded-band attractor. The differences between these attractors will be discussed in Section 2.1.1.

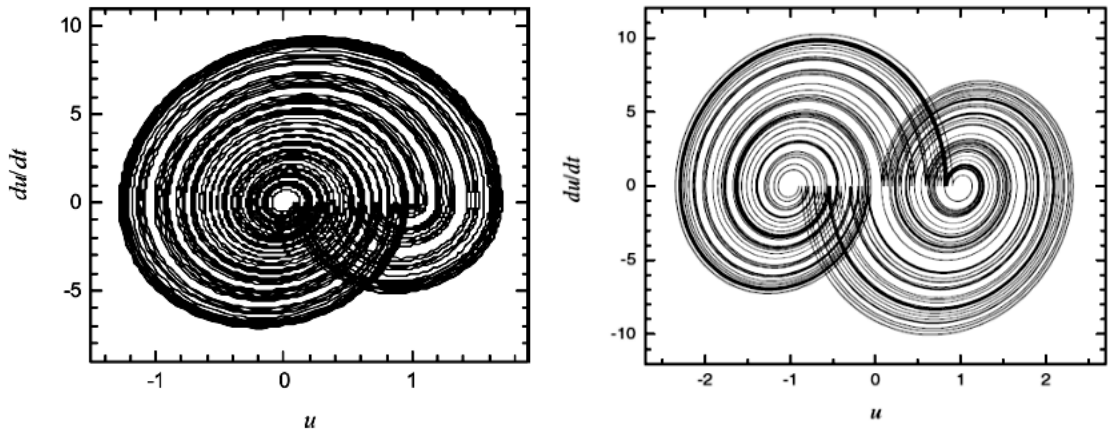

Figure 2.1: Folded-band (left) and shift-band (right) attractors [1] [2]

The trajectories for the bakers map and the shift map of this differential equation can be represented by the convolution of an acausal basis function with a random

process [9]. Because of this, the differential equation's solution can be described as a linear representation. The matched filter discussed in this thesis is possible due this linearity.

2.1.1 Continuous-time/Discrete-time Hybrid

The system described in [2] consists of a continuous-time state $u(t) \in \Re$ which provides the necessary exponential divergence and a discrete-time state $s(t) \in \{\pm 1\}$ which maintains boundedness. This state places the system in the shift-band (see Figure 2.1). A similar system described in [1] operates in the folded-band due to its discrete-time state $s(t) \in \{0, 1\}$. In both systems, the continuous-time differential equation is

$$\frac{d^2u}{dt^2} - 2\beta\frac{du}{dt} + (\omega^2 + \beta^2) * (u - s) = 0 \tag{2.1}$$

where $\omega = 2\pi$ for both bands, $0 < \beta \leq \ln 2$ for the shift-band, and $\beta > 0$ for the folded-band. The discrete-time state can experience a switching event only when the derivative of the continuous-time equation is zero - this occurs every $1/2t$. This switching event is governed by

$$\frac{du}{dt}(t) = 0 \rightarrow s(t) = sgn(u(t)) \tag{2.2}$$

where

$$sgn(u) = \begin{cases} -1 & u < 0 \\ +1 & u \geq 0 \end{cases} \tag{2.3}$$

when operating in the shift-band and

$$\frac{du}{dt}(t) = 0 \rightarrow s(t) = H(u(t) - 1) \tag{2.4}$$

where

$$H(u) = \begin{cases} 0 & u \leq 1 \\ 1 & u > 1 \end{cases} \tag{2.5}$$

when operating in the folded-band. Figure 2.2 illustrates the continuous-time state leading up to, during, and after a switching event in the discrete-time state in folded-band operation.

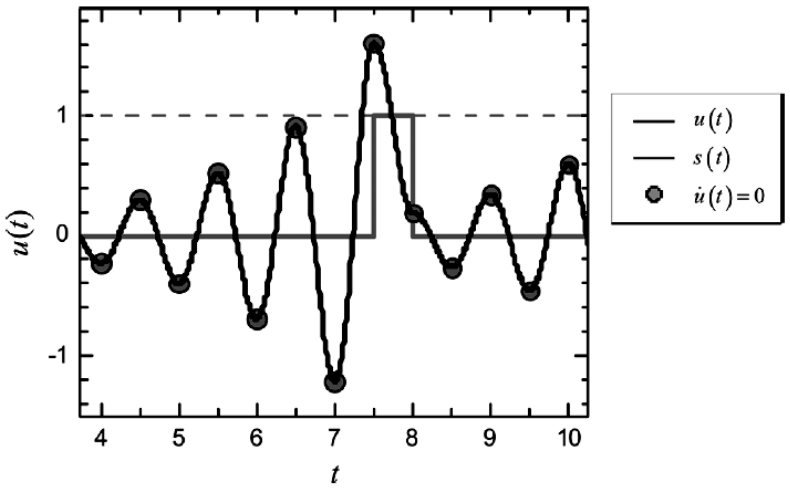

Figure 2.2: Illustration of continuous-time and discrete-time states [1]

As the oscillation grows, each relative maximum and minimum triggers the condition in (2.5); for the first seven events, however, the magnitude of $u(t)$ is not large enough to cause a change in $s(t)$. The eighth event sees the magnitude $u(t)$ large enough to cause a $1/2t$ duration switch in $s(t)$ that bounds the growth of $u(t)$. Operation in the shift-band is analogous, with $s(t)$ switching between ± 1.

2.1.2 Solution

For an initial condition u_n with $\frac{du_n}{dt} = 0$, which places the system at a switching event, an exact solution in the shift-band can be described by the superposition of

fixed basis functions given by $P(t)$

$$u(t) = \sum_{m=-\infty}^{\infty} \sigma_m * P(t-m) \tag{2.6}$$

where

$$P(t) = \begin{cases} [1-\exp(-\beta)] * \exp(\beta t) * [\cos\omega t - \frac{\beta}{\omega}\sin\omega t] & t < 0 \\ 1 - \exp(\beta(t-1)) * [\cos\omega t - \frac{\beta}{\omega}\sin\omega t] & t = 0 \\ 0 & t > 0 \end{cases} \tag{2.7}$$

and σ_m represents the values of $s(t)$ [2]. For the folded-band, the initial condition remains the same, but the basis function is described by $Q(t)$, yielding

$$u(t) = \sum_{m=0}^{\infty} \sigma_m * Q(t - t_n - \frac{m}{2}) \tag{2.8}$$

where

$$Q(t) = \begin{cases} [1+\exp(-\frac{\beta}{2})] * \exp(\beta t) * [\cos\omega t - \frac{\beta}{\omega}\sin\omega t] & t < 0 \\ 1 + \exp(\beta(t-\frac{1}{2})) * [\cos\omega t - \frac{\beta}{\omega}\sin\omega t] & 0 \le t < \frac{1}{2} \\ 0 & \frac{1}{2} \le t \end{cases} \tag{2.9}$$

and σ_m again represents the values of $s(t)$ [1].

2.1.3 Dynamics

For the shift band, (2.6) evaluated at t_n yields an inverse coding function

$$u_n = (1 - \exp(\beta)) \sum_{m=0}^{\infty} \sigma_{m+n}(\exp(-m\beta)) \tag{2.10}$$

for which a sequence of satisfactory $\sigma_m \in \{\pm 1\}$ exists. Analogously, the folded-band solution (2.8) evaluated at t_n yields an inverse coding function

$$u_n = (1 + \exp(-\frac{\beta}{2})) \sum_{m=0}^{\infty} \sigma_m (-\exp(-\frac{\beta}{2}))^m \tag{2.11}$$

for which a sequence of satisfactory $\sigma_m \in \{0, 1\}$ exists. The utility of this function comes from its generation of an amplitude sequence $\{\sigma_0, \sigma_1, ..., \sigma_m\}$ dependent on the u_n initial condition. By controlling u_n, it is possible to generate a desired σ_m sequence which can then be mapped to symbols. This technique is described in [3] and [10] where it is also detailed how the adjustment required to generate a symbol decreases as the distance in the future that the symbol is desired increases. This control has not yet been implemented, but it is expected that its implementation will use folded-band operation exclusively; to this end, shift-band control will not be discussed.

A relationship relating the successive u_n maxima can be derived, thus allowing $u(t_n + 1/2)$ to be expressed in terms of u_n and s_n. From this, the tent map in Figure 2.3 can be constructed.

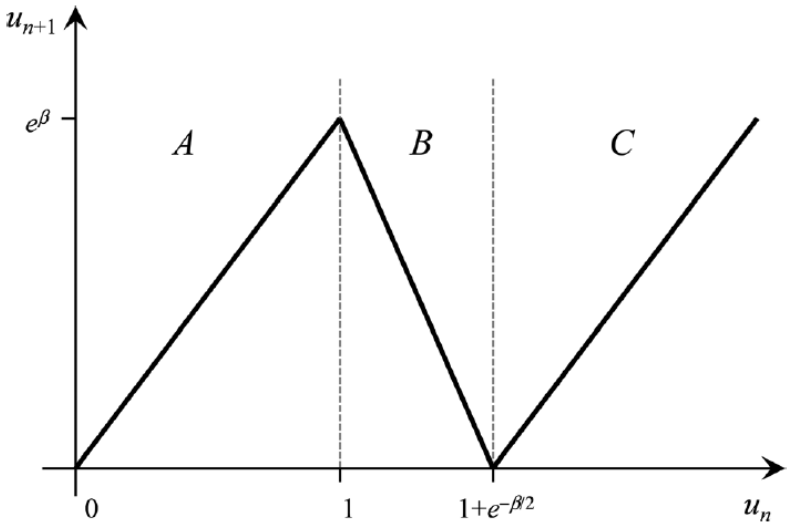

Figure 2.3: Folded-band Tent Map [1]

As shown, three symbols - `A`, `B`, and `C`, - are chosen corresponding to the three separate regions on the map. The spacing between u_n values, also known as the return time, is not uniformly spaced. Previous values of $s(t)$, which occur every $1/2t$, determine which symbol is generated. These are mapped as `A`→`00` `B`→`100` `C`→`10`; it is expected that `C` will not be used for simplification of the receiving system design. To generate an arbitrary symbol sequence, the required u_n can be determined with the inverse coding function by inserting the amplitude sequence corresponding to the desired symbol sequence. For example, the symbol sequence {`B A B B A A`} would map to the amplitude sequence {`100001001000000`}, which would then be used for σ_m values in the inverse coding function. It is important to note that a practical implementation of this coding would be restricted by a yet-to-be-developed grammar, such that only sequences of `A` and `B` contained in that grammar could be used.

2.1.4 Control

Due to the need to adjust u_n, the equation describing continuous-time portion of the circuit must be modified slightly to include a control input h, resulting in

$$\frac{d^2u}{dt^2}[1-(2\beta-GH(u-h))]+(\omega^2+\beta^2)(u-s)=0 \tag{2.12}$$

where H is defined in (2.5) and G $\in \{\Re \gg 1\}$ [11]. Whenever $u > h$, charge is rapidly removed from the system until $u < h$, at which point the controlling mechanism is stopped. With this type of control, only small corrections are necessary after initial transients have settled.

2.2 Matched Filter

Detection of the chaotic waveform can be performed by a simple matched filter which has a time-reversed impulse response to that of the waveform to be detected.

This is described by the equations

$$\frac{d\eta}{dt} = v(t+1) - v(t) \tag{2.13}$$

for the shift-band,

$$\frac{d\eta}{dt} = v(t+\frac{1}{2}) - v(t) \tag{2.14}$$

for the folded-band, and

$$\frac{d^2\xi}{dt^2} + 2\beta\frac{d\xi}{dt} = (\omega^2 + \beta^2) * \eta(t) \tag{2.15}$$

where v is the input, ξ is the output, and η is an intermediate state which is defined in Section 2.3.2 [2].

2.3 Chaotic System Realization

Theoretical circuits built from standard components have been developed for both the chaotic waveform generator and the corresponding matched filter. While they are not the exact designs used for the work discussed in this thesis, these circuits provide the foundation on which this work builds.

2.3.1 Oscillator

The realization of the chaotic waveform generator takes the form of a chaotic oscillator, whose schematic is shown in Figure 2.4.

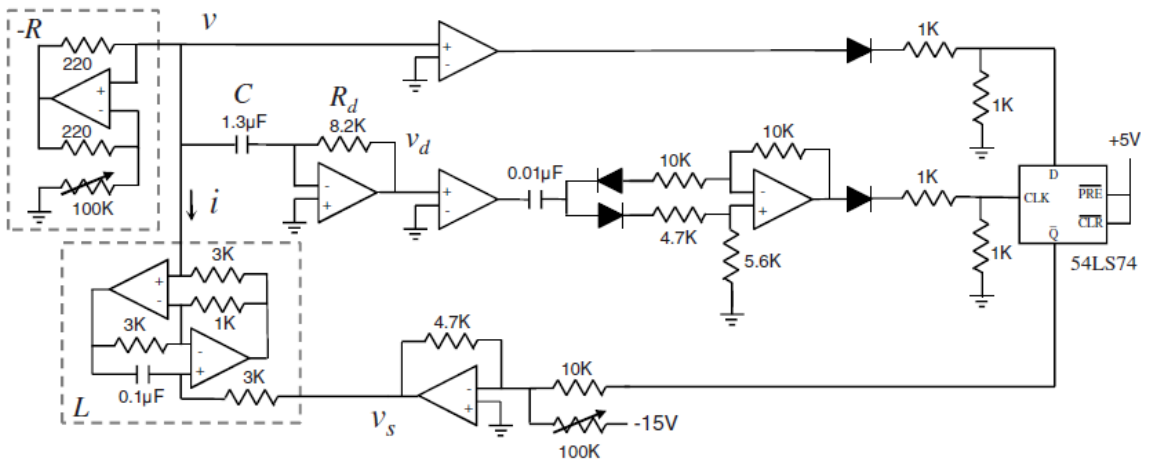

Figure 2.4: Published Chaotic Oscillator Schematic [2].

Oscillation is realized through the use of a RLC oscillator with negative resistance (the active circuitry required to realize the -R and L components is discussed in Chapter 3), which can be modeled by writing the standard RLC differential equations

$$C\frac{dv}{dt} - \frac{v}{R} + i = 0 \tag{2.16}$$

and

$$L\frac{di}{dt} = v - v_s \tag{2.17}$$

Assuming $\tau = \frac{t}{T}$ yields

$$T = 2RC\omega\sqrt{\frac{L}{4R^2C - L}} \tag{2.18}$$

where T represents the oscillator's return time. (2.16) combined with (2.17) results in

$$\frac{d^2v}{d\tau^2} - 2\beta\frac{dv}{d\tau} + (\omega^2 + \beta^2)(v - v_s) = 0 \tag{2.19}$$

where

$$\beta = \frac{T}{2RC} \tag{2.20}$$

This portion of the circuit provides the necessary exponential divergence and is then bounded by feedback from the additional circuitry which triggers at each $\frac{du}{dt} = 0$ point.

2.3.2 Matched Filter

The matched filter is also realized through the use of a RLC circuit with additional supporting circuitry. Shown in Figure 2.5, its left section implements (2.13) and (2.14) while its RLC section implements (2.15).

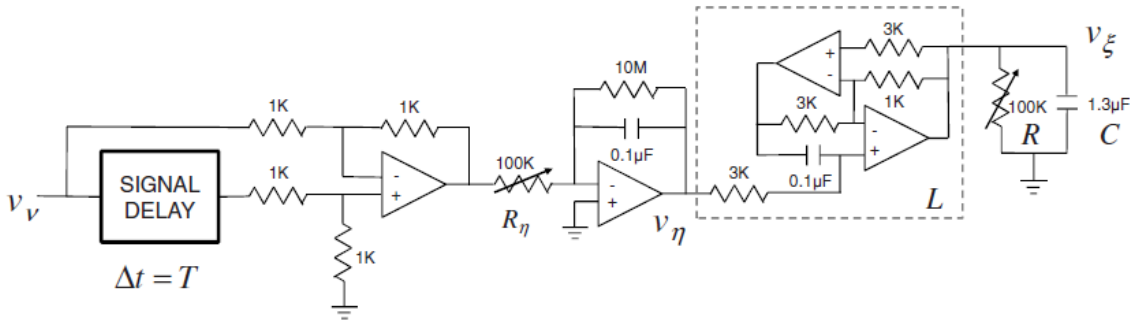

Figure 2.5: Published Chaotic Matched Filter Schematic [2].

The values for the R, L, and C components are the same as those used in the oscillator, with the distinction that the R value is positive. The value for R_η is calculated using

$$R_\eta = \frac{T}{0.1\ \mu\text{F}} \tag{2.21}$$

where T is defined in (2.18). η, the intermediate state introduced in (2.13) and (2.14), can be defined as

$$\eta = \frac{-V_\eta(t - \Delta t)}{V} \tag{2.22}$$

Chapter 3

Oscillator

3.1 Implemetation

As in the published design, the implementation of the chaotic oscillator uses an RLC oscillator combined with additional circuitry to provide bounding; these blocks will be referred to as the continuous-time section and the discrete-time section respectively. To best explain their operation, these blocks are examined separately in the proceeding sections. A combined schematic can be found in Appendix A.

3.1.1 Continuous-time Section

The function of the continuous-time section, shown in Figure 3.1, continues to be provision of the exponential divergence necessary for chaos.

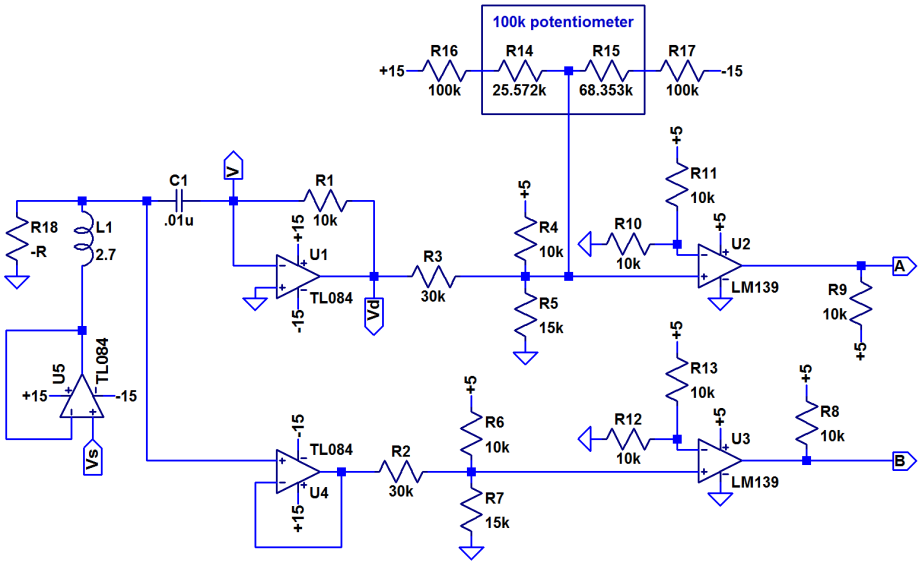

Figure 3.1: Continuous-time Section Schematic

Because both the negative resistance and large inductance values required for this implementation are not possible with standard components, active components are used in their place.

Negative Impedance Converter

A negative resistance is realized through the use of a NIC as shown in Figure 3.2.

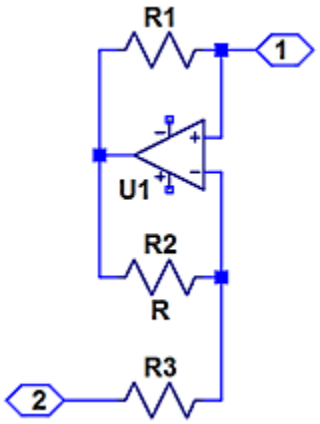

Figure 3.2: NIC Schematic

Resistance seen between nodes `1` and `2` is described as

$$R = -R1\frac{R3}{R2} \tag{3.1}$$

where, for this application, the value of the negative resistance must be adjustable; this is accomplished by replacing R3 with a 100 kΩ potentiometer [12].

Generalized Impedance Converter

Although a large inductance is physically possible with a passive element, a GIC is used in this circuit to minimize space requirements. Figure 3.3 shows an implementation of this element.

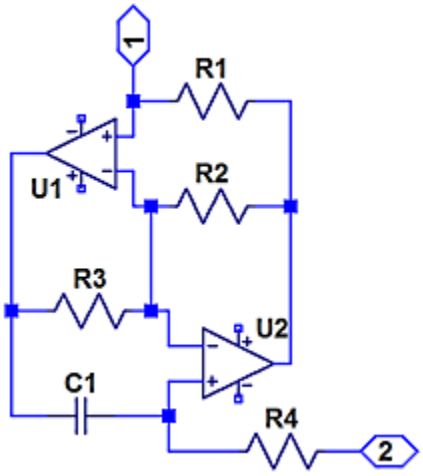

Figure 3.3: GIC Schematic

Equations 3.2 and 3.3 are used to determine the appropriate values for the passive elements [13],

$$L_{eq} = \frac{R_1 R_3 R_4 C_1}{R_2} \tag{3.2}$$

$$R_4 = \frac{R_2 L_{eq}}{R_1 R_3 C_1} \tag{3.3}$$

where values for R1, R2, R3, R4, and C1 are 3.01 kΩ, 1 kΩ, 3.01 kΩ, 3.01 kΩ, and 0.01 μF respectively. This combination yields and effective inductance of 0.273 H seen between nodes `1` and `2`.

Node `V`, generated by the RLC oscillator, corresponds to the continuous-time state $u(t)$. Bounding for `V` is provided from the discrete-time section in the form of a voltage at node `Vs`. The top branch of the continuous-time circuit includes a differentiator which provides node `Vd`, corresponding to $\frac{du}{dt}$. `Vd` is offset by an amount adjustable by the top potentiometer so that its zero crossings occur at the constant voltage supplied to the negative input to the comparator. Appendix C describes this tuning process in its discussion of tuning the overall circuit. The result of this comparison becomes node `A`, which is high when `Vd` is positive, low when `Vd` is negative, and switching states when `Vd` is zero. The bottom branch compares `V`

to the same constant voltage used in and has the same output behavior as the top branch. This branch does not need to be tuned as no operations are performed on its input. Both `A` and `B` are shifted to appropriate voltage levels to be processed by the discrete-time section.

3.1.2 Discrete-time Section

As shown in Figure 3.4, the discrete-time section of the circuit operates on nodes `A` and `B` to produce the discrete-time state $s(t)$, which corresponds to node `S`.

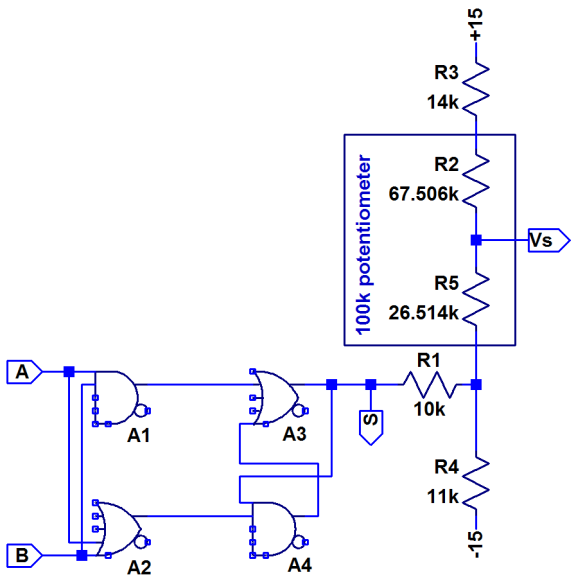

Figure 3.4: Discrete-time Section Schematic

The behavior of the logic blocks is detailed in Table 3.1:

A	**B**	**A1**	**A2**	**A3**	**A4**	**S**
L	L	L	L	A4	L	L
L	H	L	H	A4	A3	A3
H	L	L	H	A4	A3	A3
H	H	H	H	H	A3	H

Table 3.1: Discrete-time Circuit States

where the values A1, A2, A3, and A4 represent the output levels of their respective gates.

From this table, it can be shown that `S` is forced low whenever `B` is low, and is forced high after a slight delay (dependent on the characteristics of the logic components used) `B` going high. `A` acts as a clock to restrict changes in `S` to $1/2t$ intervals. Figure 3.5 demonstrates this behavior.

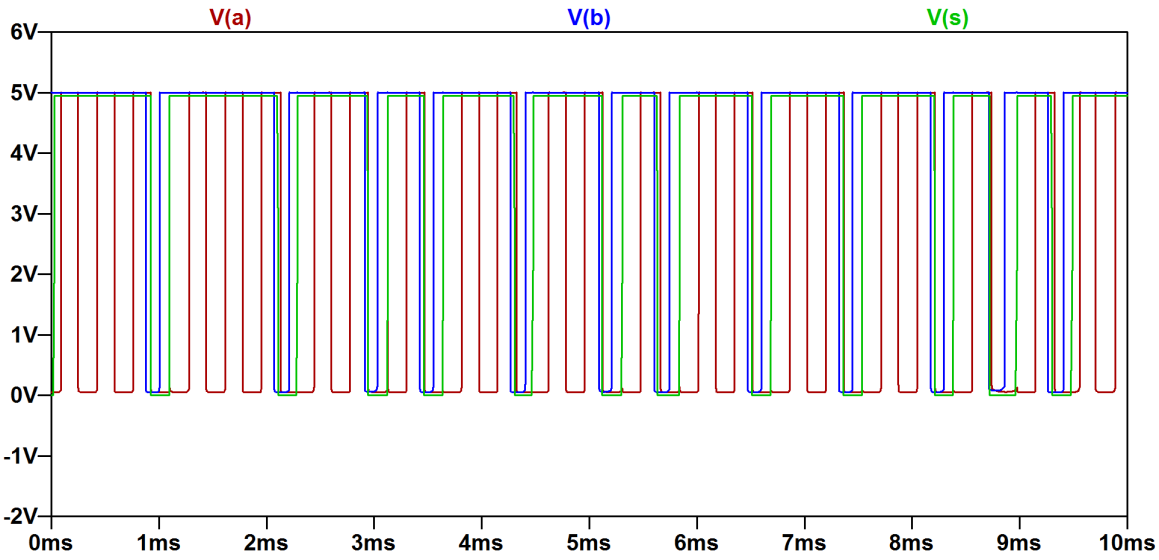

Figure 3.5: Logic Behavior

`S` is tuned by the right potentiometer to either be centered at zero (for shift-band operation) or to vary between zero and a higher voltage (for folded-band operation), resulting in the output node `Vs`. This tuning is also discussed in Appendix C.

3.2 Simulation

LTSpiceIV's schematic entry was used to create a netlist for the oscillator circuit with guidance from [14]; all data presented in this thesis was generated with the latest publicly available version of LTSpiceIV as of June 12, 2012. The overarching goal of this work was to model as closely as possible the behavior of the oscillator hardware. To assist with this, SPICE models for the TL084 op-amps and the LM339

comparators, from [15] and [16], respectively, were used. Exact SPICE models were not available for the 74HC08 and 74HC32 logic gates, so LTSpiceIV's included logic models were used with parameters from [17]. While these gates are able to be operated with a Vcc higher than 5 V, setting the SPICE model's output to any value greater than 4.95 V resulted in the simulator failing to converge, so this value was used in both the simulation and the hardware. The potentiometers used in the oscillator hardware were also not able to be directly mapped to SPICE components; for these, a combination of two resistors (or a single resistor when operating in rheostat mode) were used. Values of the hardware potentiometers were taken and then refined through the use of LTSpiceIV's parameter sweep functionality to determine final values used in the simulation.

Transient analysis was used to calculate results with the command `.tran 0 30ms .5ms .1ms startup uic`. The startup and uic parameters are of particular note due to their necessity for reliable convergence [18] - without these parameters, the simulation would fail with a singular matrix error. All data in this section was generated starting at 0.5ms (to avoid displaying the initial transients) with a run time of 30ms and a minimum step size of 0.1ms. Both this schematic and the resulting netlist are provided in Appendix A. Evaluation of the simulation's performance in both the shift-band and folded-band follows.

3.2.1 Shift-band Performance

Figure 3.6 demonstrates the simulation's performance when tuned to operate in the shift-band.

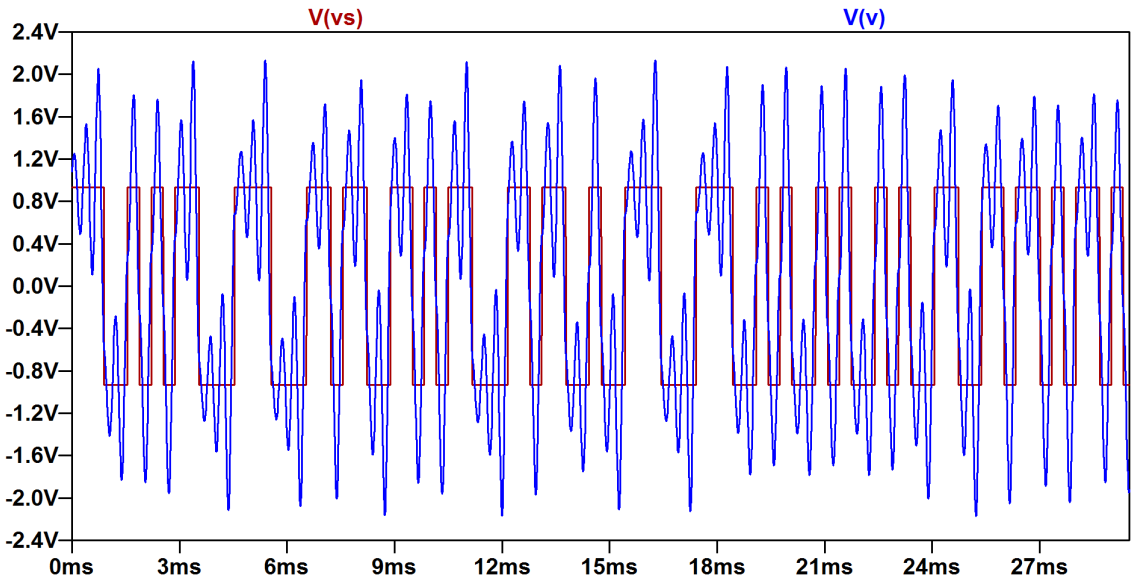

Figure 3.6: LF Oscillator Simulation Shift-band `V` and `Vs` vs. t

As expected, `Vs` switches between approximately -1 and 1 V each time `V` exceeds the magnitude of the set switching condition (`Vd` is not plotted for clarity). Operation in this band is expected to cause the oscillations of `V` to switch their center between the high and low levels of `Vs` each time `Vs` switches; the presented results verify this expectation.

The strange attractor, introduced in Figure 2.1, is also recreated in the simulation by plotting `Vd` on the Y axis and `V` on the X axis as shown in Figure 3.7.

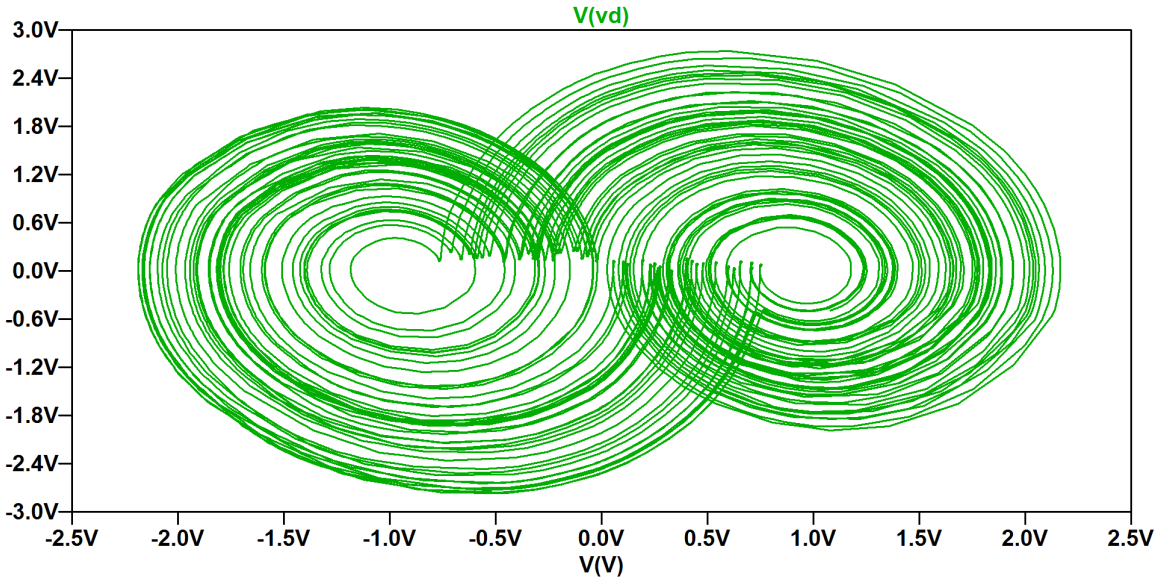

Figure 3.7: LF Oscillator Simulation Shift-band `Vd` vs. `V`

It is readily apparent that the centers of the attractor are correctly located at the low and high levels of `Vs`.

For better comparison to its hardware counterpart, LTSpiceIV's built-in FFT function is used to display the frequency content of `V`. Figure 3.8 displays the result of this transform.

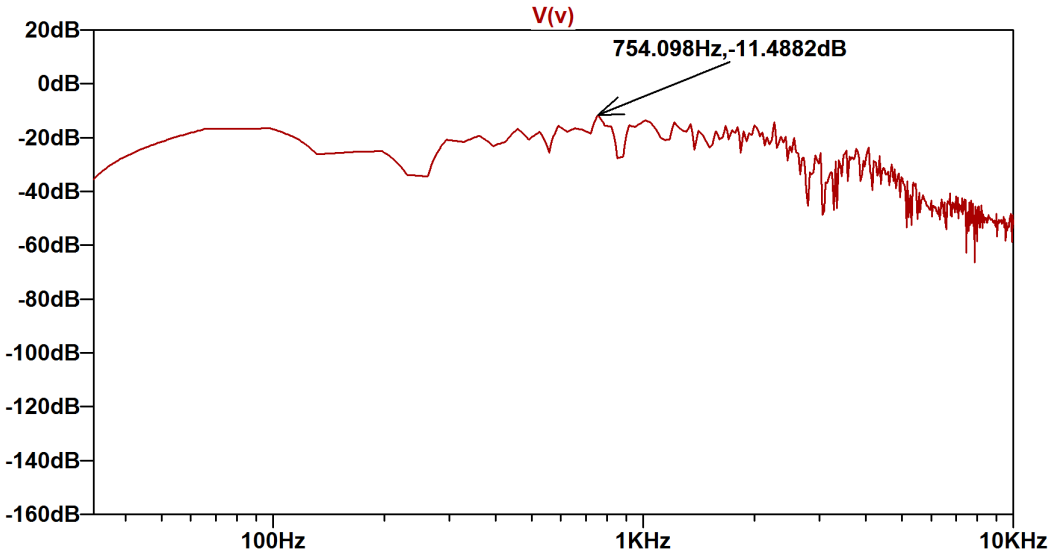

Figure 3.8: LF Oscillator Simulation Shift-band FFT(V)

The frequency content approaches even distribution from DC to approximately 2 kHz with its maximum occurring at 754 Hz. From this figure, it can be seen that this relatively even distribution allows V to effectively behave in a manner similar to that of a spread-spectrum signal without any modulation having been applied; previous work on spread-spectrum techniques confirms that a chaotic signal inherently behaves in a spread-spectrum manner [19].

3.2.2 Folded-band Performance

Figure 3.9 demonstrates the simulation's performance when tuned to operate in the folded-band.

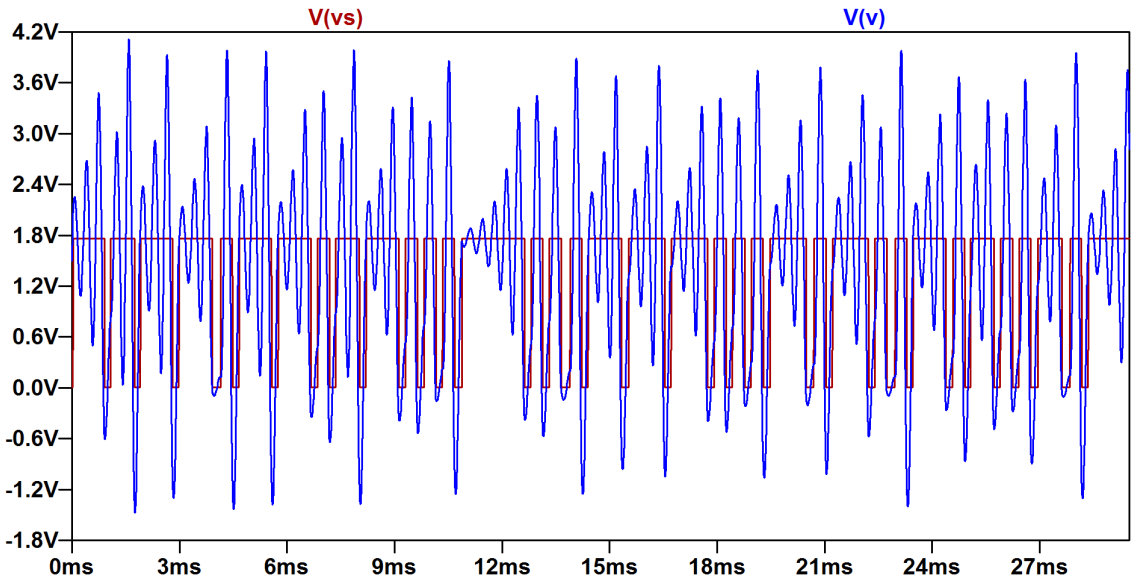

Figure 3.9: LF Oscillator Simulation Folded-band `V` and `Vs` vs. t

In this band, `Vs` also performs as expecting by switching between approximately 0 and 1.8 V. Because the sign of `Vs` does not change, the oscillation of `V` is always centered at the high value of `Vs`.

The strange attractor for folded-band operation is generated by the plot in Figure 3.10 and correctly has its centers at zero and the high level of `Vs`.

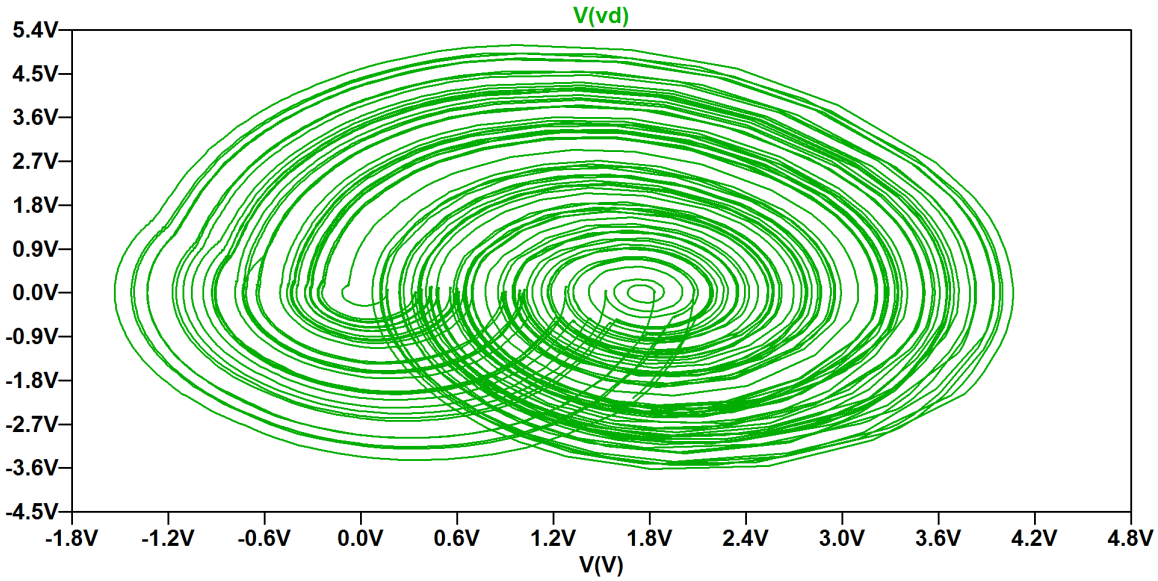

Figure 3.10: LF Oscillator Simulation Folded-band `Vd` vs. `V`

The frequency content of `V`, shown in Figure 3.11, has a higher maximum, 2.19 kHz, than the peak in the shift-band, but still maintains a similar approximately even frequency distribution from DC to 3 kHz. Because of this frequency content, the folded-band oscillator also exhibits spread-spectrum behavior.

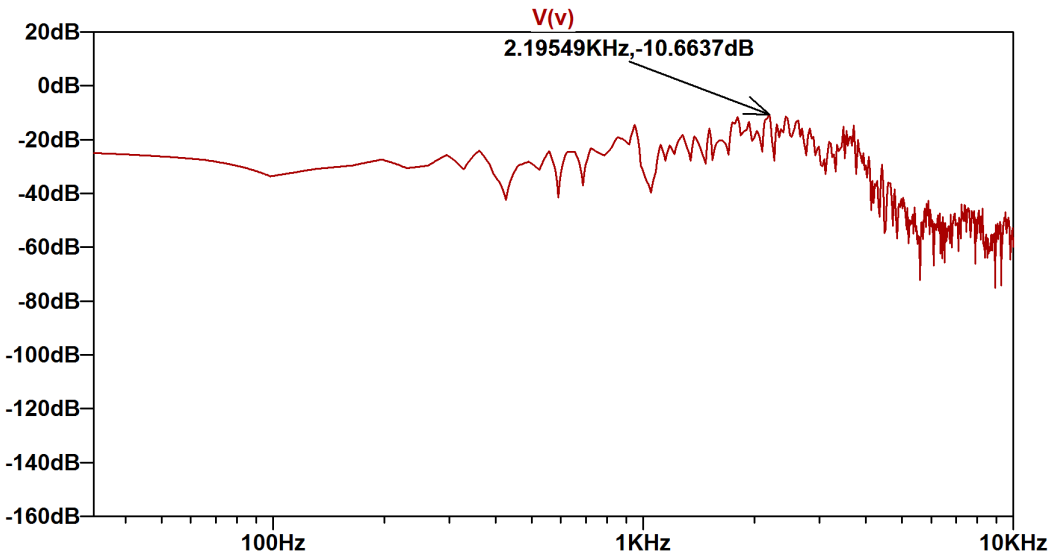

Figure 3.11: LF Oscillator Simulation Folded-band FFT(V)

3.3 Hardware

A hardware version of the chaotic oscillator has been built with a strong emphasis on matching the circuit described in the simulation. As with the previously published circuit, standard readily-available components were used and assembly was performed on a solderless breadboard. A picture of the functional hardware is provided in Figure 3.12, and a full parts list can be found in Appendix A.

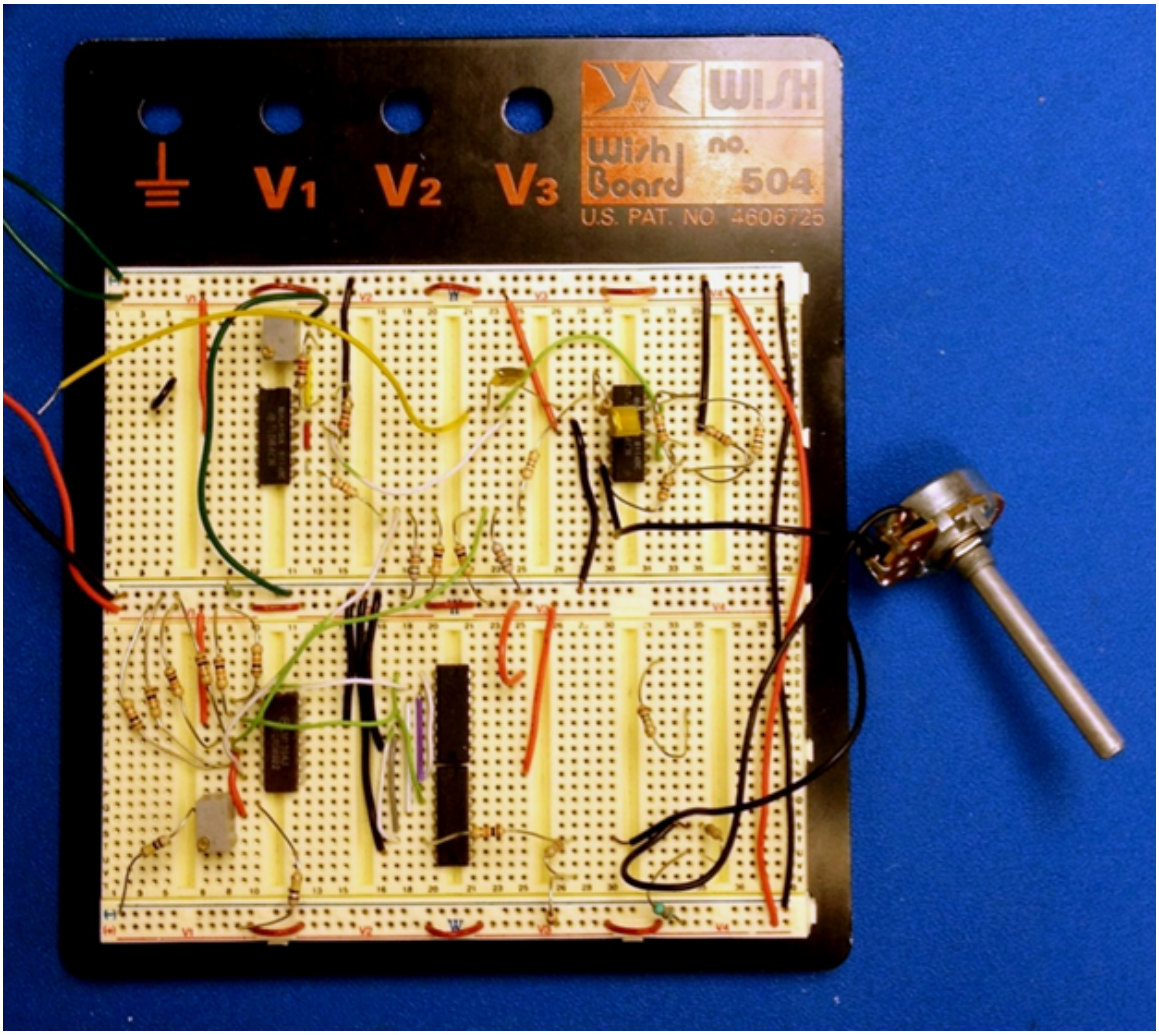

Figure 3.12: LF Chaotic Oscillator Hardware Implementation

The large potentiometer seen in Figure 3.12 was necessary to allow easy tuning of `Vs`; as the band of operation is set largely by this potentiometer's value, fine control was required for proper operation. All measurements were taken on an Agilent DSO-X 3034A Oscilloscope and power was supplied by an Agilent E3631A Triple Output DC Power Supply. Using a single power supply eased the process of power cycling the circuit, which was necessary if potentiometer value adjustment caused the circuit to become stuck in a non-operational state. Power rails were set to +15 V, -15 V, and 4.95 V, with all elements sharing a common ground, and voltage levels were measured using Agilent 10:1 probes referenced to the circuit ground. The Y axis and X axis

scales used in this section were adjusted between measurements for clarity; for this reason, the settings used for each measurement are visible.

3.3.1 Shift-band Performance

To stay in line with the simulation focus of this work, hardware performance discussion is limited to a comparison with simulated results. In shift-band operation, Figures 3.13, 3.14, and 3.15 verify that the oscillator simulation very closely describes the actual hardware.

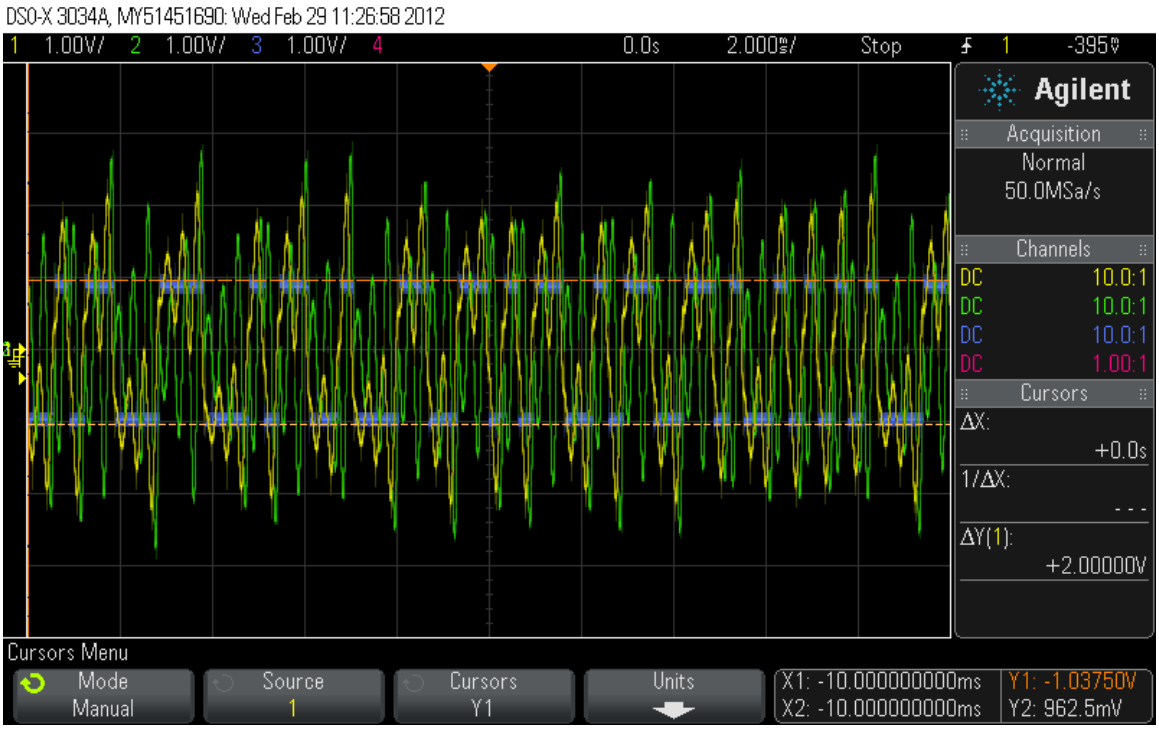

Figure 3.13: LF Oscillator Hardware Shift-band `V` and `Vs` vs. t

`Vs` was measured to be slightly offset towards -1 V, which can be attributed to the limited precision afforded by the single-turn (but easily adjustable) potentiometer used in its generation.

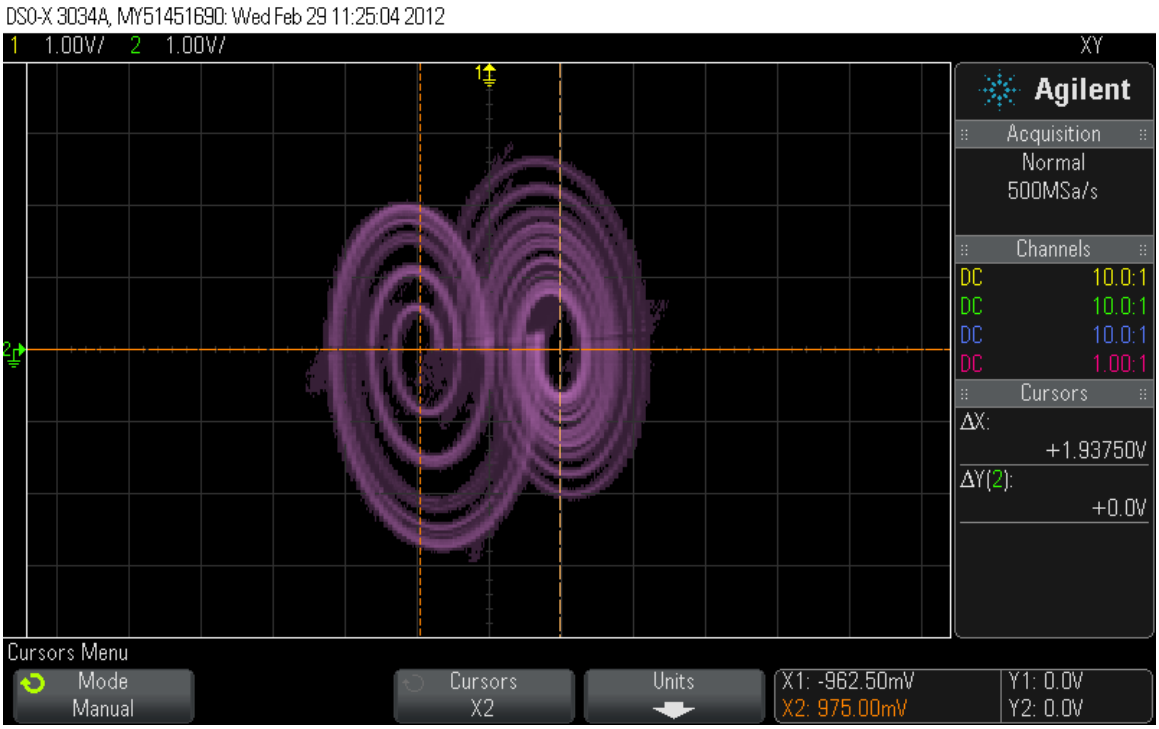

Figure 3.14: LF Oscillator Hardware Shift-band `Vd` vs `V`

The magnitude of the approximate locations of the strange attractor's centers are measured to differ by 12.5 mV; as these locations are determined by placing the oscilloscope cursors manually, this difference is small enough to reasonably consider their locations equal.

Figure 3.15: LF Oscillator Hardware Shift-band FFT(V)

Using the oscilloscope to calculate FFT(V) results in a peak frequency of 760 Hz - 6 Hz away from the simulated peak of 754 Hz - and a spread-spectrum like frequency content. From the combination of these results, it can be shown that the shift-band simulation provides an excellent approximation of the shift-band hardware.

3.3.2 Folded-band Performance

As expected, Figures 3.16, 3.17, and 3.18 demonstrate that the oscillator also matches the simulation closely in folded-band operation. For this band, hardware and simulation are found to correspond even more closely than in the shift-band.

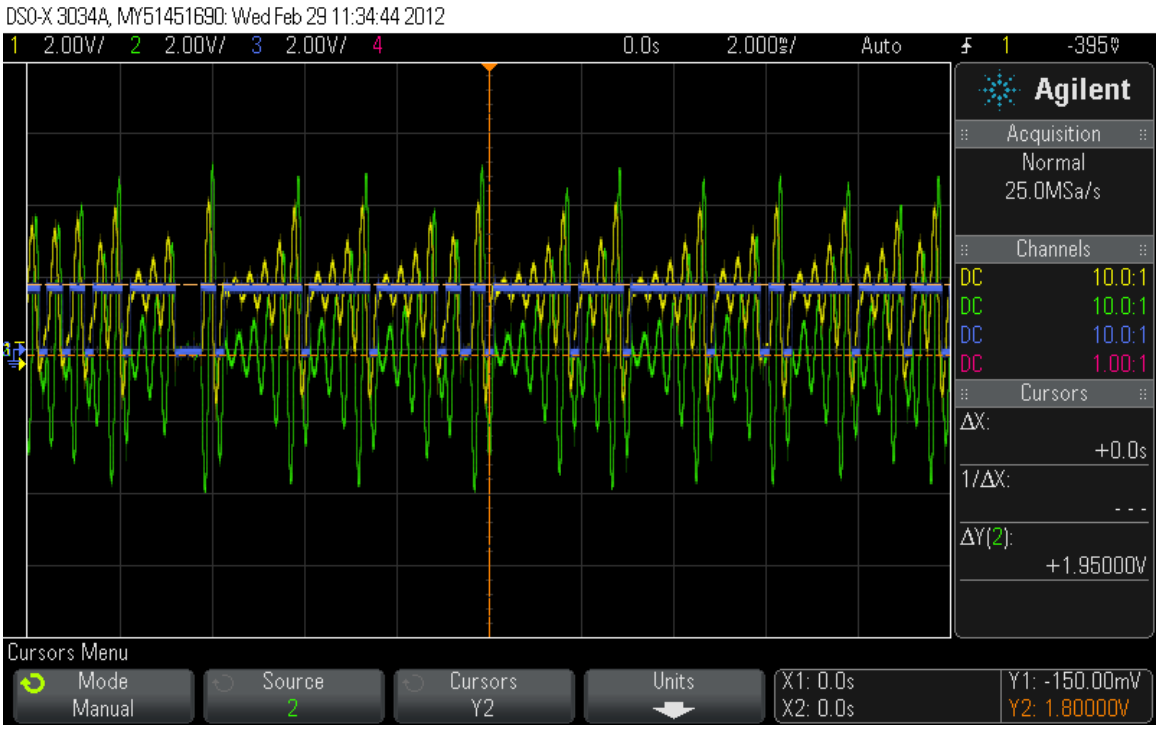

Figure 3.16: LF Oscillator Hardware Folded-band `V` and `Vs` vs. t

When operating in the folded-band, it was possible to match the levels of `Vs` to approximately equal those seen in simulation.

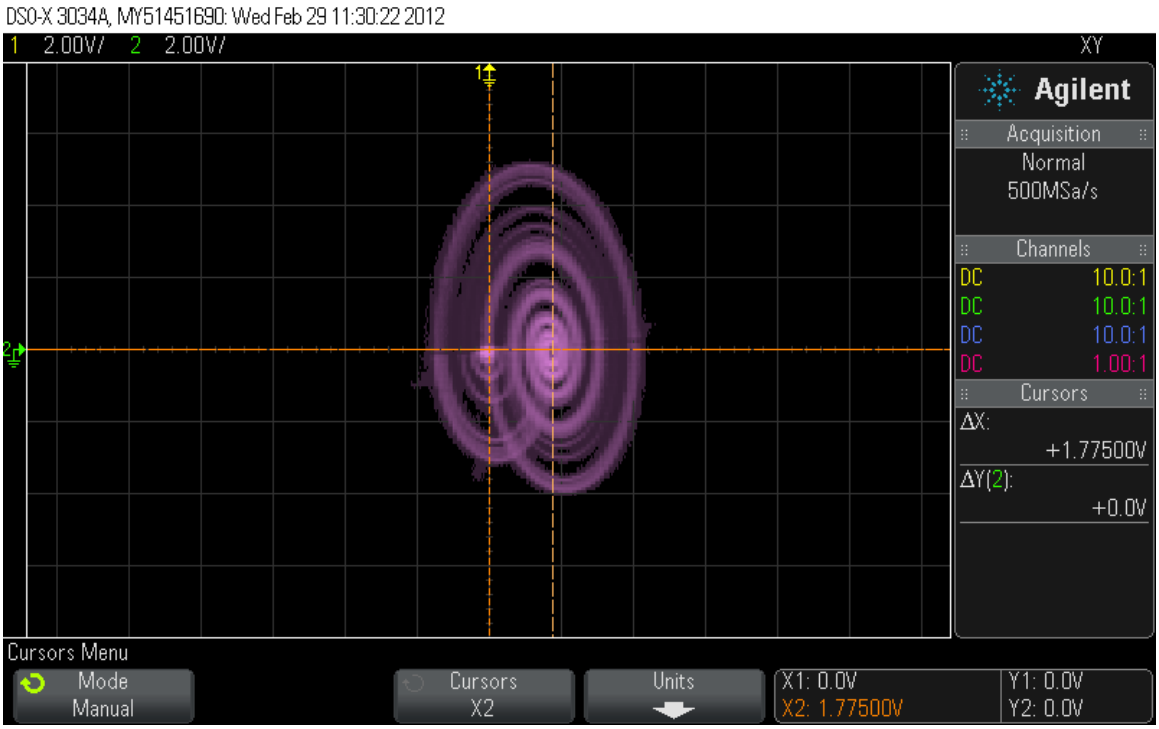

Figure 3.17: LF Oscillator Hardware Folded-band `Vd` vs `V`

As in the shift-band, it is not possible to to get exact locations for the attractor's centers, but the best estimates presented in Figure 3.17 show centers that are within 0.005 V of the simulation.

Figure 3.18: LF Oscillator Hardware Folded-band FFT(V)

The FFT(V) results continue this close correspondence - the measured peak frequency of the hardware oscillator differs from that of the simulated oscillator by less than 10 Hz and again behaves in a manner similar to that of a spread-spectrum modulated signal. As expected, the folded-band hardware is also approximated closely by the folded-band simulation. The culmination of these comparisons demonstrates that the SPICE model simulation can be used to evaluate hardware performance with a high level of confidence in the validity of the results.

Chapter 4

Matched Filter

4.1 Implementation

The matched filter has been implemented as presented in Figure 2.5 with some component values changed to reflect their corresponding values in the oscillator implementation. The L component is again realized through the use of a GIC, but as the R in the filter requires an adjustable positive value, its realization requires only a 100k potentiometer. Using (2.18) and (2.21) yields a required signal delay of 0.3299 ms and a required R_η value of 3.299 kΩ.

4.2 Simulation

Schematic entry in LTSpiceIV was again used to generate the matched filter circuit's netlist. Using the TL084 op-amp, however, lead to a simulation which would not converge; to solve this issue, the LT1001 op-amp was used in its place. For this application, the LT1001 acts as a drop-in replacement and it is expected that a hardware implementation would use either component [20]. The signal delay block was implemented as a behavioral voltage source with the parameter `V=delay(V(V),.3299m)`. As in the oscillator simulation, the potentiometer required for the R value was implemented as a resistor. Input data was provided from the oscillator simulation through the use of PWL files for the `V` input, which were generated by the oscillator SPICE model (these were not combined into a single model to ease the debugging process). As in the oscillator simulations, transient analysis was used with the command `.tran`

`0 30ms .5ms .1ms startup uic`. A complete schematic and the resulting netlist can be found in Appendix B.

4.2.1 Shift-band Performance

Figure 4.1 displays the filter output `Vout` with `Vs` from the oscillator simulation plotted for comparison (`Vs` is not used by the matched filter).

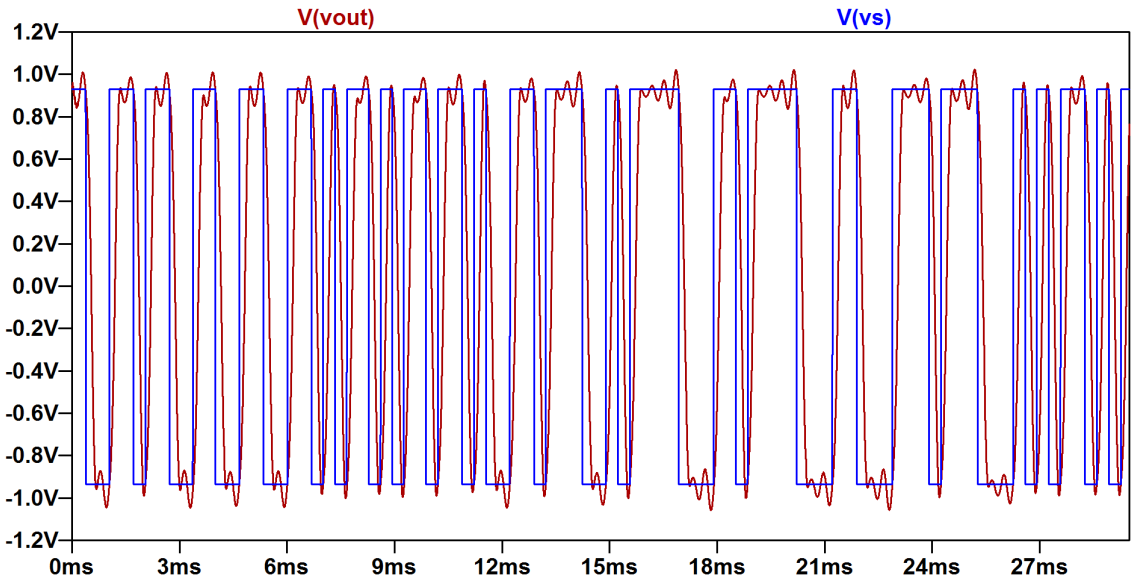

Figure 4.1: LF Matched Filter Shift-band `Vout` and `Vs` vs. t

It is readily apparent that the matched filter's output corresponds closely to the discrete-time state of the oscillator, albeit with an expected slight time delay. A published analytical solution is provided in Figure 4.2 for comparison, and verifies that the matched filter simulation behaves as expected. The blue waveform represents `V` in the top plot and `Vout` in the bottom plot.

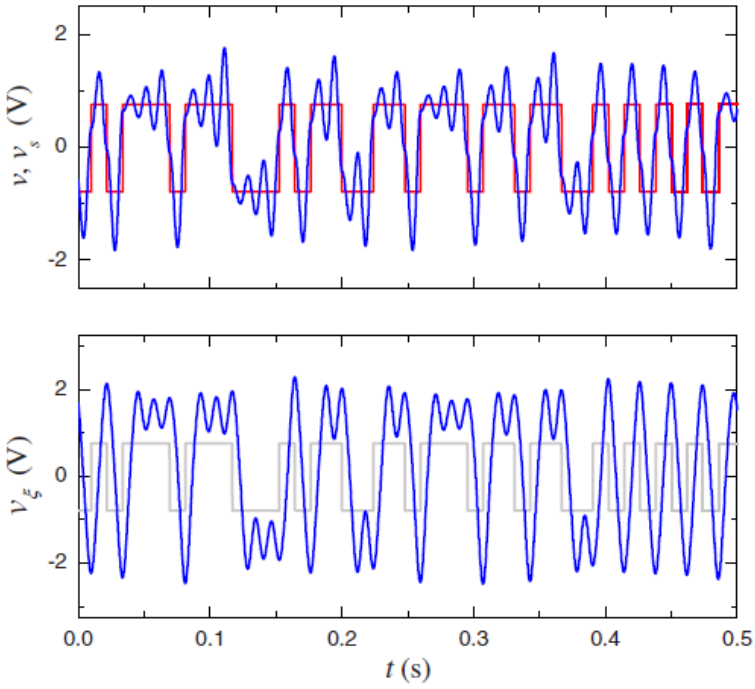

Figure 4.2: Published Oscillator Analytical Solution (top) and Matched Filter Analytical Solution (bottom) [2]

4.2.2 Folded-band Performance

Folded-band operation of the matched filter, shown in Figure 4.3, also closely corresponds to the discrete-time state of the oscillator operating in the folded-band. No analytical solution was available for comparison for folded-band operation, but, based on the results of shift-band operation, it is expected that the matched filter simulation behaves correctly in the folded-band as well.

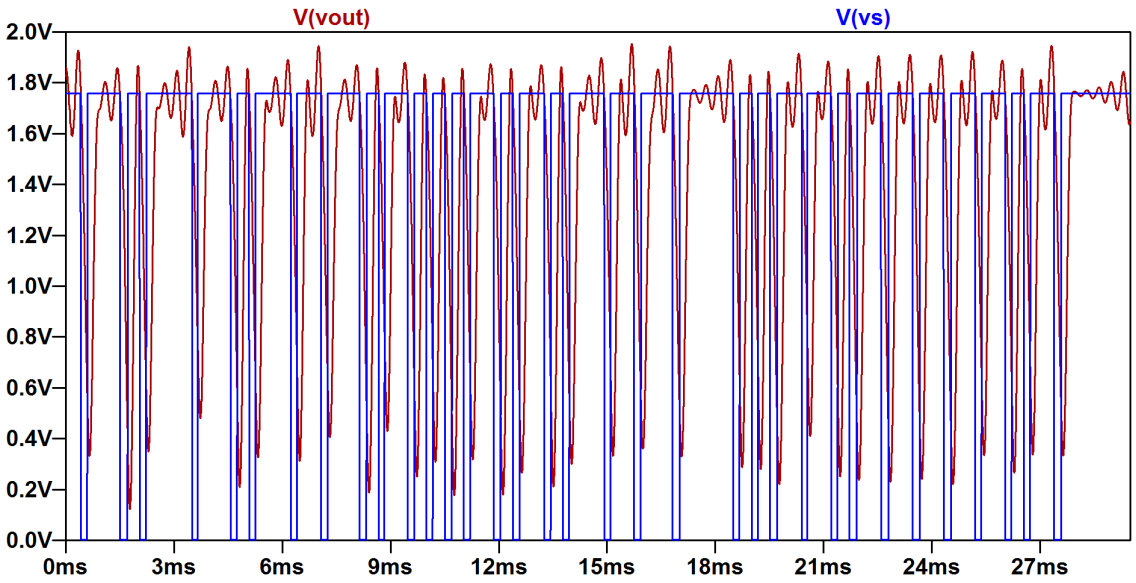

Figure 4.3: LF Matched Filter Folded-band `Vout` and `Vs` vs. t

4.3 Input Frequency Requirements

Because the simulation has been verified to match expectations, and to better evaluate its suitability for communication applications, the frequency requirements for the input signal `V` have also been explored. Low-pass, high-pass, and band-pass filtering of `V` has been performed and will be presented in this section.

The second order transfer functions for a low-pass filter, a high-pass filter, and a band-pass filter, all with unity gain [21], were used to filter `V`. These are given as:

$$H(s) = \frac{1}{1 + \frac{s}{Q\omega_c} + (\frac{s}{\omega_c})^2} \tag{4.1}$$

for low-pass and

$$H(s) = \frac{(\frac{s}{\omega_c})^2}{1 + \frac{s}{Q\omega_c} + (\frac{s}{\omega_c})^2} \tag{4.2}$$

for high-pass where $Q = \frac{1}{\sqrt{2}}$, and

$$H(s) = \frac{\frac{s}{Q\omega_0}}{1 + \frac{s}{Q\omega_0} + (\frac{s}{\omega_0})^2} \tag{4.3}$$

for band-pass where

$$Q = \sqrt{\frac{\omega_{c1}}{\omega_{c2}}} \tag{4.4}$$

and

$$\omega_0 = \sqrt{\omega_{c1}\omega_{c2}} \tag{4.5}$$

These were implemented by passing the parameters:

```
;low-pass
V=V(V) laplace=(1)/(1+s/({Q}*{w})+(s/{w})^2)
;high-pass
V=V(V) laplace=((s/{w})^2)/(1+s/({Q}*{w})+(s/{w})^2)
;band-pass
V=V(V) laplace=(s/({Q}*{w}))/(1+s/({Q}*{w})+(s/{w})^2)
```

to a behavioral voltage source. This implementation requires that the transfer function be defined for all inputs; as a result, the figures in this section do not contain results with an ω of 0 or ∞. The frequency response of each of these components has been evaluated and is presented in Figure 4.4 for the low-pass filter, high-pass filter, and band-pass filter. 50 Hz was used as f_{c1} (f_c for the high-pass filter) and 5 kHz was used as f_{c2} (f_c for the low-pass filter).

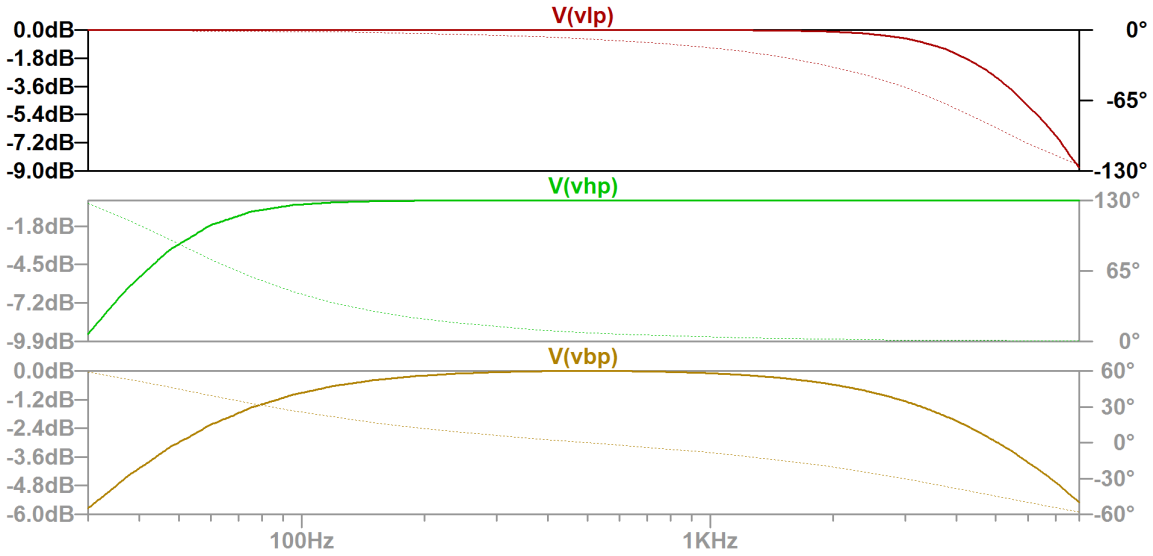

Figure 4.4: Low-pass (top), High-pass (middle), and Band-pass (bottom) Filter Frequency Response for f_{c1}=50 Hz and f_{c2}=5 kHz

Frequency response was calculated using LTSpiceIV's AC analysis with the command `.ac dec 10 30 8k`.

4.3.1 Low-pass Filtering

The effects of low-pass filtering, shown in Figure 4.5 for the shift-band reveals that the input signal from the oscillator does not require significant high frequency components for proper matched filter operation.

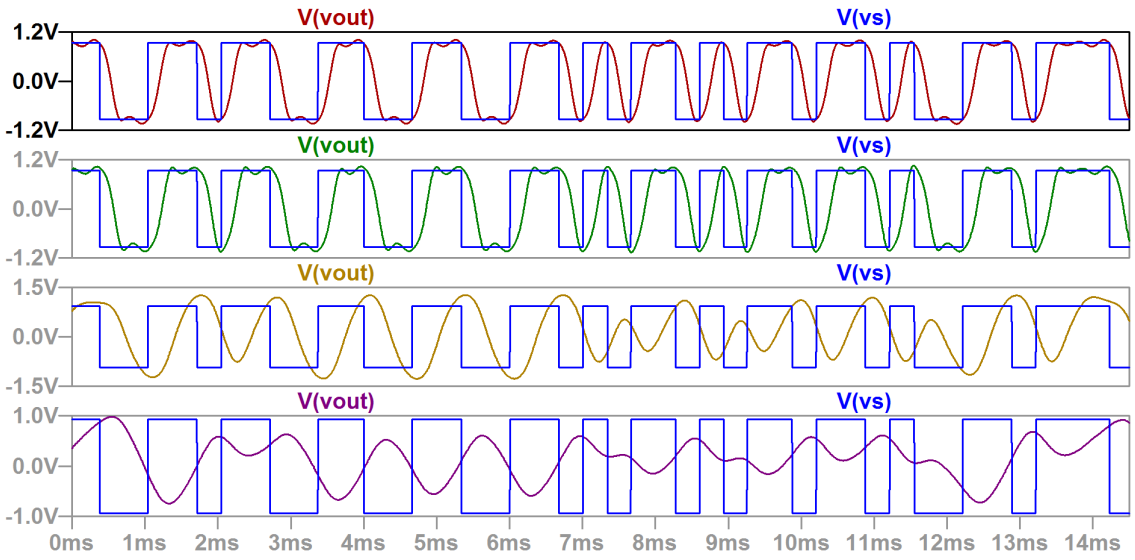

Figure 4.5: LF Matched Filter Shift-band Low-pass Filtered Input Performance with f_c=1 MHz (red), f_c=5 kHz (green), f_c=1 kHz (gold), and f_c=500 Hz (purple)

At a f_c of 5 kHz, `Vout` is approximately equal to `Vout` with no filtering applied, but filtering below 5 kHz causes `Vout` to become distorted in a manner that renders the original `Vs` unrecoverable. Figure 4.6 shows that the folded-band is also tolerant of high frequency filtering.

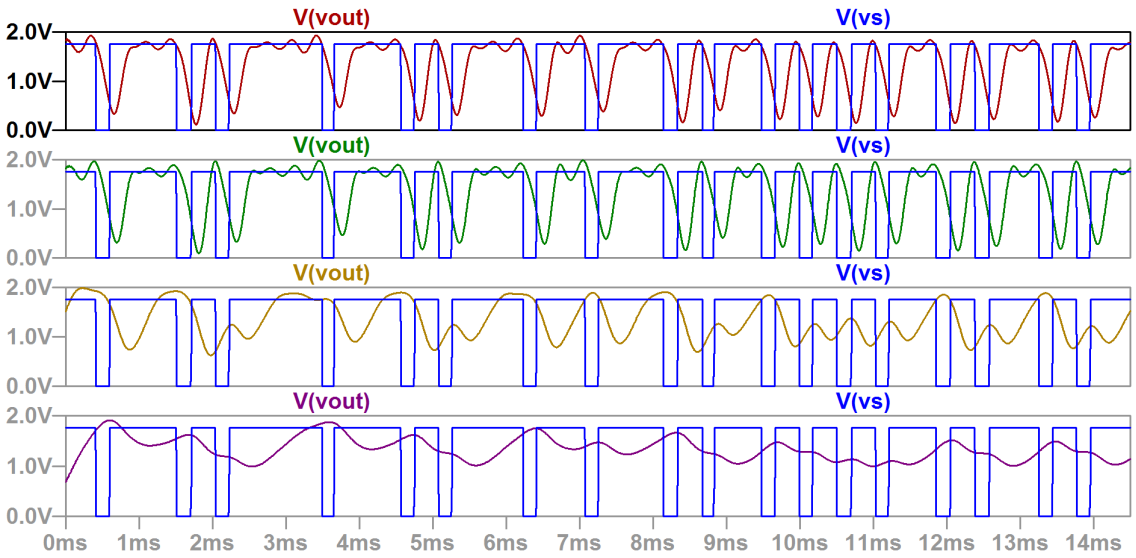

Figure 4.6: LF Matched Filter Folded-band Low-pass Filtered Input Performance with f_c=1 MHz (red), f_c=5 kHz (green), f_c=1 kHz (gold), and f_c=500 Hz (purple)

For both bands, a f_c of 5 kHz produces a `Vout` which is indiscernible from the unfiltered `Vout`, but filtering frequency components below 5 kHz, however, irrecoverably distorts `Vout`. This distortion arises from the smoothing of the sharp changes in `Vout`, caused by removal of high-frequency components, preventing the matched filter from varying its output rapidly enough to recreate `Vs`. Using the assumption that the peak frequencies calculated in Sections 3.2.1 and 3.2.2 are the fundamental frequencies (f_0) for `V`, these observations show that operation in the shift-band requires significantly more bandwidth (f_c was found to be approximately $5*f_0$) above the fundamental frequency than operation in the folded-band (where f_c was approximately $2*f_0$).

4.3.2 High-pass filtering

Despite its relative insensitivity to low-pass filtering, the matched filter shows a high sensitivity to high-pass filtering. Figure 4.7 demonstrates this finding for the shift-band.

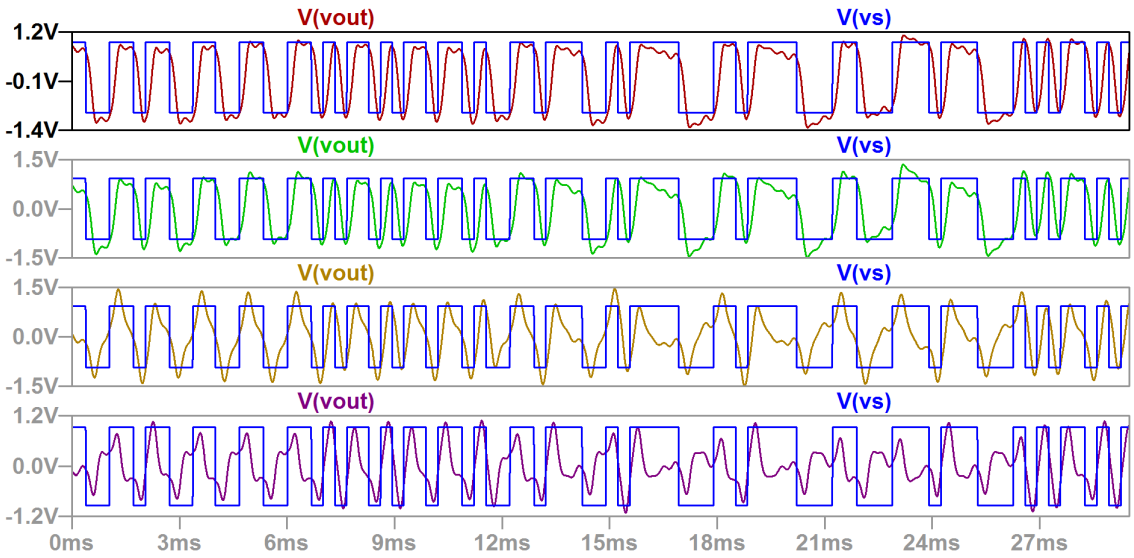

Figure 4.7: LF Matched Filter Shift-band High-pass Filtered Input Performance with f_c=50 Hz (red), f_c=100 Hz (green), f_c=500 Hz (gold), and f_c=1 kHz (purple)

Even with f_c at 100 Hz, distortion begins to occur in `Vout`; an f_c of 500 Hz renders `Vout` unrecoverable due to the waveform drifting towards 0 V whenever `Vs` remains in the same state for more than $1/2t$. Figure 4.8 demonstrates a similar finding for the folded-band.

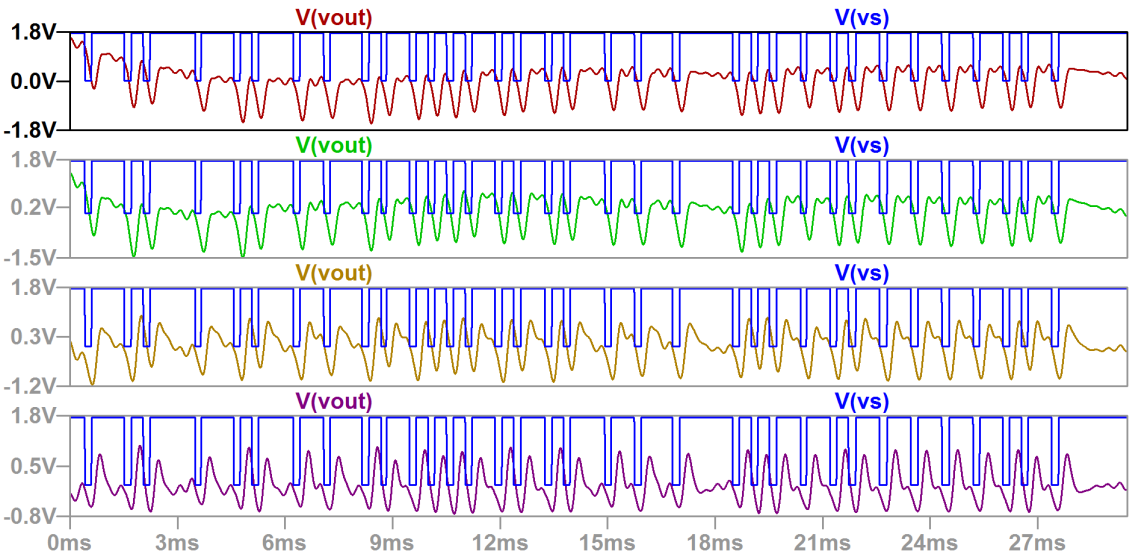

Figure 4.8: LF Matched Filter Folded-band High-pass Filtered Input Performance with f_c=50 Hz (red), f_c=100 Hz (green), f_c=500 Hz (gold), and f_c=1 kHz (purple)

From these figures, it can be seen that high-pass filtering above 100 Hz renders the matched filter's output unusable in either band of operation. In addition, high-pass filtering the input while operating in the folded-band causes `Vout` to shift in offset below 0 V. While these results show a need for leaving the vast majority of the low frequency components in place, it is notable that they demonstrate `Vout` does not require a DC component for proper operation.

4.3.3 Band-pass filtering

To provide the most relevant performance evaluation for communications, the results from low-pass filtering high-pass filtering must be combined to provide a range to be used in constructing a band-pass filter. By confining `V` to a finite bandwidth with no DC component, it would then be possible to perform modulation such that transmission and reception with reasonably-sized antennas could be realized. Figures 4.9 and 4.10 show the matched filter's performance for an f_{c1} of 50 Hz and an f_{c2} of 5 kHz.

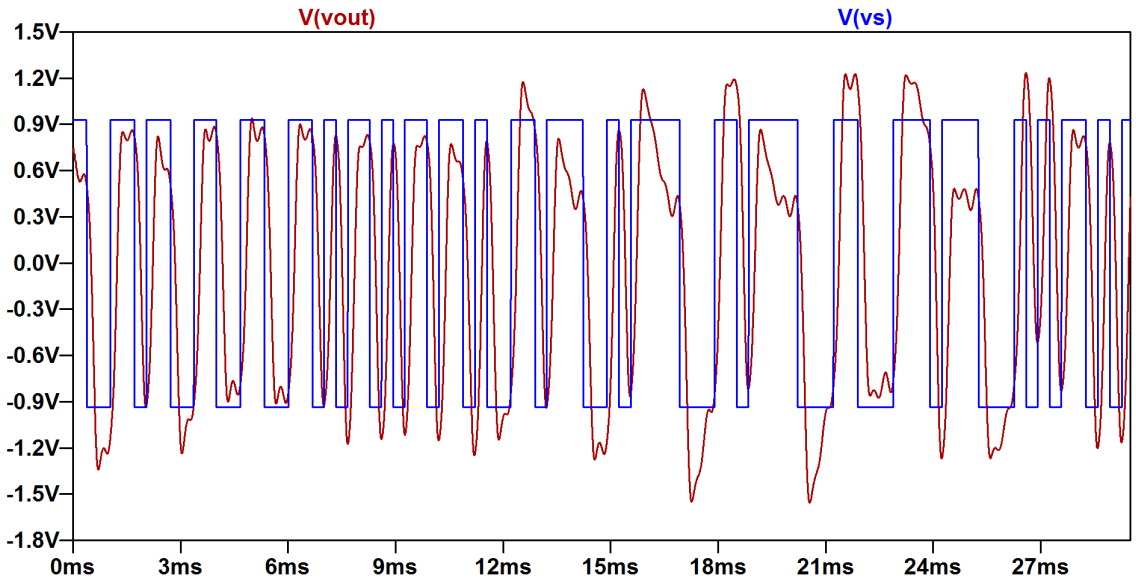

Figure 4.9: LF Matched Filter Shift-band Band-pass Filtered Input Performance

Although some distortion is present in `Vout`, it remains a close enough approximation to be recoverable.

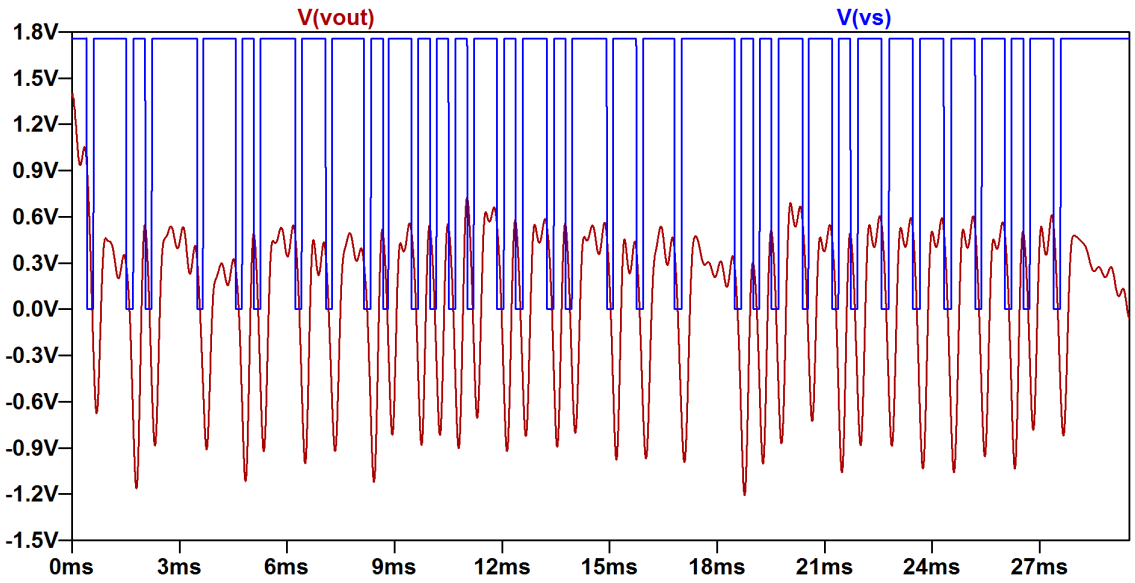

Figure 4.10: LF Matched Filter Folded-band Band-pass Filtered Input Performance

Restricting `V` to 50-5000 Hz results in a `Vout` that proves usable in the folded-band as well. The shift in offset voltage seen in high-pass filtering the folded-band input is also present with band-pass filtered input.

To verify that the discrete-time state of the oscillator can be accurately recovered from `Vout`, a LT1018 comparator is used to generate the recovered state `Srec`. Because the exact offset of `Vout` can vary, a 100k potentiometer is used to provide an adjustable comparison input which is set to match this offset. Figure 4.11 provides a schematic for this circuit.

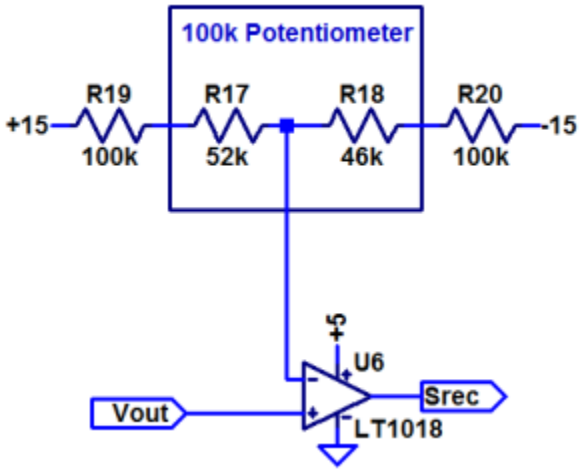

Figure 4.11: Discrete-time State Recovery Circuit

With this circuit placed at node `Vout` of the matched filter, a second output `Srec` is generated. This output, which is shown in Figures 4.12 and 4.13, exactly represents the discrete-time state of the oscillator (discounting the time delay introduced by the matched filter - this delay is expected from the original published design [2]).

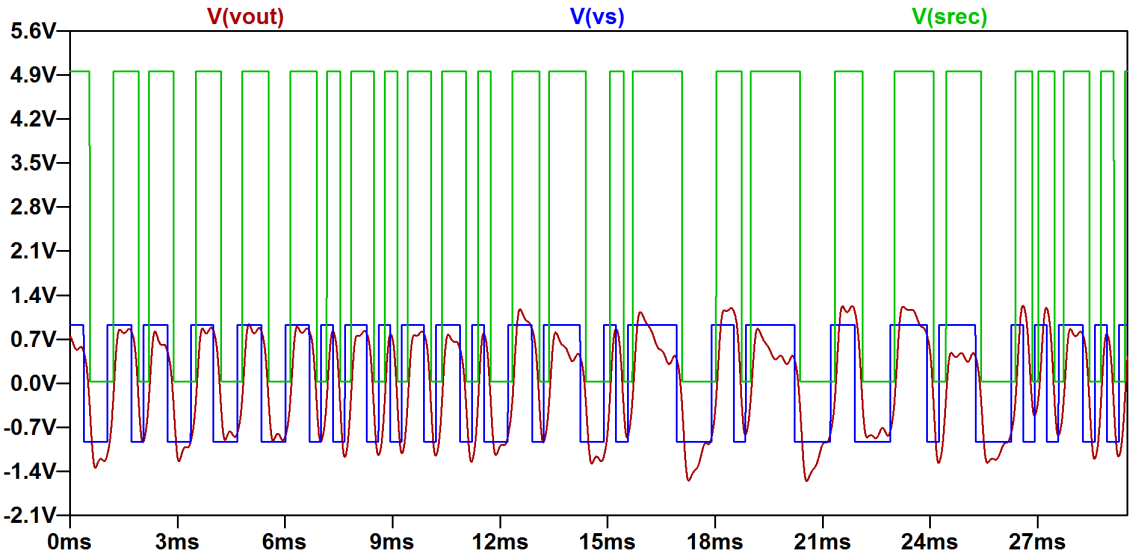

Figure 4.12: LF Matched Filter Shift-band Recovered Output

Because of the use of 0 V and 5 V as the negative and positive references for the comparator, the recovered output shifts between these levels rather than a positive and negative voltage of the same magnitude. Adjusting the reference voltages to the desired levels could more exactly recreate the shift-band `Vs` if required.

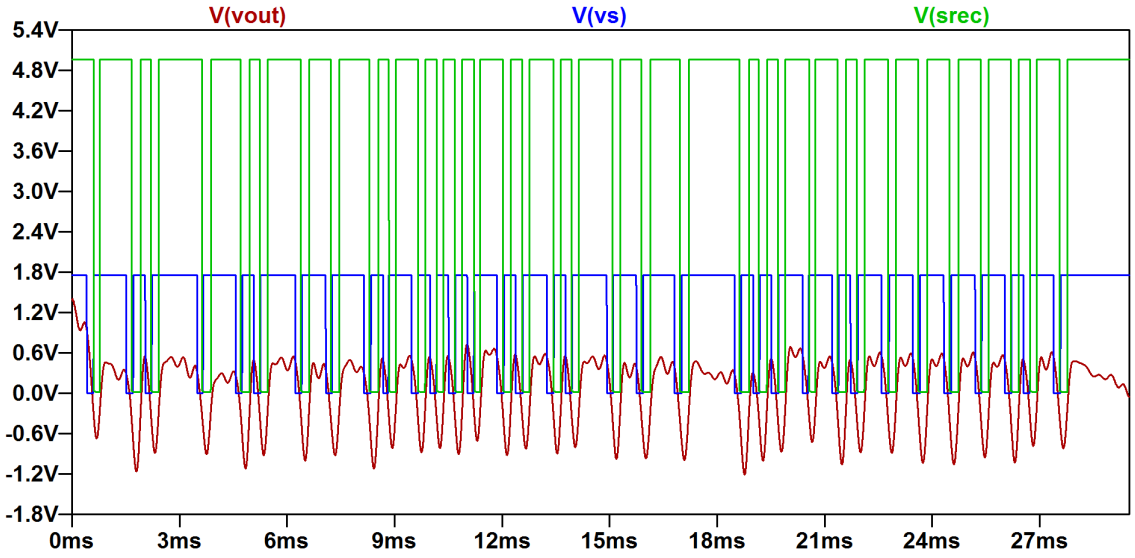

Figure 4.13: LF Matched Filter Folded-band Recovered Output

These results provide the conclusion that the entire communication chain, from generation with the oscillator to detection and recovery with the matched filter, can be successfully and accurately simulated with the developed SPICE model for both shift-band and folded-band operation.

Chapter 5

Conclusions and Future Work

Although commonly mistaken to be too difficult to find use in actual applications, recent work in the field of chaos has proven that it can, in fact, act as a powerful tool in multiple real-world scenarios. Fields such as radar and secure communications could potentially benefit greatly from the use of chaos. An overview of the previously published mathematics behind an exact solvable chaotic oscillator and a corresponding matched filter demonstrating this potential utility has been provided.

While previous work on this subject has used both analytical solutions and physical hardware as implementations of the mathematics, this thesis has added SPICE simulation using freely available tools as a method for exploring the nature of these solvable chaotic systems. Explanation of the SPICE implementation has been provided to assist with any future work in the area, and the accuracy of this simulation has been shown to be high when compared to an identical hardware implementation.

Building on this simulation capability, the previously strictly theoretical matched filter has been demonstrated as functional in simulation. An evaluation of its performance has been detailed and found to match published expected results. The completion of an accurate SPICE implementation has also enabled the input frequency requirements of this matched filter to be characterized. It has been found that, at LF operation, the matched filter's input can be bandwidth-restricted to a range suitable for communication applications.

Continuation of this work can proceed in a number of areas. Perhaps most importantly, a functional simulation provides a means for rapidly determining which coding sequences are practical for inclusion in a grammar for use with this system.

Compilation of this data will allow the system to move out of the realm of theory and into real application.

In addition, simulation capabilities will also allow design of an oscillator/matched filter pair operating at higher frequencies - this work has already begun as described in [11]. By possessing the ability to find and solve many potential problems before any actual hardware is assembled, the likelihood that the final hardware will be functional will be greatly increased.

Finally, the tolerances of all components used in the hardware design can be thoroughly characterized in simulation. As it is impossible to avoid slight variances between individual hardware components, examining the effects of these tolerances will be essential to developing a final design. For components where temperature data is available, simulation will also allow exploration of temperature effects, further enhancing the overall tolerance characterization.

References

[1] N. Corron and J. Blakely, "Exact folded-band chaotic oscillator," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 2, pp. 023 113–023 113, 2012.

[2] N. Corron, J. Blakely, and M. Stahl, "A matched filter for chaos," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 2, pp. 023 123–023 123, 2010.

[3] S. Hayes, C. Grebogi, and E. Ott, "Communicating with chaos," *Physical review letters*, vol. 70, no. 20, p. 3031, 1993.

[4] H. Leung, "Applying chaos to radar detection in an ocean environment: an experimental study," *Oceanic Engineering, IEEE Journal of*, vol. 20, no. 1, pp. 56–64, 1995.

[5] A. Tsonis and J. Elsner, "Chaos, strange attractors, and weather." *Bulletin of the American Meteorological Society*, vol. 70, pp. 14–23, 1989.

[6] K. Umeno, "Method of constructing exactly solvable chaos," *Physical Review E*, vol. 55, no. 5, p. 5280, 1997.

[7] S. Katsura and W. Fukuda, "Exactly solvable models showing chaotic behavior," *Physica A: Statistical and Theoretical Physics*, vol. 130, no. 3, pp. 597–605, 1985.

[8] E. Lorenz, "Deterministic nonperiodic flow," *Journal of the atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[9] D. Drake and D. Williams, "Linear, random representations of chaos," *Signal Processing, IEEE Transactions on*, vol. 55, no. 4, pp. 1379–1389, 2007.

[10] S. Hayes, "Chaos from linear systems: Implications for communicating with chaos, and the nature of determinism and randomness," in *Journal of Physics: Conference Series*, vol. 23. IOP Publishing, 2005, p. 215.

[11] A. Beal, J. Bailey, S. Hale, R. Dean, M. Hamilton, J. Tugnait, D. Hahs, and N. Corron, "Design and simulation of a high frequency exact solvable chaotic oscillator."

[12] R. C. Jaeger and T. N. Blalock, *Microelectronic Circuit Design*, S. W. Director, Ed. McGraw-Hill, 2008.

[13] J. T. Taylor and Q. Huang, Eds., *CRC Handbook of Electrical Filters*. CRC Press, 1997.

[14] *LTSpiceIV Manual*, LTspiceHelp.chm, Linear Technology, Jun. 2012. [Online]. Available: http://cds.linear.com/docs/ltspice

[15] *TL084, TL084A, TL084B PSpice Model*, sloj071.zip, Texas Instruments, Jan. 2002. [Online]. Available: http://www.ti.com/product/tl084#simulationmodels

[16] *LM139 PSpice Model*, slcj002.zip, Texas Instruments, Jan. 2002. [Online]. Available: http://www.ti.com/product/lm139#simulationmodels

[17] *74HC08; 74HCT08 Quad 2-input AND gate*, 74HC_HCT08.pdf, Philips Semiconductors, Jul. 2003. [Online]. Available: http://www.nxp.com/documents/data_sheet/

[18] D. Hamill, "Learning about chaotic circuits with spice," *Education, IEEE Transactions on*, vol. 36, no. 1, pp. 28–35, 1993.

[19] G. Heidari-Bateni and C. McGillem, "A chaotic direct-sequence spread-spectrum communication system," *Communications, IEEE Transactions on*, vol. 42, no. 234, pp. 1524–1527, 1994.

[20] *LT1001 Precision Operational Amplifier*, 1001fb.pdf, Linear Technology, 1983. [Online]. Available: http://http://cds.linear.com/docs/Datasheet/

[21] E. Kamen and B. Heck, *Fundamentals of Signals and Systems: With MATLAB Examples*. Prentice Hall PTR, 2000.

Appendix A

LF Oscillator Parts List, SPICE Netlist, and Schematic

Schematic	**Description**	**Manuf.**	**Part Number**
A1	74HC08 quad 2-input AND gate	NXP	74HC08N
A2	74HC32 quad 2-input OR gate	NXP	74HC32N
A3	74HC32 quad 2-input OR gate	NXP	74HC32N
A4	74HC08 quad 2-input AND gate	NXP	74HC08N
C1	.01 μF capacitor	TDK	FK28C0G1H102J
C2	.01 μF capacitor	TDK	FK28C0G1H102J
R1	221 Ω resistor	Vishay	MRS25000C2210FRP00
R2	221 Ω resistor	Vishay	MRS25000C2210FRP00
R3	100 kΩ potentiometer	Murata	PV36W104C01B00
R4	3.01 kΩ resistor	Stackpole	RNF14FTD3K01
R5	1 kΩ resistor	Stackpole	RNF14FTD1K00
R6	3.01 kΩ resistor	Stackpole	RNF14FTD3K01
R7	3.01 kΩ resistor	Stackpole	RNF14FTD3K01
R8	30.1 kΩ resistor	Stackpole	RNF14FTD30K1
R9	10 kΩ resistor	Vishay	MRS25000C1002FRP00
R10	15 kΩ resistor	Vishay	MRS25000C1502FRP00
R11	10 kΩ resistor	Vishay	MRS25000C1002FRP00
R12	10 kΩ resistor	Vishay	MRS25000C1002FRP00
R13	15 kΩ resistor	Vishay	MRS25000C1502FRP00
R14	10 kΩ resistor	Vishay	MRS25000C1002FRP00
R15	10 kΩ resistor	Vishay	MRS25000C1002FRP00
R16	10 kΩ resistor	Vishay	MRS25000C1002FRP00

R18	10 kΩ resistor	Vishay	MRS25000C1002FRP00
R19	10 kΩ resistor	Vishay	MRS25000C1002FRP00
R20	30.1 kΩ resistor	Stackpole	RNF14FTD30K1
R21	10 kΩ resistor	Vishay	MRS25000C1002FRP00
R22	10 kΩ resistor	Vishay	MRS25000C1002FRP00
R23+R30	100 kΩ potentiometer	Murata	PV36W104C01B00
R24	14 kΩ resistor	Vishay	MRS25000C1402FRP00
R25	11.3 kΩ resistor	Vishay	MRS25000C1132FRP00
R26+R27	100 kΩ potentiometer	Murata	PV36W104C01B00
R28	100 kΩ resistor	Vishay	MRS25000C1003FRP00
R29	100 kΩ resistor	Vishay	MRS25000C1003FRP00
U1	LM339 low power quad comparator	TI	LM339AN
U2	LM339 low power quad comparator	TI	LM339AN
U4	TL084 JFET-input op-amp	TI	TL084CN
U5	TL084 JFET-input op-amp	TI	TL084CN
U6	TL084 JFET-input op-amp	TI	TL084CN
U7	TL084 JFET-input op-amp	TI	TL084CN
U8	TL084 JFET-input op-amp	TI	TL084CN
U9	TL084 JFET-input op-amp	TI	TL084CN

Table A.1: LF Oscillator Parts List

SPICE models for the TL084 and LM339 components are available from [15] and [16], respectively. These models were used to generate this netlist.

```
r1 v n005 221
r2 n011 n005 221
r3 n011 0 25.33k
r4 n016 v 3.01k
r5 n016 n018 1k
r6 n018 n017 3.01k
r7 vs_1 n019 3.01k
c1 n019 n017 .01u
v1 -15 0 -15
v3 +15 0 15
r11 n006 vd 10k
v2 +5 0 5
r8 n015 c2_i 30k
r20 vd c1_i 30k
r9 +5 c1_i 10k
r10 c1_i 0 15k
r12 +5 c2_i 10k
r13 c2_i 0 15k
r14 +5 c2_o 10k
r15 +5 n007 10k
r16 0 n004 10k
r18 +5 n004 10k
r19 0 n014 10k
r21 +5 n014 10k
r22 n009 n010 10k
r23 vs n003 62.711k
r24 n003 +15 14k
r25 -15 n010 11k
r26 n001 c1_i 25.572k
r27 c1_i n002 68.353k
r28 n001 +15 100k
r29 -15 n002 100k
r30 n010 vs 31.309k
c2 n006 v .01u
a1 n007 c2_o 0 0 0 0 n008 0 and vhigh=4.95v
a2 0 0 0 n007 c2_o 0 n013 0 or vhigh=4.95v
a3 0 n008 0 0 n012 0 n009 0 or vhigh=4.95v
a4 n009 n013 0 0 0 0 n012 0 and vhigh=4.95v
c:u6:1 u6:11 u6:12 3.498e-12
c:u6:2 u6:6 u6:7 15.00e-12
d:u6:c vd u6:53 u6:dx
d:u6:e u6:54 vd u6:dx
d:u6:lp u6:90 u6:91 u6:dx
d:u6:ln u6:92 u6:90 u6:dx
```

```
d:u6:p -15 +15 u6:dx
b:u6:§egnd u6:99 0 v=.5*v(+15)+.5*v(-15)
b:u6:§fb u6:7 u6:99 i=4.715e6*i(v:u6:b)+-5e6*i(v:u6:c)+5e6*i(v:
+u6:e)+5e6*i(v:u6:lp)+-5e6*i(v:u6:ln)
g:u6:a u6:6 0 u6:11 u6:12 282.8e-6
g:u6:cm 0 u6:6 u6:10 u6:99 8.942e-9
i:u6:ss +15 u6:10 dc 195.0e-6
h:u6:lim u6:90 0 v:u6:lim 1k
j:u6:1 u6:11 n006 u6:10 u6:jx
j:u6:2 u6:12 0 u6:10 u6:jx
r:u6:2 u6:6 u6:9 100.0e3
r:u6:d1 -15 u6:11 3.536e3
r:u6:d2 -15 u6:12 3.536e3
r:u6:o1 u6:8 vd 150
r:u6:o2 u6:7 u6:99 150
r:u6:p +15 -15 2.143e3
r:u6:ss u6:10 u6:99 1.026e6
v:u6:b u6:9 0 dc 0
v:u6:c +15 u6:53 dc 2.200
v:u6:e u6:54 -15 dc 2.200
v:u6:lim u6:7 u6:8 dc 0
v:u6:lp u6:91 0 dc 25
v:u6:ln 0 u6:92 dc 25
f:u1:1 u1:9 +5 v:u1:1 1
i:u1:ee +5 u1:7 dc 100.0e-6
v:u1:i1 u1:21 c1_i dc .75
v:u1:i2 u1:22 n004 dc .75
q:u1:1 u1:9 u1:21 u1:7 u1:qin
q:u1:2 u1:8 u1:22 u1:7 u1:qin
q:u1:3 u1:9 u1:8 0 u1:qmo
q:u1:4 u1:8 u1:8 0 u1:qmi
e:u1:1 u1:10 0 u1:9 0 1
v:u1:1 u1:10 u1:11 dc 0
q:u1:5 n007 u1:11 0 u1:qoc
d:u1:p 0 +5 u1:dx
r:u1:p +5 0 46.3e3
f:u2:1 u2:9 +5 v:u2:1 1
i:u2:ee +5 u2:7 dc 100.0e-6
v:u2:i1 u2:21 c2_i dc .75
v:u2:i2 u2:22 n014 dc .75
q:u2:1 u2:9 u2:21 u2:7 u2:qin
q:u2:2 u2:8 u2:22 u2:7 u2:qin
q:u2:3 u2:9 u2:8 0 u2:qmo
q:u2:4 u2:8 u2:8 0 u2:qmi
e:u2:1 u2:10 0 u2:9 0 1
v:u2:1 u2:10 u2:11 dc 0
q:u2:5 c2_o u2:11 0 u2:qoc
```

```
d:u2:p 0 +5 u2:dx
r:u2:p +5 0 46.3e3
c:u4:1 u4:11 u4:12 3.498e-12
c:u4:2 u4:6 u4:7 15.00e-12
d:u4:c n015 u4:53 u4:dx
d:u4:e u4:54 n015 u4:dx
d:u4:lp u4:90 u4:91 u4:dx
d:u4:ln u4:92 u4:90 u4:dx
d:u4:p -15 +15 u4:dx
b:u4:§egnd u4:99 0 v=.5*v(+15)+.5*v(-15)
b:u4:§fb u4:7 u4:99 i=4.715e6*i(v:u4:b)+-5e6*i(v:u4:c)+5e6*i(v:
+u4:e)+5e6*i(v:u4:lp)+-5e6*i(v:u4:ln)
g:u4:a u4:6 0 u4:11 u4:12 282.8e-6
g:u4:cm 0 u4:6 u4:10 u4:99 8.942e-9
i:u4:ss +15 u4:10 dc 195.0e-6
h:u4:lim u4:90 0 v:u4:lim 1k
j:u4:1 u4:11 n015 u4:10 u4:jx
j:u4:2 u4:12 v u4:10 u4:jx
r:u4:2 u4:6 u4:9 100.0e3
r:u4:d1 -15 u4:11 3.536e3
r:u4:d2 -15 u4:12 3.536e3
r:u4:o1 u4:8 n015 150
r:u4:o2 u4:7 u4:99 150
r:u4:p +15 -15 2.143e3
r:u4:ss u4:10 u4:99 1.026e6
v:u4:b u4:9 0 dc 0
v:u4:c +15 u4:53 dc 2.200
v:u4:e u4:54 -15 dc 2.200
v:u4:lim u4:7 u4:8 dc 0
v:u4:lp u4:91 0 dc 25
v:u4:ln 0 u4:92 dc 25
c:u5:1 u5:11 u5:12 3.498e-12
c:u5:2 u5:6 u5:7 15.00e-12
d:u5:c n005 u5:53 u5:dx
d:u5:e u5:54 n005 u5:dx
d:u5:lp u5:90 u5:91 u5:dx
d:u5:ln u5:92 u5:90 u5:dx
d:u5:p -15 +15 u5:dx
b:u5:§egnd u5:99 0 v=.5*v(+15)+.5*v(-15)
b:u5:§fb u5:7 u5:99 i=4.715e6*i(v:u5:b)+-5e6*i(v:u5:c)+5e6*i(v:
+u5:e)+5e6*i(v:u5:lp)+-5e6*i(v:u5:ln)
g:u5:a u5:6 0 u5:11 u5:12 282.8e-6
g:u5:cm 0 u5:6 u5:10 u5:99 8.942e-9
i:u5:ss +15 u5:10 dc 195.0e-6
h:u5:lim u5:90 0 v:u5:lim 1k
j:u5:1 u5:11 n011 u5:10 u5:jx
j:u5:2 u5:12 v u5:10 u5:jx
```

```
r:u5:2 u5:6 u5:9 100.0e3
r:u5:d1 -15 u5:11 3.536e3
r:u5:d2 -15 u5:12 3.536e3
r:u5:o1 u5:8 n005 150
r:u5:o2 u5:7 u5:99 150
r:u5:p +15 -15 2.143e3
r:u5:ss u5:10 u5:99 1.026e6
v:u5:b u5:9 0 dc 0
v:u5:c +15 u5:53 dc 2.200
v:u5:e u5:54 -15 dc 2.200
v:u5:lim u5:7 u5:8 dc 0
v:u5:lp u5:91 0 dc 25
v:u5:ln 0 u5:92 dc 25
c:u7:1 u7:11 u7:12 3.498e-12
c:u7:2 u7:6 u7:7 15.00e-12
d:u7:c vs_1 u7:53 u7:dx
d:u7:e u7:54 vs_1 u7:dx
d:u7:lp u7:90 u7:91 u7:dx
d:u7:ln u7:92 u7:90 u7:dx
d:u7:p -15 +15 u7:dx
b:u7:§egnd u7:99 0 v=.5*v(+15)+.5*v(-15)
b:u7:§fb u7:7 u7:99 i=4.715e6*i(v:u7:b)+-5e6*i(v:u7:c)+5e6*i(v:
+u7:e)+5e6*i(v:u7:lp)+-5e6*i(v:u7:ln)
g:u7:a u7:6 0 u7:11 u7:12 282.8e-6
g:u7:cm 0 u7:6 u7:10 u7:99 8.942e-9
i:u7:ss +15 u7:10 dc 195.0e-6
h:u7:lim u7:90 0 v:u7:lim 1k
j:u7:1 u7:11 vs_1 u7:10 u7:jx
j:u7:2 u7:12 vs u7:10 u7:jx
r:u7:2 u7:6 u7:9 100.0e3
r:u7:d1 -15 u7:11 3.536e3
r:u7:d2 -15 u7:12 3.536e3
r:u7:o1 u7:8 vs_1 150
r:u7:o2 u7:7 u7:99 150
r:u7:p +15 -15 2.143e3
r:u7:ss u7:10 u7:99 1.026e6
v:u7:b u7:9 0 dc 0
v:u7:c +15 u7:53 dc 2.200
v:u7:e u7:54 -15 dc 2.200
v:u7:lim u7:7 u7:8 dc 0
v:u7:lp u7:91 0 dc 25
v:u7:ln 0 u7:92 dc 25
c:u8:1 u8:11 u8:12 3.498e-12
c:u8:2 u8:6 u8:7 15.00e-12
d:u8:c n017 u8:53 u8:dx
d:u8:e u8:54 n017 u8:dx
d:u8:lp u8:90 u8:91 u8:dx
```

```
d:u8:ln u8:92 u8:90 u8:dx
d:u8:p -15 +15 u8:dx
b:u8:§egnd u8:99 0 v=.5*v(+15)+.5*v(-15)
b:u8:§fb u8:7 u8:99 i=4.715e6*i(v:u8:b)+-5e6*i(v:u8:c)+5e6*i(v:
+u8:e)+5e6*i(v:u8:lp)+-5e6*i(v:u8:ln)
g:u8:a u8:6 0 u8:11 u8:12 282.8e-6
g:u8:cm 0 u8:6 u8:10 u8:99 8.942e-9
i:u8:ss +15 u8:10 dc 195.0e-6
h:u8:lim u8:90 0 v:u8:lim 1k
j:u8:1 u8:11 n018 u8:10 u8:jx
j:u8:2 u8:12 v u8:10 u8:jx
r:u8:2 u8:6 u8:9 100.0e3
r:u8:d1 -15 u8:11 3.536e3
r:u8:d2 -15 u8:12 3.536e3
r:u8:o1 u8:8 n017 150
r:u8:o2 u8:7 u8:99 150
r:u8:p +15 -15 2.143e3
r:u8:ss u8:10 u8:99 1.026e6
v:u8:b u8:9 0 dc 0
v:u8:c +15 u8:53 dc 2.200
v:u8:e u8:54 -15 dc 2.200
v:u8:lim u8:7 u8:8 dc 0
v:u8:lp u8:91 0 dc 25
v:u8:ln 0 u8:92 dc 25
c:u9:1 u9:11 u9:12 3.498e-12
c:u9:2 u9:6 u9:7 15.00e-12
d:u9:c n016 u9:53 u9:dx
d:u9:e u9:54 n016 u9:dx
d:u9:lp u9:90 u9:91 u9:dx
d:u9:ln u9:92 u9:90 u9:dx
d:u9:p -15 +15 u9:dx
b:u9:§egnd u9:99 0 v=.5*v(+15)+.5*v(-15)
b:u9:§fb u9:7 u9:99 i=4.715e6*i(v:u9:b)+-5e6*i(v:u9:c)+5e6*i(v:
+u9:e)+5e6*i(v:u9:lp)+-5e6*i(v:u9:ln)
g:u9:a u9:6 0 u9:11 u9:12 282.8e-6
g:u9:cm 0 u9:6 u9:10 u9:99 8.942e-9
i:u9:ss +15 u9:10 dc 195.0e-6
h:u9:lim u9:90 0 v:u9:lim 1k
j:u9:1 u9:11 n018 u9:10 u9:jx
j:u9:2 u9:12 n019 u9:10 u9:jx
r:u9:2 u9:6 u9:9 100.0e3
r:u9:d1 -15 u9:11 3.536e3
r:u9:d2 -15 u9:12 3.536e3
r:u9:o1 u9:8 n016 150
r:u9:o2 u9:7 u9:99 150
r:u9:p +15 -15 2.143e3
r:u9:ss u9:10 u9:99 1.026e6
```

```
v:u9:b u9:9 0 dc 0
v:u9:c +15 u9:53 dc 2.200
v:u9:e u9:54 -15 dc 2.200
v:u9:lim u9:7 u9:8 dc 0
v:u9:lp u9:91 0 dc 25
v:u9:ln 0 u9:92 dc 25
.model u9:jx pjf(is=15.00e-12 beta=270.1e-6 vto=-1)
.model u9:dx d(is=800.0e-18)
.model u8:jx pjf(is=15.00e-12 beta=270.1e-6 vto=-1)
.model u8:dx d(is=800.0e-18)
.model u7:jx pjf(is=15.00e-12 beta=270.1e-6 vto=-1)
.model u7:dx d(is=800.0e-18)
.model u5:jx pjf(is=15.00e-12 beta=270.1e-6 vto=-1)
.model u5:dx d(is=800.0e-18)
.model u4:jx pjf(is=15.00e-12 beta=270.1e-6 vto=-1)
.model u4:dx d(is=800.0e-18)
.model u2:dx d(is=800.0e-18)
.model u2:qoc npn(is=800.0e-18 bf=20.29e3 cjc=1e-15 tf=942.6e
+-12 tr=543.8e-9)
.model u2:qmo npn(is=800.0e-18 bf=1000 cjc=1e-15 tr=807.4e-9)
.model u2:qmi npn(is=800.0e-18 bf=1002)
.model u2:qin pnp(is=800.0e-18 bf=2.000e3)
.model u1:dx d(is=800.0e-18)
.model u1:qoc npn(is=800.0e-18 bf=20.29e3 cjc=1e-15 tf=942.6e
+-12 tr=543.8e-9)
.model u1:qmo npn(is=800.0e-18 bf=1000 cjc=1e-15 tr=807.4e-9)
.model u1:qmi npn(is=800.0e-18 bf=1002)
.model u1:qin pnp(is=800.0e-18 bf=2.000e3)
.model u6:jx pjf(is=15.00e-12 beta=270.1e-6 vto=-1)
.model u6:dx d(is=800.0e-18)
.tran 0 31ms .5ms .1ms startup uic
.end
```

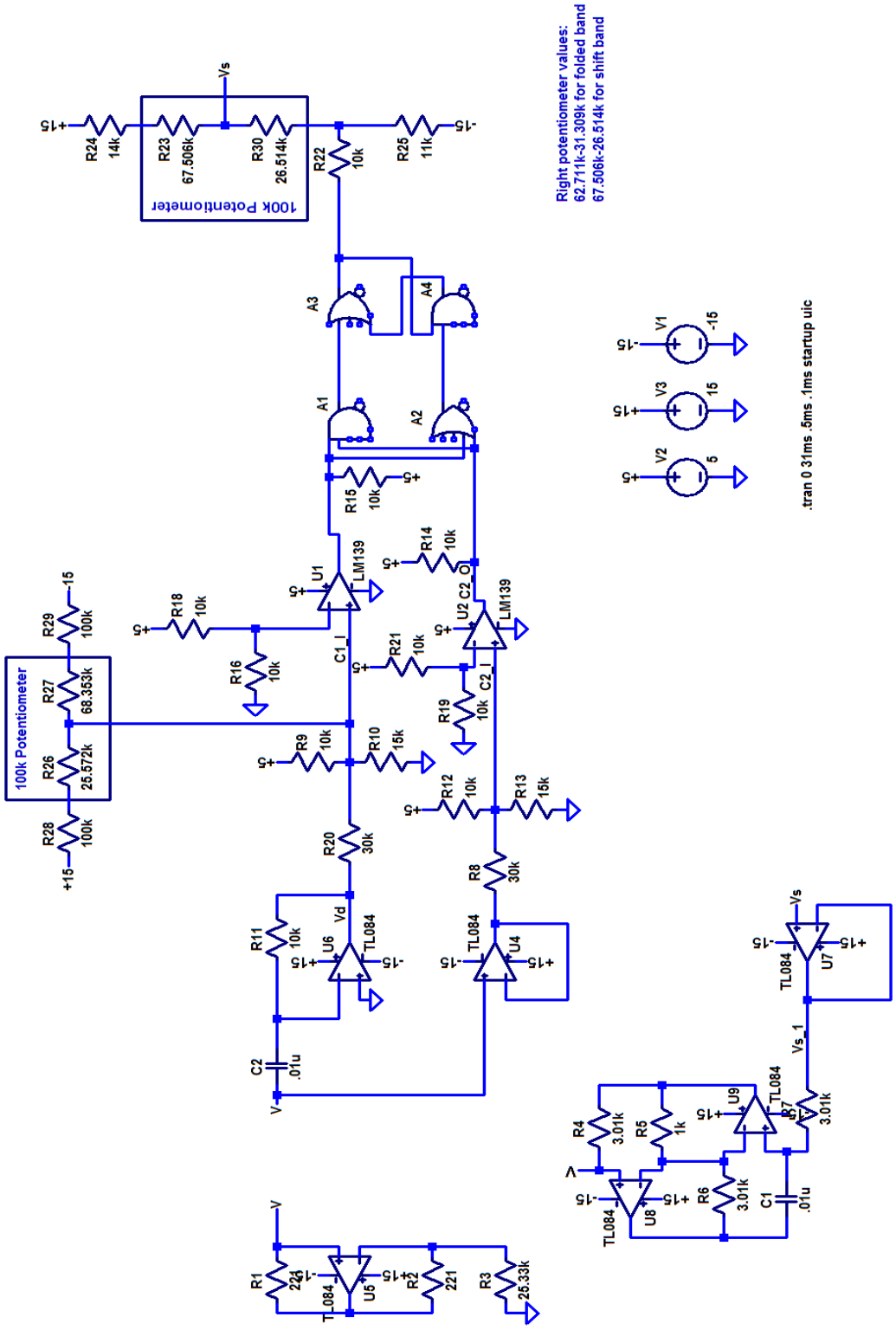

Figure A.1: Low Frequency Chaotic Oscillator Schematic

Appendix B
LF Matched Filter SPICE Netlist and Schematic

```
r1 n004 n007 3299
r2 n009 vdelay 1k
r3 n006 v 1k
r4 0 n009 1k
r5 n006 n007 1k
c1 n005 n004 .1u
r6 n008 n005 3.01k
r7 vout n001 3.01k
r8 vout n003 1k
r9 n003 n002 3.01k
c2 n008 n002 .01u
r10 vout 0 25.33k
c3 vout 0 .01u
b1 vdelay 0 v=delay(v(v),.3299m)
r11 n005 n004 10meg
a:u1:1 0 u1:n004 0 0 0 0 u1:x 0 ota g=150u iout=7u cout=28p en
+=9.8n enk=4 vhigh=1e308 vlow=-1e308
a:u1:2 n006 n009 0 0 0 0 0 0 ota g=0 in=.1p ink=70
c:u1:2 u1:n004 0 .75p rpar=100k noiseless
r:u1:1 +15 u1:x 10g noiseless
r:u1:2 u1:x -15 10g noiseless
m:u1:1 +15 u1:n005 n007 n007 u1:n temp=27
m:u1:2 -15 u1:n005 n007 n007 u1:p temp=27
d:u1:1 n007 u1:x u1:x
c:u1:3 u1:n005 0 .075p rpar=1meg noiseless
d:u1:4 n006 n009 u1:di
c:u1:5 n006 n009 1p rpar=80meg
c:u1:7 +15 n007 .2p
c:u1:8 n007 -15 .2p
b:u1:1 0 u1:n004 i=10u*dnlim(uplim(v(n009),v(+15)-.9,.3), v
+(-15)+.9, .3)+1n*v(n009)
b:u1:2 u1:n004 0 i=10u*dnlim(uplim(v(n006),v(+15)-.899,.3), v
+(-15)+.899, .3)+1n*v(n006)
b:u1:3 0 u1:n005 i=1u*dnlim(uplim(v(u1:x),v(+15)-.9,.5), v
+(-15)+.9,.5)+1p*v(u1:x)
c:u1:9 +15 n009 .5p rpar=1.12t noiseless
c:u1:4 n009 -15 .5p rpar=1.12t noiseless
c:u1:6 n006 -15 .5p rpar=1.12t noiseless
c:u1:10 +15 n006 .5p rpar=1.12t noiseless
```

```
a:u2:1 0 u2:n004 0 0 0 0 u2:x 0 ota g=150u iout=7u cout=28p en
+=9.8n enk=4 vhigh=1e308 vlow=-1e308
a:u2:2 n004 0 0 0 0 0 0 0 ota g=0 in=.1p ink=70
c:u2:2 u2:n004 0 .75p rpar=100k noiseless
r:u2:1 +15 u2:x 10g noiseless
r:u2:2 u2:x -15 10g noiseless
m:u2:1 +15 u2:n005 n005 n005 u2:n temp=27
m:u2:2 -15 u2:n005 n005 n005 u2:p temp=27
d:u2:1 n005 u2:x u2:x
c:u2:3 u2:n005 0 .075p rpar=1meg noiseless
d:u2:4 n004 0 u2:di
c:u2:5 n004 0 1p rpar=80meg
c:u2:7 +15 n005 .2p
c:u2:8 n005 -15 .2p
b:u2:1 0 u2:n004 i=10u*dnlim(uplim(v(0),v(+15)-.9,.3), v(-15)
++.9, .3)+1n*v(0)
b:u2:2 u2:n004 0 i=10u*dnlim(uplim(v(n004),v(+15)-.899,.3), v
+(-15)+.899, .3)+1n*v(n004)
b:u2:3 0 u2:n005 i=1u*dnlim(uplim(v(u2:x),v(+15)-.9,.5), v
+(-15)+.9,.5)+1p*v(u2:x)
c:u2:9 +15 0 .5p rpar=1.12t noiseless
c:u2:4 0 -15 .5p rpar=1.12t noiseless
c:u2:6 n004 -15 .5p rpar=1.12t noiseless
c:u2:10 +15 n004 .5p rpar=1.12t noiseless
a:u3:1 0 u3:n004 0 0 0 0 u3:x 0 ota g=150u iout=7u cout=28p en
+=9.8n enk=4 vhigh=1e308 vlow=-1e308
a:u3:2 n003 n008 0 0 0 0 0 0 ota g=0 in=.1p ink=70
c:u3:2 u3:n004 0 .75p rpar=100k noiseless
r:u3:1 +15 u3:x 10g noiseless
r:u3:2 u3:x -15 10g noiseless
m:u3:1 +15 u3:n005 vout vout u3:n temp=27
m:u3:2 -15 u3:n005 vout vout u3:p temp=27
d:u3:1 vout u3:x u3:x
c:u3:3 u3:n005 0 .075p rpar=1meg noiseless
d:u3:4 n003 n008 u3:di
c:u3:5 n003 n008 1p rpar=80meg
c:u3:7 +15 vout .2p
c:u3:8 vout -15 .2p
b:u3:1 0 u3:n004 i=10u*dnlim(uplim(v(n008),v(+15)-.9,.3), v
+(-15)+.9, .3)+1n*v(n008)
b:u3:2 u3:n004 0 i=10u*dnlim(uplim(v(n003),v(+15)-.899,.3), v
+(-15)+.899, .3)+1n*v(n003)
b:u3:3 0 u3:n005 i=1u*dnlim(uplim(v(u3:x),v(+15)-.9,.5), v
+(-15)+.9,.5)+1p*v(u3:x)
c:u3:9 +15 n008 .5p rpar=1.12t noiseless
c:u3:4 n008 -15 .5p rpar=1.12t noiseless
c:u3:6 n003 -15 .5p rpar=1.12t noiseless
```

```
c:u3:10 +15 n003 .5p rpar=1.12t noiseless
a:u4:1 0 u4:n004 0 0 0 0 u4:x 0 ota g=150u iout=7u cout=28p en
+=9.8n enk=4 vhigh=1e308 vlow=-1e308
a:u4:2 n003 n001 0 0 0 0 0 0 ota g=0 in=.1p ink=70
c:u4:2 u4:n004 0 .75p rpar=100k noiseless
r:u4:1 +15 u4:x 10g noiseless
r:u4:2 u4:x -15 10g noiseless
m:u4:1 +15 u4:n005 n002 n002 u4:n temp=27
m:u4:2 -15 u4:n005 n002 n002 u4:p temp=27
d:u4:1 n002 u4:x u4:x
c:u4:3 u4:n005 0 .075p rpar=1meg noiseless
d:u4:4 n003 n001 u4:di
c:u4:5 n003 n001 1p rpar=80meg
c:u4:7 +15 n002 .2p
c:u4:8 n002 -15 .2p
b:u4:1 0 u4:n004 i=10u*dnlim(uplim(v(n001),v(+15)-.9,.3), v
+(-15)+.9, .3)+1n*v(n001)
b:u4:2 u4:n004 0 i=10u*dnlim(uplim(v(n003),v(+15)-.899,.3), v
+(-15)+.899, .3)+1n*v(n003)
b:u4:3 0 u4:n005 i=1u*dnlim(uplim(v(u4:x),v(+15)-.9,.5), v
+(-15)+.9,.5)+1p*v(u4:x)
c:u4:9 +15 n001 .5p rpar=1.12t noiseless
c:u4:4 n001 -15 .5p rpar=1.12t noiseless
c:u4:6 n003 -15 .5p rpar=1.12t noiseless
c:u4:10 +15 n003 .5p rpar=1.12t noiseless
v1 -15 0 -15
v2 +15 0 15
v3 v 0 pwl file=5v_lf_v.txt
v4 vs 0 pwl file=5v_lf_vs.txt
.model u4:di d(ron=1.15k roff=1g vfwd=1 vrev=1 epsilon=1
+revepsilon=1 noiseless)
.model u4:p vdmos(vto=300m kp=50m pchan)
.model u4:n vdmos(vto=-300m kp=50m)
.model u4:x d(ron=10k roff=1t vfwd=.82 vrev=.82 epsilon=.1
+revepsilon=.1)
.model u3:di d(ron=1.15k roff=1g vfwd=1 vrev=1 epsilon=1
+revepsilon=1 noiseless)
.model u3:p vdmos(vto=300m kp=50m pchan)
.model u3:n vdmos(vto=-300m kp=50m)
.model u3:x d(ron=10k roff=1t vfwd=.82 vrev=.82 epsilon=.1
+revepsilon=.1)
.model u2:di d(ron=1.15k roff=1g vfwd=1 vrev=1 epsilon=1
+revepsilon=1 noiseless)
.model u2:p vdmos(vto=300m kp=50m pchan)
.model u2:n vdmos(vto=-300m kp=50m)
.model u2:x d(ron=10k roff=1t vfwd=.82 vrev=.82 epsilon=.1
+revepsilon=.1)
```

```
112 .model u1:di d(ron=1.15k roff=1g vfwd=1 vrev=1 epsilon=1
    +revepsilon=1 noiseless)
113 .model u1:p vdmos(vto=300m kp=50m pchan)
114 .model u1:n vdmos(vto=-300m kp=50m)
115 .model u1:x d(ron=10k roff=1t vfwd=.82 vrev=.82 epsilon=.1
    +revepsilon=.1)
116 .tran 0 30ms .5ms .1ms startup uic
117 .end
```

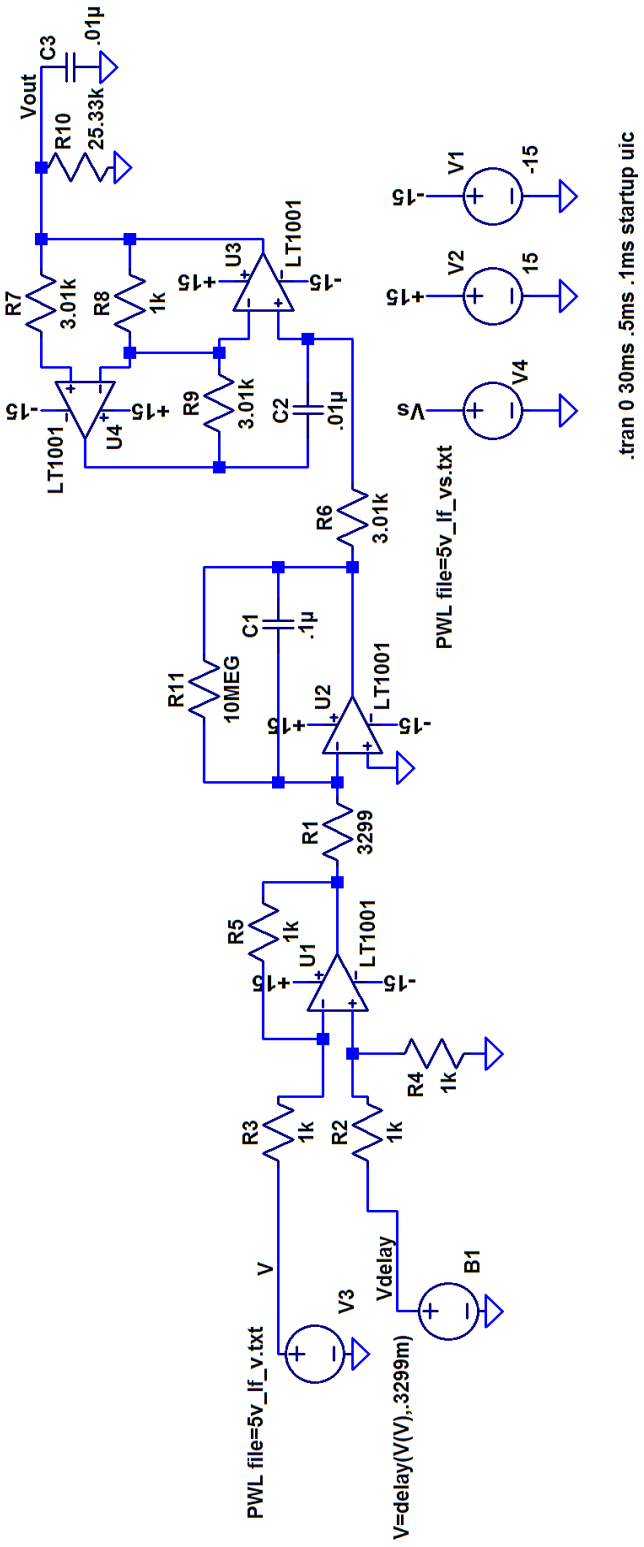

Figure B.1: Low Frequency Matched Filter Schematic

Appendix C
LF Oscillator Tuning Procedure

1 Remove capacitor C2.

2 Measure voltages at both the positive and negative input to the top comparator, U1.

3 Adjust the top potentiometer, modeled as R26 and R27, until both comparator inputs are as close to equal as possible.

4 Reinsert C2.

5 Measure the voltage at circuit nodes `V`, `Vd`, and `Vs`.

6 Adjust the left potentiometer, modeled as R3, until `V` appears to be chaotic. If this does not occur, skip to step **8**.

7 Verify chaotic operation by displaying `Vd` vs. `V` (this will generate an attractor).

8 Tune the right potentiometer, modeled as R23 and R30, until the circuit is operating in the desired band.

9 Repeat steps **6-8** as necessary for fine-tuning.