Constraint Satisfaction Modules: A Methodology for Analog Circuit Design

by

Piotr Mitros

B.S., Mathematics (2004)
B.S., Electrical Science and Engineering (2004)
M.Eng., Electrical Engineering and Computer Science (2004)
Massachusetts Institute of Technology

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2007

Author .....
Department of Electrical Engineering and Computer Science
August 30, 2007

Certified by .....
Gerald Jay Sussman
Panasonic Professor of Electrical Engineering
Thesis Supervisor

Certified by .....
Thomas F. Knight, Jr
Senior Research Scientist
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Constraint Satisfaction Modules:

A Methodology for Analog Circuit Design

by

Piotr Mitros

Submitted to the Department of Electrical Engineering and Computer Science
on August 30, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This dissertation describes a methodology for solving convex constraint problems using analog circuits. It demonstrates how this methodology can be used to design circuits that solve function-fitting problems through iterated gradient descent. In particular, it shows how to build a small circuit that can model a nonlinearity by observation, and predistort to compensate for this nonlinearity. The system fits into a broader effort to investigate non-traditional approaches to circuit design. First, it breaks the traditional input-output abstraction barrier; all ports are bidirectional. Second, it uses a different methodology for proving system stability with local rather than global properties. Such stability arguments can be scaled to much more complex systems than traditional stability criteria.

Thesis Supervisor: Gerald Jay Sussman
Title: Panasonic Professor of Electrical Engineering

Thesis Supervisor: Thomas F. Knight, Jr
Title: Senior Research Scientist

Acknowledgments

I would first like to thank Professor Gerald Jay Sussman. Gerry has been my mentor since I arrived at MIT as a freshman. He has always gone to great lengths to help me with problems, both research and personal. Without his help and guidance, this thesis would have been impossible. Gerry is a brilliant scientist, an amazing teacher, a great supervisor, and most importantly, a good friend.

Next, I would like to thank Tom Knight. I have known Tom for almost as long as Gerry. In that time, Tom has gone significantly out of his way for me a number of times. When I ran out of funding one semester, he was able to miraculously find new funding on short notice. At one point, he taught a class on digital circuit design simply because a couple of friends and I asked him to. He even rescheduled it to before he would normally wake up so that I could attend. Tom is an amazing person.

I'd like to thank Steve Leeb for agreeing to be on my thesis committee, for supervising my RQE exam, and otherwise being a great guy. In addition, I'd like to thank the many people with whom I've discussed this thesis, and who helped guide its direction. I'd like to single out Al-Thaddeus Avestruz, John Wyatt, Neil Gershenfeld, and Xu "Andy" Sun. In addition, I'd like to thank those professors who took the time to guide me before I began work on this thesis. In particular, I'd like to thank Hal Abelson for his guidance over the years, as well as his help in finding funding for the last two years of my undergraduate studies. I would like to thank Rafael Reif. Rafael supervised my research for a year. In that time, he was always kind and fair to his students, and his guidance was invaluable in choosing my research goals and professional career path. I also wanted to thank Anant Agarwal for his help with admissions to the Ph.D. program. Finally, I'd like to thank those who set me on this path, and in particular Walter Bender of the Media Lab (now One Laptop per Child), John Gibson of Landmark Graphics Corporation (later, Halliburton, and Paradigm), as well as the staff of the Research Science Institute, for their guidance and support while I was still in high school. Last, I would like to thank Jesus del Alamo for providing funding through the iLabs project while I did this research, and

for giving me the opportunity to see the world.

Finally, I'd like to thank my family. First of all, I'd like to thank Stefanie Tellex for her love and support over the past three years, as well as always being willing to help code before a deadline. I'd like the thank my sister Ania for always being there to listen to my personal problems, as well as for the many technical discussions about both of our dissertations. I'd like to thank my father, Jozef Mitros, who was always ready to assist me throughout the program with advice about my life, as well as with explanations of device physics and fabrication. Finally, I'd like to thank my mom, Katarzyna Mitros, for her moral support and her faith in me throughout the process.

Contents

List of Figures

Chapter 1

Introduction

This dissertation presents a methodology for the design of analog circuits. Under this methodology, the ports of all circuit blocks are bidirectional. The stability of circuits designed following this methodology can be determined by local, rather than global, analysis. If the circuit blocks follow a certain design discipline, they can be connected in nearly arbitrary configurations while maintaining stability. This methodology may be applied to designing circuits that solve several classes of problems. In particular, it is applied to designing a small circuit that can build a model of a memoryless nonlinear system, and to predistort to linearize this system.

First, this document shows how this methodology can be used to solve basic constrained optimization problems in analog circuits. Given a set of constraint equations, it shows how to build a circuit that will solve that set of equations. This circuit is sufficiently small that it can be used for simple analog processing in basic analog circuits; for instance, it may be used to set optimal bias currents for another circuit.

Next, this thesis shows how to use this methodology to build a circuit that can be used to determine the parameters of a model of a system through observation of the system's behavior. This circuit can, for instance, be used to monitor and model the nonlinearity of the power output stage of an amplifier. Given this model, a symmetric, matched circuit can predistort to compensate for the nonlinearity.

These circuits are small enough that they can be used as blocks for other analog circuits, particularly in places where digital processing is impossible due to size and

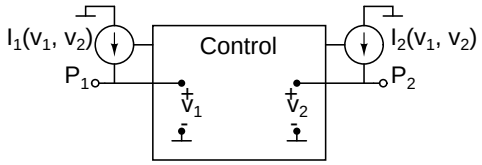


Figure 1-1: A block-diagram of a 2-port element. A controller monitors the voltage at the adjacent nodes, and injects current into both nodes if a constraint is not met.

cost constraints. They also lend themselves well to automated synthesis.

1.1 System Overview

This section begins with an overview of the constraint equation solver, and later develops it into the modeling and linearization circuits.

1.1.1 Constraint Optimization Solver

The constraint optimization circuit solves systems of simultaneous equations and inequalities. Given n constraints, the circuit is composed of n blocks, each corresponding to a constraint. Each of these blocks has m_i ports, where m_i is the number of variables in the ith equation. Each block then attempts to hold the constraint corresponding to this equation by measuring the voltage on each port, and generating an error current feeding back into that port. This error current moves the voltages in the correct direction to meet the constraints. This can be viewed as a generalization of the concept of a transformer. The block-diagram of a 2-port block is shown in Figure 1-1.

Each block is assigned an objective function $L_i(V_{P1}, V_{P2}, ..., V_{PN})$ that describes how well the constraints in each block are met (e.g., the square distance from the constraint). L_i achieves its minimum when the constraints are completely met. The current output from each block through each port is then proportional to $-\frac{dL_i}{dV_{Pi}}$ where V_{Pi} is the voltage seen at that port[1]. Inequalities are approximated by crafting

[1]This may also be implemented with the dual circuit, swapping currents with voltages, and capacitors with inductors.

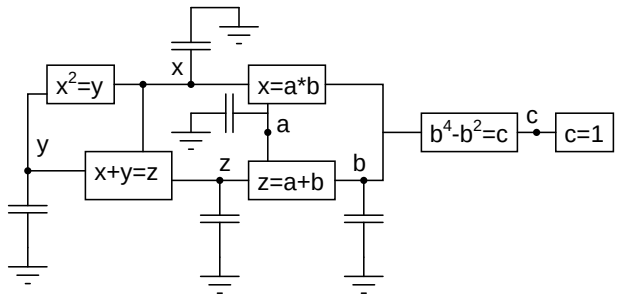


Figure 1-2: A system for solving a network of equations consisting of connected constraint blocks.

appropriate objective functions.

Take the constraint $V_{P1} = 2 \cdot V_{P2}$ (in other words, a block behaving as a 2:1 transformer). One possible objective function is $L_i(V_{P1}, V_{P2}) = (V_{P1} - 2 \cdot V_{P2})^2$. Then, to implement this constraint, the block shown in Figure 1-1 should be implemented with a controller that will output the current equal to $I_{P1} = -\frac{dL}{dV_{P1}} = -2V_{P1} + 4V_{P2}$ and $I_{P2} = \frac{dL}{dV_{P2}} = -8V_{P2} + 4V_{P1}$. Notice that, as with a physical 2:1 transformer, the currents have a 1:2 ratio.

These blocks can tie together into arbitrary networks to solve more complex constraint problems. For instance, the network shown in Figure 1-2 will solve the set of equations:

$$x^2 = y, \quad x + y = z, \quad z = a + b$$
$$c = 1, \quad x = ab, \quad b^4 + b^2 = c$$

If the objective function associated with this set of equations were convex and had a unique solution, this network would find that solution. In this case, assuming least squares objective functions, the objective function is not convex. It has a discrete set of local minima, and therefore the network will converge to one of these local minima. If the system were underconstrained such that the objective function had a connected subspace where it was minimized, the system would enter a minimum, but could drift through the subspace associated with that minimum.

If the system of constraints is strictly convex, systems of this form are stable. This criterion is sufficient, but not necessary — a number of non-convex constrained and

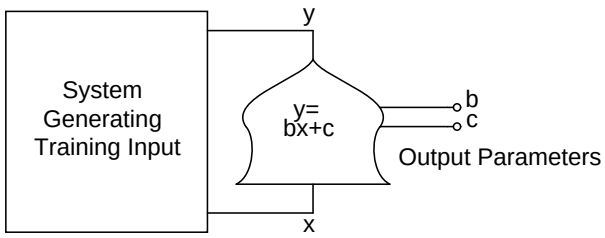


Figure 1-3: This is a function fitting circuit. It consists of an underconstrained constraint block monitoring a system. In this case, the system will find the parameters b and c of the affine model $y = bx + c$ of the system by iterated gradient descent.

overconstrained systems can be solved as well.

1.1.2 Modeling

This section describes how to build circuits that can model systems by observation. The circuits described can be connected to multiple terminals, monitor those terminals, and develop an approximate model of the relationship between the voltages on those terminals. The circuit is primarily useful for memoryless models, but can model a variety of functions, including a superset of general linear regression.

The model parameters are found using the same type of circuit as described in Section 1.1.1, but operating over a set of underconstrained equations. An example of this procedure is shown in Figure 1-3.

Here, as x and y change, the system continuously moves b and c towards the line defined by $y = bx + c$. This procedure is a form of iterated gradient descent. For a large class of models (a superset of general linear regression) it will find what is, in at least one sense, an optimal estimate of the parameters of the model of the original system.

1.1.3 Linearization

Once one can function-fit a nonlinearity, it is often straightforward to invert it, as shown in Figure 1-4. Here, the top block builds a model of the nonlinearity, while the bottom block inverts the function. Since the blocks are bidirectional, the top block

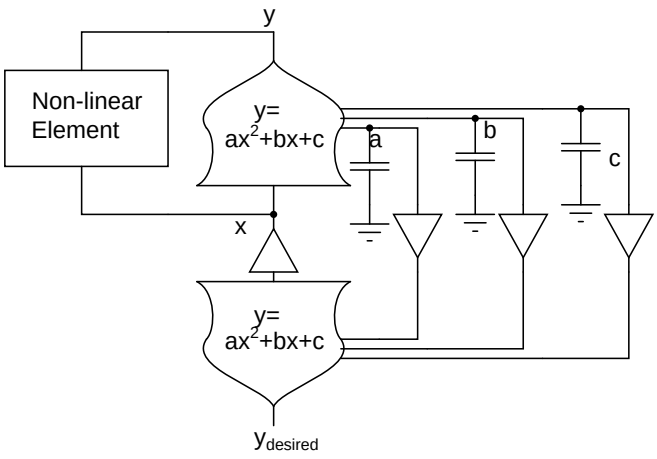


Figure 1-4: Shown is a circuit that can build a model of a nonlinearity, and predistort to linearize it. Since the modeling circuit is bidirectional, an identical, matched circuit is used to implement the model, predistorting to linearize the nonlinearity.

and bottom block may be implemented as identical, matched circuits. The buffers in Figure 1-4 are not necessary. They are included in the figure primarily to show direction of flow of information, and to simplify explanation and analysis.

One caveat is that not all models can be inverted. While monotonic models are invertible, non-monotonic models may have multiple local minima. In that case, the matched block may become trapped in one of those minima.

It is possible to compile most circuits designed under this methodology into a more efficient implementation. Many blocks contain redundant components already contained in adjoining blocks, which may, instead, be shared. In addition, for many constraints circuits, approximations can significantly reduce circuit complexity. Chapter 4 will work through an example of this procedure by taking a simple second order Taylor series linearizer circuit, and compiling it down to an implementation that requires about a dozen major components (operational amplifiers or Gilbert multipliers). This implementation will also further improve matching, since many calculation will be shared among blocks.

As shown, this technique is useful for controlling nonlinear memoryless systems. There are natural ways of extending the work to systems with memory, but these are not yet fully developed, and so will not be explored in this thesis.

1.2 Motivation

One goal of this work is the exploration of analog techniques that scale to more complex systems. The use of a local stability criterion may allow the construction of much more complex analog circuits than would be possible with traditional loop-shaping techniques. In addition, the work explores several other areas of unconventional circuit design, such as breaking the input-output abstraction barrier.

1.2.1 Analog Architectures

Moore's Law scaling has been much more generous to digital systems than to analog systems[1]. Moore's Law benefits circuit designers in two ways:

- More transistors
- Better transistors

While analog circuit design has been able to take advantage of better transistors, it has not been able to exploit the advantages brought by the increased number of transistors as well as digital electronics has[2]. There are a number of reasons for this problem, but probably the most important one is that existing analog circuit design methodologies are optimized for extracting optimal performance from circuits consisting of a small number of devices. As a result, they do not scale to very large circuits. The problem of determining stability grows very quickly with the complexity of the circuit. Simulators do not perform well on complex circuits[3]. As a result, almost all progress in circuits is either from device improvements, or small topology optimizations. Compared to digital, little progress (beyond integration) comes from advances at an architectural level.

The utilization of more devices is grossly inefficient. For instance, to achieve better matching of devices in an operational amplifier, the standard approach simply doubles the size of all devices, effectively building two operational amplifiers, connected

[2]There are some exceptions to this generalization. For instance, flash ADCs use large amounts of parallelism to achieve high conversion rates.

[3]Their poor performance is most likely due, in part, to their inability to extract simple models of subcircuits.

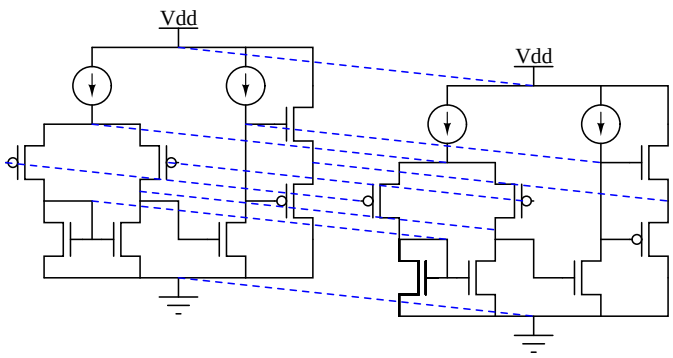


Figure 1-5: In traditional circuit design, to improve matching between components, devices are simply increased in size, or multiple devices are placed in parallel, as shown in this double-width operational amplifier.

together at all matching nodes, as shown in Figure 1.2.1. This design technique is almost certainly not the most efficient way to design a circuit. One could, instead, try connecting two different operational amplifier designs, one with high bandwidth, and the other with low distortion, in some clever way to build a single operational amplifier with high bandwidth, but low distortion at low frequencies.

Many techniques for exploiting complexity exist. Multiple amplifiers on a transmission line, to form a distributed circuit, as shown by Percival[27] and popularized by Hajimiri[15], can substantially improve bandwidth over traditional techniques. Distributed circuits can route signals outside the normal signal path to introduce an effective negative time delay. Kim[23] built a high-performance ring oscillator based on this technique. Systems may also include large amounts of additional analog computation around a conventional circuit in order to fine-tune bias levels, cancel out distortion, and do other types of processing to aid in circuit operation. Similar tricks are commonly used in digital circuits, where advanced branch prediction algorithms and LRU cache schemes are placed outside of the main signal path to maximize the performance of a simple microprocessor core. Similarly, a processor may have multiple execution units, or more recently, multiple cores, and route tasks to whichever is idle. This sort of efficient use of parallelism is rare in analog design.

Hundreds of similar techniques that tradeoff circuit complexity for better performance exist. Each is very effective in isolation. Few circuits, however, exist that

exploit a significant number of these techniques — designing such a circuit, and furthermore, determining its performance and demonstrating its stability is simply too hard.

The field of electronic engineering requires a new set of methodologies, abstractions and mathematics that will allow the design and synthesis of complex, interconnected, stable systems, and the characterization of their performance. In this paper, I present one such methodology. I believe that other methodologies may also be created, especially deriving from work in distributed dynamics of complex systems, and in particular, from the areas of SIMD systems, amorphous computing, and dynamics of two-dimensional physical systems. These methodologies, or others like it, may eventually allow the construction, or perhaps more importantly, the automatic synthesis of complex circuits that will achieve drastically higher performance than traditional analog design techniques.

1.2.2 Breaking the Input-Output Abstraction

Traditional circuits are built based on an input-output abstraction. Analog circuit design relies on guard-rings and other complex structures to minimize coupling between parts of the circuit. Feed-through from the output of an amplifier to its input is generally viewed as an undesired interfering signal.

Especially in high-end RF systems (in particular in MIMO), feed-through very significantly limits performance. Conventional approaches try to solve this by minimizing the level and effect of feed-through. An alternative way to deal with this problem is to try to control and understand the effects of feed-through, and potentially try to exploit them.

S-parameter models do this to a limited extent, but are primarily used in LTI systems, such as gain blocks or filters, rather than for nonlinear blocks that perform actual computation, such as modulation or demodulation. Treating all ports as both inputs and outputs may eventually give rise to a powerful new set of circuit design techniques where feed-through is a desirable, controlled design parameter, rather than an undesirable side effect of imperfect isolation.

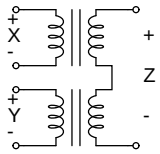


Figure 1-6: The constraint $ax + by = z$ expressed with transformers.

1.3 Background

This work is based on the traditional technique of solving systems of linear equations using transformers. This method was introduced by Mallock[24], and was common use in the early days of analog computing, but died off as the input-output abstraction took over in the forties and fifties. Seidel and Knight revived it by reimplementing this type of circuit in an IC using switched capacitor transformers[29]. They expressed constraints such as $ax + by = z$ using a pair of transformers, as shown in Figure 1-6. Their circuit consisted of multiple transformer blocks like this and solved arbitrary linear equations. This approach allowed for some basic nonlinearities — for instance, diodes could be used to express inequalities — but as presented, it did not directly scale to arbitrary nonlinear constraints.

My approach is a direct extension of Seidel and Knight's work. My first step was to replace the switched capacitor transformers with active transformers. This was not entirely novel — for instance, Chua, *et al.*[7] build a transformer with many ports, and showed that it could be used for mathematical programming. Developing active transformers lead to a number of issues. First, while physical transformers are passive, an active circuit approximating a transformer may no longer be passive if the implementation is off by an arbitrarily small ϵ. Therefore, stability can no longer follow from passivity. Second, the definition of a transformer in terms of voltage and current ratios is slightly ambiguous. If a 2:1 transformer is driven with 1V on one side, and 1.5V on the other side, it needs to output reasonable currents. Whether those currents should be in a ratio of 1.5:1, or 2:1, or otherwise, falls outside of the traditional definition of a transformer. These issues lead to the general formulation of transformers found in this thesis. This formulation includes the non-linear (and not

necessarily energy-conserving) generalizations thereof. Finally, once this definition was in place, I noticed that this approach generalized to the problem of modeling, and therefore, linearization.

There is a wide body of prior work that attempts to solve problems similar to the one presented in this thesis. Researchers from a variety of fields have, independently, worked on circuits similar to those in this thesis, circuits that solve similar problems in other ways, or worked on mathematics relevant to these circuits in domains outside of electronics.

Dennis's seminal Ph.d. thesis[11] was one one of the first papers on solving mathematical programs with electrical networks. In his thesis, Dennis proves that any direct current network made up of voltage and current sources, ideal DC transformers, and ideal diodes is equivalent to a pair of dual linear programs. A network made up of linear resistors together with these elements is equivalent to a pair of dual quadratic programs. Conversely, he shows that any linear or quadratic program may be modeled by an electrical circuit. His thesis begins to approach the problem of more general nonlinear mathematical programming, taking two significant steps. First, he integrates a class of nonlinear resistors into his framework. Second, he shows how his work with linear components can be used to compute the direction of steepest descent. His circuits are much more compact than those in this thesis. He does not, however, approach the level of generality of this work.

The most important (although, unfortunately, not best known) work on solving general nonlinear mathematical programs in analog circuits is Chua's canonical nonlinear programming circuit[6]. Chua developed a system for solving nonlinear programs that is very similar to the static case of the constraint solver in this thesis — the mathematical model of an individual constraint block is essentially the same as in my circuit, although the mathematical foundations are slightly different, as are the circuit implementations. Chua's original paper was limited in the way that it treated stability – Chua shows that the system with no dynamics and no node capacitances would have an equilibrium operating point at the solution of the mathematical program, but other than citing general stability criteria from a previous paper[5], did not

give a way to guarantee that the equilibrium will be stable. Stability analysis, rather, had to be done on a circuit-by-circuit basis. In a later paper[20], Kennedy and Chua added node capacitance, and so were able to demonstrate stability in the general case in a manner very similar to that in this document. Further analysis of stability in the linear case may be found in Song[30], who performs a stability analysis from a control system perspective.

There are two minor extensions to Chua's work. Chua[3] published a paper showing how constraints and inequalities may be implemented within his framework. In this paper, he expresses constraints like $f(x) > 0$ as $y = |f(x)|$ or $y = f(x)^2$, and minimizes y according to his framework. Jayadeva, *et al.*[16] point out several problems with this approach. These problems mostly center around convex problems becoming non-convex and having multiple local minima, choice of how much weight should be given to the constraints, and computational complexity. To a large extent, these problems are more relevant in computer science contexts, where optimizers solve complex, poorly understood numerical optimization problems, than they are to circuits, where the problems are likely to be simpler, and better understood. The authors propose a solution, but it is not obvious how easy it would be to apply in a circuits context. Forti, *et al.*[14] extend Chua's work, somewhat theoretically, to non-smooth mathematical programming problems.

I found this work in fairly late stages of my thesis, and was pleasantly surprised to find that high quality work had been pursued in the area before. My thesis extends on Chua's in several ways. It is somewhat more general, especially in that it targets dynamic systems. The stability proofs in Chua's work (even in the latter papers) are also somewhat less developed.

1.3.1 Neural VLSI Systems

A large number of similar systems exist in the field of neural VLSI systems. There are dozens or hundreds of papers in this area, mostly of very low quality, so it is difficult to give a concise summary. The most important (and rather good) paper in this area is by Tank and Hopfield[17]. This paper presents a way of solving linear programming

problems using a circuit that consists of neural summing nodes, and inspired the vast majority of papers on solving constraints problems with neural VLSI. Chua published a careful analysis of Hopfield's circuit[19] that shows that it is nearly identical to a special case of the general solver shown in Chua's canonical nonlinear programming circuit[6], as well as, by extension, this thesis. The circuits differ in two significant ways. First, Tank's circuit is modularized by neurons, rather than by constraint blocks. This modularity makes it difficult to implement general nonlinear constraints, and Tank's circuit is therefore limited to linear (and very limited nonlinear) programs. Second, Tank's way of proving stability requires all variable nodes to have resistors to ground. This causes the circuit to generate approximate, rather than exact, solutions.

Due to the difficulty of computing derivatives, a number of followup VLSI circuits rely on injecting noise to compute derivatives. For instance, Jelonek[18] created a system that uses an analog technique based on simulated annealing. First, large amounts of noise are added to come close to the global minimum. Smaller amounts of noise are then added to perform basic gradient descent to find a local minimum. In addition, a neural network holds constraints. The resulting circuit is moderately complex, and as a result, only simulation results are available.

Learning by gradient descent is a standard technique in the field of neural networks. Two main techniques are back propagation and weight propagation. A good overview of neural network techniques may be found in any introductory artificial intelligence text, such as Winston[32] or Russell, Norvig[28]. These texts tend to be clear and concise, and put neural networks within the broader context of machine learning. Most neural network texts, in contrast, while more detailed, tend to understate the limitations of neural networks, and do not adequately contrast them with other, similar machine learning techniques. The best work describing what perceptions, the building blocks of neural networks, can and cannot do is by Minsky and Papert[26].

VLSI systems have been built around virtually all computer science neural network techniques. Most of these systems may be used for building models, and several can solve constraint problems. There is also a fairly large field of neural network-based

controllers. Since there is such a wide variety of work, it is difficult to make general comparisons to my work. One major limitation of these systems, in general, is that performance is difficult to characterize. While neural networks often perform very well, their operation is poorly understood. As a result, it is difficult to predict in what contexts they will work well, and the overall function of the system can often only be characterized experimentally. The problem is that the model being fitted to is not well characterized, or, in some cases, even deliberately expressed. Even if the model space is understood, it is often difficult to predict whether a given system will reach the global minimum, or some local minimum. As a result, in many cases, it is difficult to build a neural-network based system with deterministic performance.

1.3.2 Passivity and Other Stability Methodologies

There are a number of other criteria for designing complex, interconnected, stable circuits. Digital circuit design uses a methodology that relies on locally stable elements and, within a given clock cycle, a one-way flow of information. Analog circuits may be integrated the same way — stable local elements with a one-way flow of information — but this is usually only used to achieve greater system integration.

Passivity is a common criterion for achieving system stability. The word "passive" has several meanings, depending on the domain. Each meaning has a different, associated set of stability theorems. In control systems and circuit network theory, a passive device is one that consumes, rather than produces, energy. Examples include devices like transistors, tunnel diodes, and glow tubes, but exclude voltage and current sources. In contrast, in circuit design, passive devices are ones that are incapable of power gain. In circuit design, transistors, glow tubes, and tunnel diodes are considered active, whereas voltage and current sources are considered passive.

Passivity is a very powerful tool for demonstrating stability. It is used in a number of domains, including filter design, and control system design. Nevertheless, it is often inadequate for simulated passive devices, such as the ones in this thesis. As shown in the active transformer example, when active devices simulate passive ones, small variations from the ideal may make a system unstable, even if all devices are ideally

passive. This is especially a problem at higher frequencies, where actual behavior necessarily diverges significantly from desired behavior.

Control Systems Passivity

For the purposes of this discussion, define a resistor as a nonlinear memoryless 1-port[4]. A resistor is considered passive iff $vi \geq 0$ for all points (v, i) on its characteristic. It is considered strictly passive iff $vi > 0$, except at the origin.

The voltage within a network of passive resistors cannot exceed the voltages presented on the terminals. The current within any resistor may not exceed the total current input through all ports. This is a strong bounded input/bounded output (BIBO) stability criterion.

Informally, an arbitrary n-port, with memory, is considered passive if it can store or dissipate energy, but cannot create energy. The formal definition varies between texts, and is often incorrect in the way in which initial conditions are handled. The best definition available is found in Wyatt[33]. Wyatt also explains the problem with other definitions. An abbreviated version of this definition follows.

The available energy from an n-port in state x is defined as:

$$E_A(\mathbf{x}) = \sup_{\mathbf{x} \to T \geq 0} \left\{ \int_0^T - \langle \mathbf{v}(t), \mathbf{i}(t) \rangle \right\}$$

Where the notation $\sup_{\mathbf{x} \to T \geq 0}$ indicates the supremum is taken over all $t \geq 0$ and all admissible pairs $\mathbf{v}(t), \mathbf{i}(t)$ with the initial state $\mathbf{x}$. An n-port is considered passive if $\forall \mathbf{x}, E_A(\mathbf{x}) < \infty$.

A network consisting entirely of passive elements is itself passive, and therefore, only a finite amount of energy can be drawn from it.

A resistor is considered increasing iff $(v' - v'')/(i' - i'') \geq 0 \forall (v', i'), (v'', i'')$, and strictly increasing iff $(v' - v'')/(i' - i'') > 0 \forall (v', i') \neq (v'', i'')$. A network consisting of strictly increasing two-terminal resistors and independent sources has at most one

[4]This definition varies between texts – many circuit theory texts consider all memoryless n-ports to be resistors, while many traditional circuit design texts require resistors to be linear memoryless 1-ports.

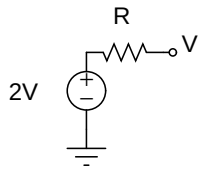


Figure 1-7: This circuit is a counterexample that shows that stability could not be shown by controls passivity. The voltage source shown holds the constraint $V = 2$, minimizing the objective function $(V - 2)^2$. It is not, however, passive, since it may output energy.

solution. Furthermore, the slope of the transfer function $\frac{v_{in}}{v_{out}}$ of a network made of strictly increasing resistors cannot exceed unity. This property is also sometimes called (strict) monotonicity and (strict) incremental passivity.

A formal discussion of passivity and monotonicity, although limited to memoryless 1-port elements, may be found in Chua, et al[4]. This is an excellent reference on the topic. Desoer and Kuh[12] proves the stability of networks consisting of capacitors, inductors, and nonlinear passive resistors. Cruz and Valkenburg[9] touches on multi-port elements, but does not develop very much theory about them.

Passivity forms a powerful discipline for designing stable systems. Indeed, it is one of the standard methods used in control systems. Khalil[21] includes a very accessible text on passivity in control systems. Although the discussion is inspired by circuits, the text primarily focuses on the design of stable control systems for traditional controls applications, rather than for electronic circuit design. Vidyasgar[31] is a more rigorous control systems book that also discusses stability by passivity, although the discussion is shorter than that in Khalil.

Circuits designed under the methodology presented in this dissertation are not passive in the control systems sense. A circuit implementing the constraint such as $x = 2$, with a least squares objective function, is simply a Thevenin voltage source, as shown in Figure 1-7. This block is trivially not passive in the controls sense, since it outputs power.

Circuit Design Passivity

In circuits terminology (in contrast to control systems), passive devices are ones not capable of gain, whereas active devices are ones capable of gain. For two terminal devices, this definition is identical to the one of monotonicity given above. I have not been able to find a formal definition for multi-terminal devices, but a suggested definition is that the small signal model is thermodynamically passive. Active devices, in circuit design terminology, include many dissipative devices capable of gain that would be considered passive by control and network theorists, including two-terminals such as tunnel diodes, glow tubes, and multi-terminal devices, such as transistors, relays, and tubes.

In addition to being another methodology for determining stability, the work in this thesis is built on passivity (in the circuit sense) – the work on solving constraints with transformers[29] upon which my work is based showed stability by passivity.

Nevertheless, even lacking a good definition for multi-port devices, it is easy to show that the methodology allows for non-passive components by constructing an active one-port. Take a circuit implementing the constraint associated with the objective function $L(x) = x^4 - x^2$. This constraint, $x \in \left\{\frac{-1}{\sqrt{2}}, \frac{+1}{\sqrt{2}}\right\}$, is shown graphically in figure 1-8. This objective function has two minima — $\pm\frac{1}{\sqrt{2}}$. It has the voltage-current relationship $I = 4V_x^3 - 2V_x$, shown in Figure 1-9. This is non-monotonic. Indeed, connected in parallel with an inductor and capacitor, it acts as an oscillator, as shown in Figure 1-10.

The methodology presented in this thesis is, also a generalization of the concept of a memoryless monotonic one-port. Given a memoryless monotonic one-port, with I-V characteristics $i(v)$. Let:

$$L(x_1, x_2) = \int_0^{x_2 - x_1} (x_1 - x_2) i(x_1 - x_2 - v) dv$$

Then,

$$\frac{dL}{dx_1} = i(x_1 - x_2)$$

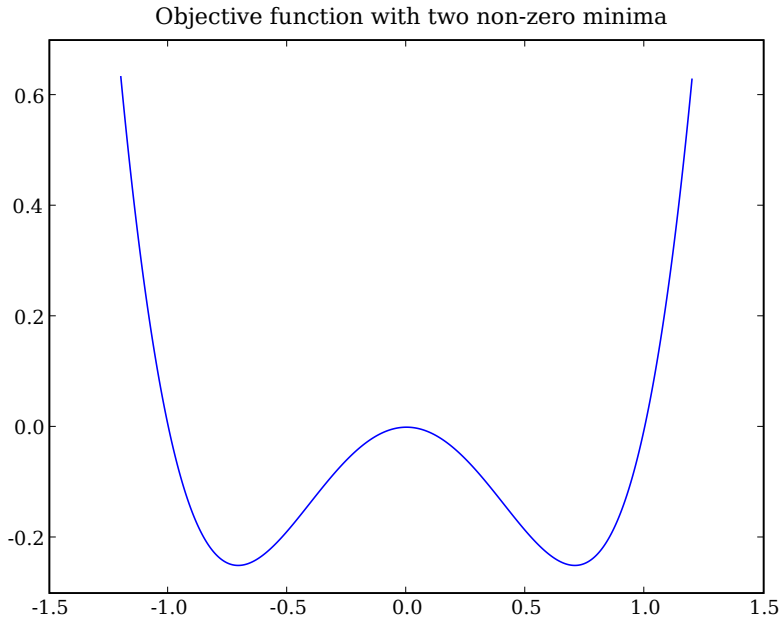


Figure 1-8: Shown is an example of an objective function with two non-zero minima. A circuit implementing such an objective function will be non-passive in both senses.

And similarly,

$$\frac{dL}{dx_2} = -i(x_1 - x_2)$$

Therefore, all incrementally passive one-ports are valid constraint block within this thesis methodology, and this thesis may be viewed as one generalization of the concept of incremental passivity[5].

1.4 Limitations

This approach is only proven to satisfy constraint optimization problems with unimodal objective functions. Efficient general-purpose optimizers do not exist since the problem is NP-hard; it is trivial to reduce 3SAT to an optimization with multiple local minima[6]. Practically, however, many non-convex systems may also be solved. The

[5]Although, unlike incrementally passive one-ports, the stability proofs for the methodology in this thesis do not extend to inductors.

[6]To show this, transform the original 3SAT problem into a continuous version by taking:

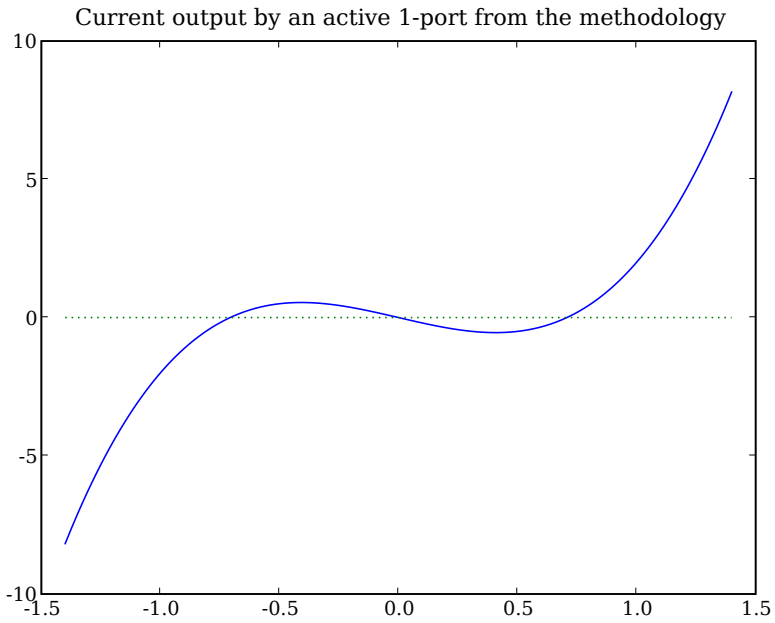


Figure 1-9: Shown are the IV characteristics of a circuit whose objective function has two stable minima. As a result, the IV-curve has two stable non-zero equilibrium points. Notice that the curve is non-monotonic, and is in all four quadrants.

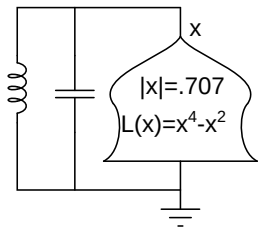


Figure 1-10: This circuit may be used as a counterexample to show that stability could not be shown by circuit passivity. This circuit has two non-zero optima. It will oscillate if it is connected to an LC tank.

system will still converge if it is started near a local minimum. In addition, in many cases, it is possible to apply analogues of numerical techniques such as momentum (reducing stability slightly), or simulated annealing (adding noise).

1.5 Document Overview

Chapter 1 is the introduction. Chapter 2 will explain the circuit for solving constraint problems, and demonstrate its stability, first at low frequencies with ideal circuits, and then in non-ideal circuits with errors in output current and limited bandwidth. It will also give an example of a constructed constraint propagation circuit. Chapter 3 will explain how to use the methodology for function fitting, and demonstrate the level of stability expected from it. Chapter 4 will show how the function fitting circuit can be used to linearize a nonlinearity, and furthermore, show a way to compile this circuit into a more efficient implementation that reduces overlap between the modeling portion and the linearization portion. The thesis will conclude with a discussion of possible future directions for the work in Chapter 5.

$$
\begin{array}{rcl}
\text{TRUE} & \longrightarrow & 0 \\
\text{FALSE} & \longrightarrow & 1 \\
\neg A & \longrightarrow & 1 - A \\
A \vee B & \longrightarrow & A \cdot B \\
A \wedge B & \longrightarrow & A + B
\end{array}
$$

The system is satisfiable iff the resulting formula has a minimum of 0.

Chapter 2

Static Case — Constrained Problems

This chapter will demonstrate circuits that can solve many nonlinear mathematical programming problems. The mathematics found in this section is very similar to those found in Chua[6][20]. Formally, a nonlinear mathematical programming problem can be expressed as finding values for the components of the vector $\mathbf{x}$ that minimizes the function L:

$$\min_{\mathbf{x}} L(\mathbf{x})$$

Subject to a set of constraints:

$$\begin{cases} g_i(\mathbf{x}) = 0 & : \quad i = 1, ..., m \\ g_i(\mathbf{x}) \geq 0 & : \quad i = m+1, ..., n \end{cases}$$

Practically, the constraints are usually relaxed in some way. For instance, the above equations may be approximated as:

$$\min_{\mathbf{x}} \left[L(\mathbf{x})^2 + \sum_{i=1}^{m} g_i(\mathbf{x})^2 + \sum_{i=m+1}^{n} \max\left(g_i(\mathbf{x}), 0\right)^2 \right]$$

Note that the above can approximate the original formulation arbitrarily closely by

scaling the functions g_i by arbitrarily large factors. The approach in this thesis requires some (although not necessarily the above) relaxation.

2.1 Design Discipline

The procedure for designing a constraint block is as follows:

1. Choose an objective function. For a constraint of the form $a = b$, the objective function $(a - b)^2$ is a good choice. There is an infinite number of possible objective functions. If the original constraint is $x = y^2$, valid choices would include $L = (x - y^2)^2$, $L = (y - \sqrt{x})^2$, $L = (x - y^2)^4$, and a variety of others. Choice of objective function will have a significant effect on the dynamics. For the ith constraint block, call this objective function L_i.

2. Calculate derivatives of the objective function with respect to all variables that L depends on. In the above case, $\frac{dL}{dx} = \frac{\partial}{\partial x}(x - y^2)^2 = \frac{\partial}{\partial x}(x^2 - 2xy^2 + y^4) = 2x - 2y^2$.

3. Design a circuit that has a port for each variable. Represent that variable with the voltage on that port, V_j. The circuit must output a current I_j proportional to the derivative $\frac{dL_i}{dV_j}$ onto the port associated with V_j:

$$I_j = -\alpha \frac{dL_i}{dV_j}$$

   Where α is a positive constant. Without loss of generality, for the rest of the discussion, assume $\alpha = 1$.

To construct constraint solvers, simply connect these blocks together. To insure high-frequency stability, place adequately large capacitors on all the interconnect nodes to guarantee that low-frequency design dynamics dominate.

The easiest way to understand this procedure is to work through several examples. A number of examples of this procedure are shown in Appendix C.

2.2 Stability Analysis

Two arguments will show stability — one at low frequencies, and one at high frequencies. At low frequencies, stability follows from a Lyapunov-type argument. In this case, the sum of the local objective functions of the local blocks forms a global objective function. This global objective function will be shown to be monotonically decreasing with time. If the global objective function is unimodal, LaSalle's theorem proves global stability. This argument is valid at frequencies where the system outputs currents within a small error of the current dictated by the model. In any real implementation, due to limited bandwidth, high-frequency behavior will differ significantly from ideal behavior. Since low frequency stability will not depend on the node capacitances, it is possible to have arbitrarily small high frequency gain by increasing these node capacitances. At high frequencies, stability follows from an argument analogous to the small gain theorem.

The global objective function is just the sum of the local objective functions for each constraint block. Following the design constraint that the current output on node j by block i is proportional to $-\frac{\partial L_i}{\partial V_j}$, the current injected onto the jth node is:

$$-\phi_j = -\sum_{i \in \mathcal{N}_j} \frac{\partial L_i}{\partial V_j}$$

Where $\mathcal{N}$ is the set of constraint blocks connected to the node. The rate of change in voltage on the jth node is:

$$\frac{dV_j}{dt} = \frac{-\phi_j}{C}$$

The change in the objective functions surrounding the node, as a result of a change in voltage on that node, assuming the voltages on the other nodes are fixed[1], is just:

$$\frac{\partial L_{\mathcal{N},j}}{\partial V_j} = \phi_j = \sum_{i \in \mathcal{N}_j} \frac{\partial L_i}{\partial V_j}$$

Therefore, the resulting rate of change in the sum of the objective functions surround-

[1]This kind of node-wise analysis follows from continuity.

ing the node is:

$$\frac{\partial L_{\mathcal{N},j}}{dt} = \frac{dV_j}{dt}\frac{\partial L_{\mathcal{N},j}}{\partial V_j} = \frac{-\phi_j^2}{C} \leq 0$$

The overall change in global objective function is just the sum of the change in global objective function due to the change on each node:

$$\frac{dL}{dt} = \sum_j \frac{\partial L_{\mathcal{N},j}}{dt} \leq 0$$

It follows that the global objective function L is monotonically decreasing. Notice that this stability criterion is independent of the capacitance values on the joint nodes.

By LaSalle's theorem[2], the system will converge to some set $S \subseteq \{x|\dot{L}(x) = 0\}$. This shows stability in the fully constrained case, as well as the overconstrained case with a single global minimum. In the underconstrained case (or any case where there is a set of points that form minima), it shows that the system will converge to the minima, but says nothing about the behavior within the set that forms the minima.

2.2.1 Robust Low Frequency Stability

The argument, thus far, demonstrates stability only in the case of an ideal system. Due to component mismatches, limited frequency response, and other limitations, the system may be not behave exactly as desired. Indeed, for any ϵ, a system can be constructed that is stable in the ideal case, but is unstable if the currents are off by ϵ. The active transformer, connected to an arbitrarily large resistor, is an example of such a system.

Let us assume that, in addition to the current dictated by our design discipline, each node sees an additional error current less than some constant σ. Here, the total current into the node is:

$$\sigma - \sum_{i \in \mathcal{N}} \frac{\partial L_i}{\partial V_i} = \sigma - \phi_j$$

[2]Readers unfamiliar with LaSalle's Theorem should refer to Appendix B. Readers unfamiliar with LaSalle's theorem, but familiar with Lyapunov stability, may informally substitute Lyapunov for LaSalle, although the formalism will be very slightly incorrect.

The overall change in objective function is:

$$\frac{dL}{dt} = \sum_j (\sigma - \phi_j)\phi_j = \sum_j \sigma\phi_j - \phi_j^2$$

As long as:

$$\left|\sum_j \sigma\phi_j\right| \leq \left|\sum_j \phi_j^2\right|$$

The system will continue to be stable. For ease of analysis, this bound can be weakened to:

$$\left|\sum_j \sigma\right| \leq \left|\sum_j \phi_j\right|$$

Or alternatively, to

$$\left|\sum_j \sigma\right| \leq \sqrt{\sum_j \phi_j^2}$$

This will, for most systems, not be the case globally, since the slope of the objective function will go to zero around the minimum, and so ϕ may be arbitrarily small. Define the region B_1' where this criterion does not hold. B_1' is shown graphically in Figure 2-1. Next, define $S = \sup_{B_1'} L$, $B_1 = \{\mathbf{x}|L(\mathbf{x}) \leq S\}$. Graphically, taking the level curves of the objective function, shown in Figure 2-2, B_1 is the smallest level curve that entirely contains B_1'. This is shown in Figure 2-3. If $\|\mathbf{x}\| \to \infty \Rightarrow L(\mathbf{x}) \to \infty$ then B_1 is bounded. If it L is also unimodal, than B_1 is connected.

Notice the objective criterion holds everywhere outside of B_1. Define a new objective function:

$$L'(\mathbf{x}) = \begin{cases} (L(\mathbf{x}) - I)^2 & : \quad \mathbf{x} \notin B_1 \\ 0 & : \quad \mathbf{x} \in B_1 \end{cases}$$

This will guarantee that $\mathbf{x}$ will be stable outside of B_1, converging towards B_1, and once inside B_1, will remain there. Its behavior within B_1 is unknown. For most practical objective functions, B_1 can be made arbitrarily small. This shows that, for a good objective function, the system will go to the minimum, and stay near the minimum, and so will be nearly stable (limited to small, low-frequency oscillations). Large plateaus mean that the system may have a large region of instability, since in

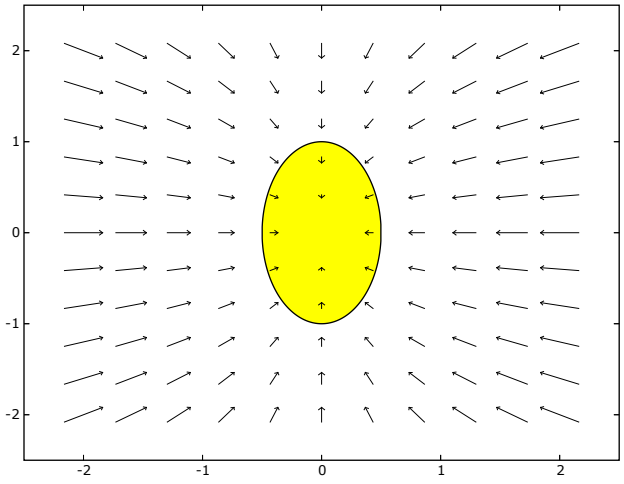


Figure 2-1: Shown is region B_1', where the robust stability criterion may not hold because σ is greater than the gradient descent vector.

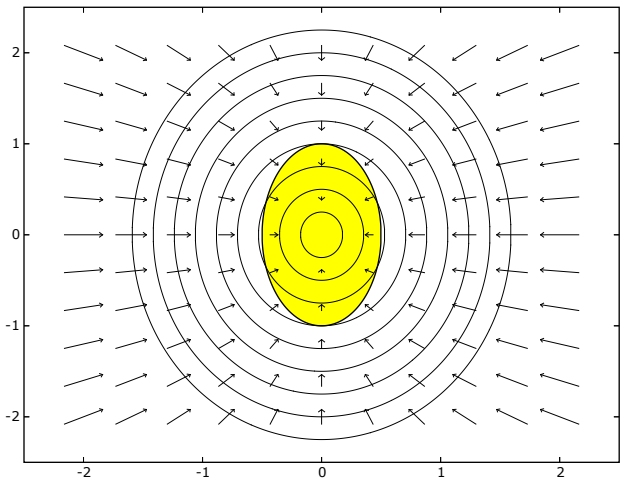


Figure 2-2: Shown are the level curves of the objective function, together with region B_1.

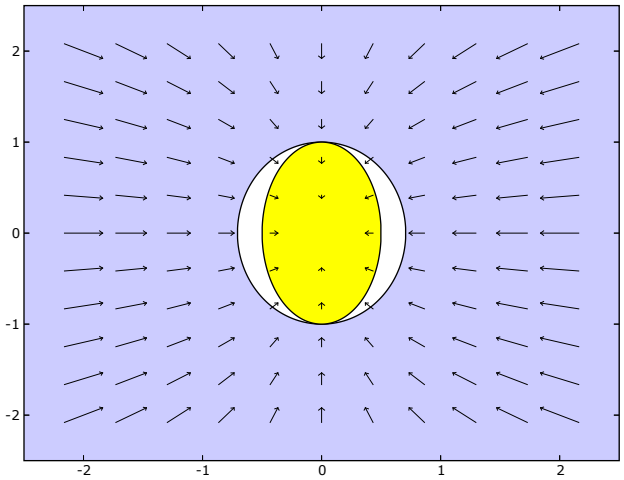


Figure 2-3: The innermost region (yellow) is B_1' — the region where the robust stability criterion holds. The white region around it is B_1 — the smallest level curve of the objective function that contains B_1'. The system is guaranteed to converge to the region B_1 and stay within the region.

that case the region B_1 may be large.

In the general case, if the error current only bounded, but no other properties of the current are known, global stability cannot be guaranteed. Indeed, as formulated, the error current σ injected may be directly sinusoidal, which, given adequately low slope of the original objective function, will translate to sinusoidal oscillations in $\mathbf{x}$. In the case of some specific circuits, however, overall stability may follow from linearity. Specifically, assume that there is some ball B_2 around the minimum where the system can be accurately modeled by the linear approximation. Assume that stability is shown everywhere except a ball B_1 around the minimum, as above. Then, since linear systems are scale invariant, if $B_2 \subset B_1$, this shows stability everywhere.

Note that for this entire argument, the system cannot be underconstrained, or otherwise have a set of more than one minimum. In the case of an underconstrained system, the objective function would only guarantee stability within a subset of the system coordinates; there may be an space in which the system is free to oscillate or saturate.

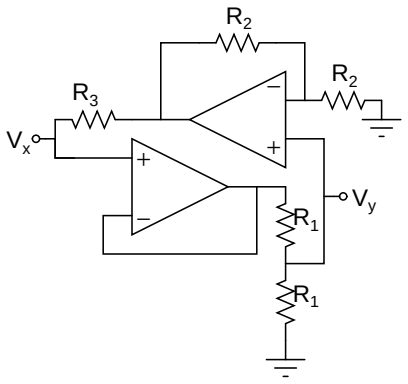


Figure 2-4: Active Transformer. Maintains voltage ratio of $V_x = 2V_y$.

For instance, consider a circuit that is intended to hold the constraint $V_x = 2V_y$, as shown in Figure 2-4. It is obvious from this circuit that if it is left unconnected, if the loop gain is just under 1, it will evolve to the solution $V_x = 2V_y = 0$. If the loop gain is more than 1, it will evolve to $V_x = 2V_y = V_{RAIL}$, where, depending on initial conditions, V_{RAIL} is either the positive or negative rail. The constraint is satisfied, but this system behaves in a potentially unstable way within the space where the constraint is satisfied. Chapter 3 shows a context in which this behavior can be exploited to serve a useful function.

2.2.2 Saturation Behavior

In many constraint circuits, saturation behaves as an additional implicit set of constraints:

$$\forall i : V_{ee} < V_i < V_{cc}$$

If this is not the case, additional constraint blocks can prevent the system from reaching saturation behavior. These blocks typically have the form:

$$\forall i : V_{min} < V_i < V_{max}$$

Where,V_{min} and V_{max} are some values that guarantee the blocks (both internally and externally) never run into saturation.

2.2.3 High Frequency Stability

In general, there will be some frequency ω_0 (which is independent of the node capacitances), up to which the above logic demonstrates stability directly. For high-frequency stability, it is adequate to make the capacitances on the joint nodes adequately large such that the global dynamics dominate. This follows from an argument analogous to the small-gain theorem.

Informally, consider a linear system. Assume that all of the nodes adjacent to node j have an error voltage of up to $1\mathrm{V}_{PP}$. Calculate the effect of those errors on the current I_j. Chose a node capacitor C_j such that the effect of this current on voltage V_j is less than $1\mathrm{V}_{PP}$. If this is done for all nodes, any high-frequency oscillations will die off.

More formally, take A_i to be the signal directly injected onto node i from some noise source. Take B_i to be the signal injected onto the node from the adjacent constraint blocks due to voltage variations on both the node and the adjacent nodes.

Assume that, for all i, at time t, $A_i < \alpha$ and $B_i < \beta$ for some constants α and β. Take a single node j in isolation. Define $g_{ij}(s)$ to be the small signal transconductance from node i to node j. Here,

$$B_j = \frac{1}{Cs} \cdot \sum_{i \in \mathcal{N}(j)} g_{ij}(A_i + B_i) \leq \frac{1}{Cs} \cdot \sum_{i \in \mathcal{N}(j)} g_{ij}(\alpha + \beta)$$

Then let:

$$Cs > \left(\frac{\alpha}{\beta} + 1\right)\left(\sum_{i \in \mathcal{N}(j)} g_{ij}\right)$$

Then $B_j < \beta$, and maintain the constraint $A_j + B_j < \alpha + \beta$. If this is satisfied for all nodes, the circuit will be high-frequency stable.

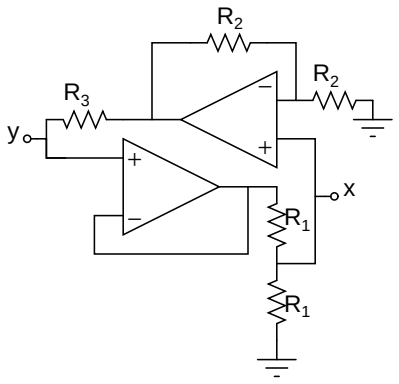


Figure 2-5: The circuit holding the constraint $2x = y$. All resistors are 10k.

2.3 Test Circuit

A test circuit solved the set of equations:

$$2x = y$$

$$x - y = z$$

$$x + y + z = 4$$

The circuit shown in Figure 2-5 implements first equation, $2x = y$. The circuit shown in Figure 2-6 implements the second equation, $x - y = z$ (or, symmetrically, $x - z = y$, $y + z = x$). Finally, the circuit in shown in Figure 2-7 implements the third equation, $x + y + z = 4$. These circuits use the NJM062 operational amplifier, and $\pm$15V rails. All nodes have 0.47μF capacitors on them. A photograph of the circuit is shown in Figure 2-8.

These circuits are more complex than is necessary. It is generally possible to use only one operational amplifier per input. This implementation has more parts than necessary to allow the simple monitoring, instrumentation, and characterization of the circuit and its behavior.

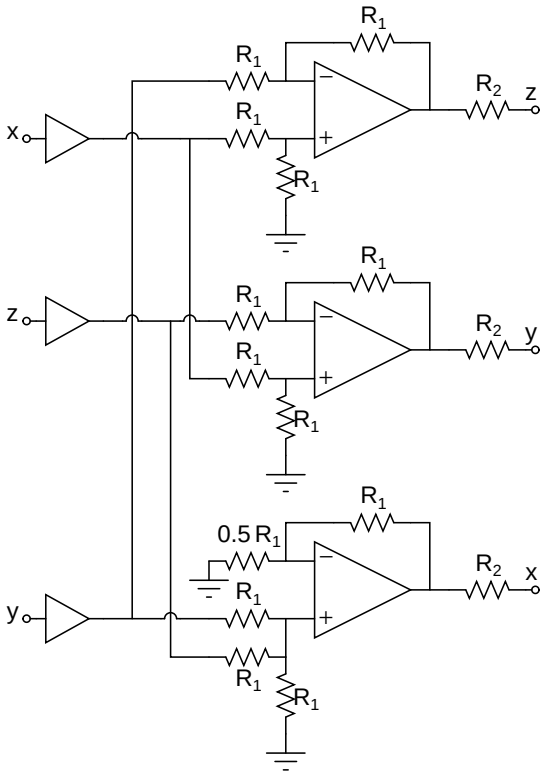


Figure 2-6: The circuit holding the constraint $x - y = z$. All resistors are 10k.

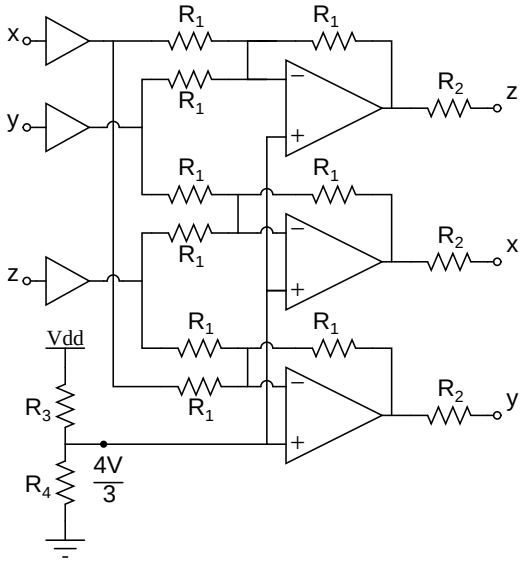


Figure 2-7: The circuit holding the constraint $x + y + z = 4$. All resistors except R3 and R4 are 10k. R3 is 9.1k, while R4 is 1.3k. These were chosen to set the $\frac{4}{3}$V node to the appropriate voltage, a given a power supply supply voltage of just under 11V.

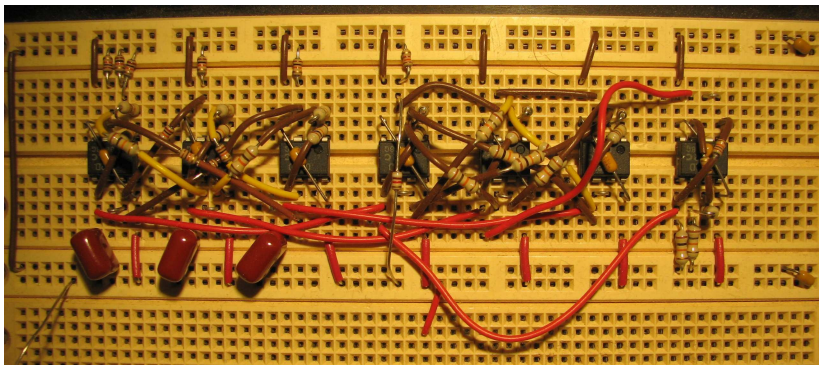


Figure 2-8: A photo of the static constraint solving test circuit.

2.3.1 Circuit performance

Rate of convergence of the circuit was measured by driving the node labeled $\frac{4V}{3}$ with a square wave rather than holding it constant. Scope traces of the results are shown in Figure 2-9, with the mapping of variables to channels and convergence times in Table 2.1.

2.4 Simplifying Circuits

In many cases, circuits designed under this methodology can be further simplified. A complete example of this will be shown in Chapter 4. While many techniques for simplification are specific to each circuit implemented, several common techniques exist. The most common ones are:

1. **Common expression elimination**. In constraint blocks, it is very common to calculate the same expression multiple times.

2. **Common buffers**. In most cases, if several constraint blocks connect to a single node, each will buffer that node voltage internally. All nodes can share a common buffer.

3. **Common current output**. In many cases, individual constraint blocks use voltage logic, followed by a voltage-to-current stage. In these cases, many con-

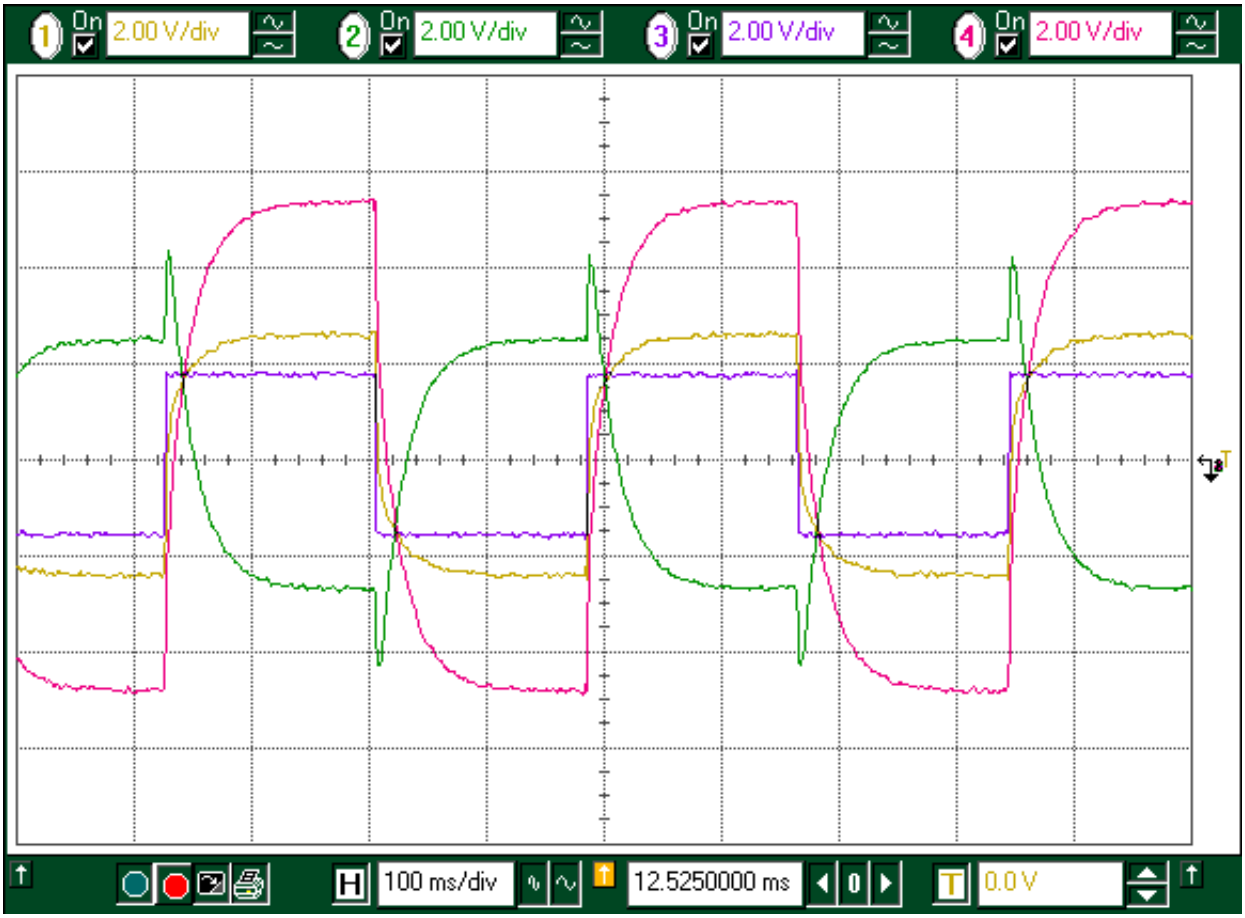


Figure 2-9: The scope traces of the dynamics of the static constraint solving circuit. Channel 3 is the input voltage from a function generator, driving the value of the constraint $x + y + z = 3v_{in}$.

Channel	Convergence Time	Variable
Channel 1	27mS	x
Channel 2	29mS	z
Channel 3	<1mS	v_{in}
Channel 4	42mS	y

Table 2.1: The convergence times of the static constraint solver test circuit. These were measured as 20% to 80% rise time of the waveforms. Surprisingly, the convergence time of channel 4 was substantially slower than channels 1 and 2.

straint blocks may share an averaging circuit and a common voltage-to-current stage. This is especially useful if the node has controlled gain.

4. **Bidirectional⟶unidirectional**. Some ports interface to external unidirectional circuits. These can sometimes be designed as unidirectional, instead of bidirectional ports, reducing circuit complexity.

Chapter 3

Dynamic Case — Function Fitting

This framework can be used for function fitting[1]. To implement function fitting, operate a constraint circuit such that, at any point in time, it is underconstrained[2]. The inputs, however, should change over time. If the constraints generated by all possible inputs are taken together, the resulting system should either be fully constrained, or overconstrained. In these cases, given some criteria on the system, the circuit will find an optimal or near-optimal solution to fit those combined constraints.

The circuit finds this solution through a form of iterated gradient descent. While this does not converge for all models, it does converge for a wide range of models, including general linear regression. As presented, it is also restricted to memoryless models, although there are limited ways to extend it to models with memory.

Consider a basic example: take an unknown plant and a system modeling this plant with the first-order Taylor approximation $y = \hat{b}x + \hat{c}$ (a basic affine line fit). To create a system to perform this fit, set up an underconstrained constraint block for this equation, and connect it to the system under observation, as shown in Figure 1-3.

Assume that the original system fits $y = bx + c$ exactly. Let $\hat{b}_0$ and $\hat{c}_0$ be the initial estimates of b and c. Each training input viewed will have its constraints satisfied by a line in the b, c plane. The constraint circuit will project the initial estimate $\hat{b}_0, \hat{c}_0$

[1]Function fitting is also often referred to as machine learning in artificial intelligence terminology.

[2]The system does not have to be underconstrained — as will be shown in Section 3.2, the system will optimize properly with overconstrained systems as well. Nevertheless, the underconstrained case is helpful for explaining system behavior, especially at lower switching speeds.

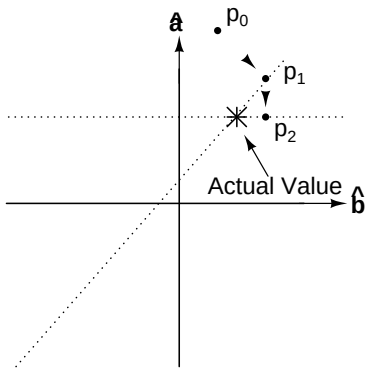


Figure 3-1: Each sample point from the plant will correspond to a hyperplane of possible model parameters. The circuit will project the current model onto that hyperplane. Two steps of this procedure are shown in this graph.

onto that line (or, if not given time to run to completion, it will move it onto some part of the line segment containing the old point and its projection onto that line), resulting in an improved estimate $\hat{b}_1, \hat{c}_1$. As this process repeats, each step moves $\hat{b}, \hat{c}$ closer to the actual values of b, c. Two steps of this process are shown in Figure 3-1.

Since each step is just a projection onto a line passing through the desired answer, with each step of this sequence $\hat{b}, \hat{c}$ move closer to the desired values of b and c (or, in the degenerate case, if they are already on the line, they stay fixed). The same argument, with hyperplanes instead of lines, may be applied to arbitrary general linear regression models:

$$\sum c_i f_i(\mathbf{x}) = g(\mathbf{x})$$

Here, $\mathbf{x}$ is the vector observed, each f_i is an arbitrary function, and c_i are the associated model parameters. This form includes Taylor expansions, Fourier series, Chebyshev polynomials, discretizations, as well as wavelet expansions. As will be shown later, a broad range of functions not of this form will also converge.

This basic analysis assumes that the model can fit the original system exactly. Where this is not the case, individual steps may move away from the ideal model. Section 3.1 shows a form of convergence in the case where the model cannot fit the system exactly. This analysis is still restricted to general linear regression. It shows stability in two cases. First, it shows stability in the continuous time case, where the

input is changing, and the circuit does not have time to reach equilibrium for any given input. Second, it shows stability for the discrete time case, where this circuit is shown successive, discrete inputs, and reaches equilibrium for each one.

As will be shown, the type of stability achieved is rather unusual. First of all, while the circuit is provably stable with a changing input, in the case of a fixed input, it may be unstable (although the oscillations would be at a comparably low frequency, so will not significantly affect the power rails, and so hopefully not affect the function of the circuit). With a fixed input, there is a space of sets of model parameters $(c_1, ..., c_n)$ that will meet the constraints. The system may drift within that range, and may do so in an unstable fashion, while continuing to satisfy the constraint generated by the current input.

Second of all, even without drift, it is possible to construct examples of system functions and inputs with very bad behaviors. If the model cannot perfectly fit to the system, with worst-case inputs, the system may diverge arbitrarily far from the best approximation. In practice, however, the range of inputs that may cause the system move in the wrong direction decreases as the circuit moves away from the ideal model. Making it diverge arbitrarily far from the optimal answer requires arbitrarily large degrees of control over the input the system. Eventually, marginal amounts of noise on the input of the system would cause the model to move towards, rather than away from, the ideal model.

In practice, this manifests itself as the system building local approximations. If the system is attempting to model $x = \sin(y)$ with the Taylor approximation $x = ay + b$, the best model parameters over the entire set of possible values are simply $(0, 0)$, giving the model $x = 0$. If the system only sees points near the origin, however, it will select the model parameters $(1, 0)$, constructing the model $x = y$, giving an error of 1 in the model parameters. If the system is forced to see points further out from the origin, this error will fall.

The system will also converge for a wide set of models beyond general linear regression. In the same way as convexity can be used as a criteria to show stability in the static case, it may also be used to show stability in the continuous time case. The

convexity-based stability criterion is shown in Section 3.2. In practice, this criterion is somewhat difficult to use. It is also only proven in the case where the input changes on a timescale much larger than the system dynamics.

3.1 Stability for General Linear Regression

This section shows stability in two cases. Section 3.1.2 shows stability in the continuous time case. Here, the input is assumed to be continuously moving, and the model moves gradually towards the hyperplane defined by the current input. This is an appropriate model for most analog systems. Section 3.1.3 covers the case in which the input moves in discrete increments, and at each step, the model is projected onto the hyperplane defined by the current input. This is an appropriate model for some analog systems interfacing to digital electronics — for instance, the analog portion of a digital radio, transmitting discrete symbols.

3.1.1 Definitions and Notation

Take a general linear regression model $C : c_1 f_1(\mathbf{x}) + c_2 f_2(\mathbf{x}) + ... + c_n f_n(\mathbf{x}) = g(\mathbf{x})$, and a system $S = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ...\}$. Define a ball of radius r around $\mathbf{c}$ in C as the set of points

$$B_C(\mathbf{c}, r) = \left\{ \mathbf{x} \middle| \exists \epsilon_1, \epsilon_2, ..., \epsilon_n \text{ s.t. } \sum_i (c_i + \epsilon_i) f_i(\mathbf{x}) = g(\mathbf{x}), \sum_i \epsilon_i^2 < r^2 \right\}$$

Define the notation that a system S can be approximated by C to within r iff:

$$\exists \mathbf{c} \text{ s.t. } S \subset B_C(\mathbf{c}, r)$$

Graphically, any function that fits in the gray region of Figure 3-2 is in the ball of radius 1 around the function $y = ax + b$, centered at $(a, b) = (1, 2)$. An example of a valid function is shown in Figure 3-3.

Assume system S is within a ball in C around $\mathbf{c}$ of radius r. Take any point in

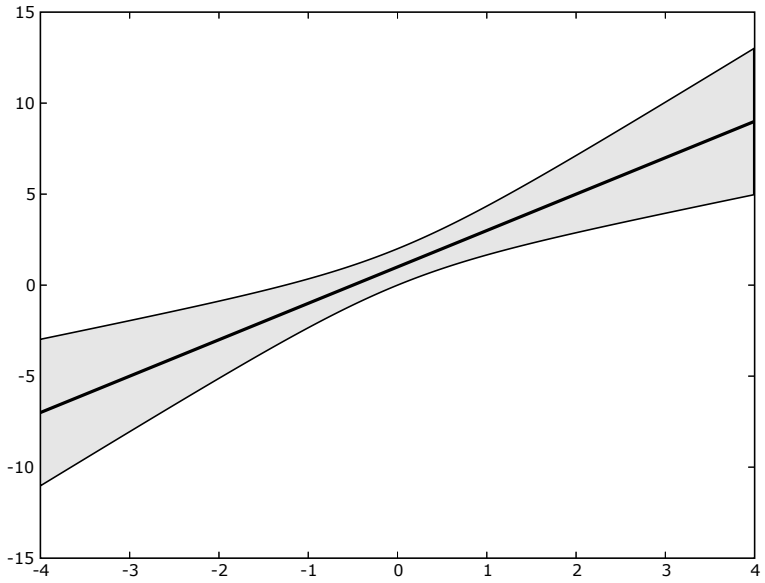


Figure 3-2: This is an example of a ball around a function. It is a ball around $y = ax + b$, centered at $(a, b) = (2, 1)$, of radius 1.

the system S. This point generates a hyperplane of possible values for $\hat{\mathbf{c}}$ that passes through the ball $B_C(\mathbf{c}, r)$. Take the worst-case: assume the point is on the perimeter of the ball[3], as shown in Figure 3-4.

3.1.2 Robust Continuous Time Stability

This section analyzes the continuous time case, where $\mathbf{x}$ is continuously changing. Here:

$$\frac{dl}{dt} = \pm r \sin\theta - l \sin^2\theta$$

Then, for convergence, the expected value of the change in l needs to be negative:

$$E\left(\frac{dl}{dt}\right) < 0$$

[3]The analysis generalizes trivially to a tighter bound if this is not the case.

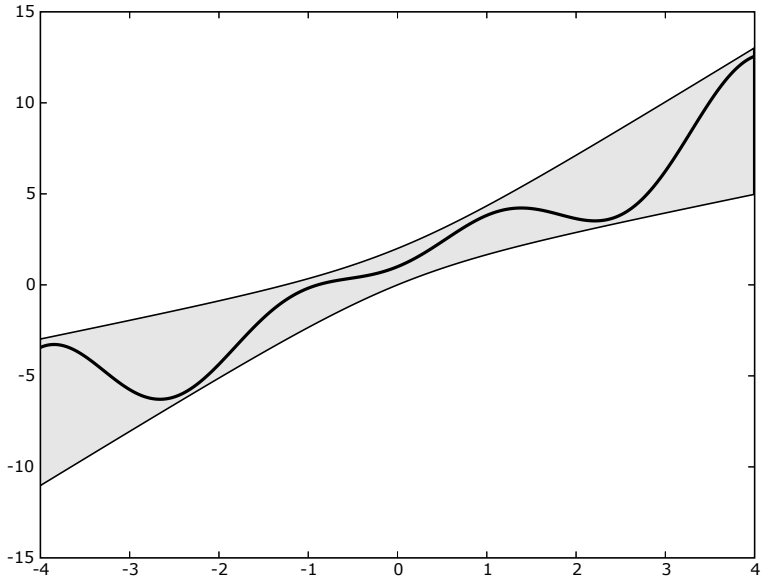


Figure 3-3: This is an example of a function contained in the ball around another function. In this case, the function $(2+0.9\sin(2x))x+1$ function is shown in the ball around $y = 2x + 1$ described in Figure 3-2.

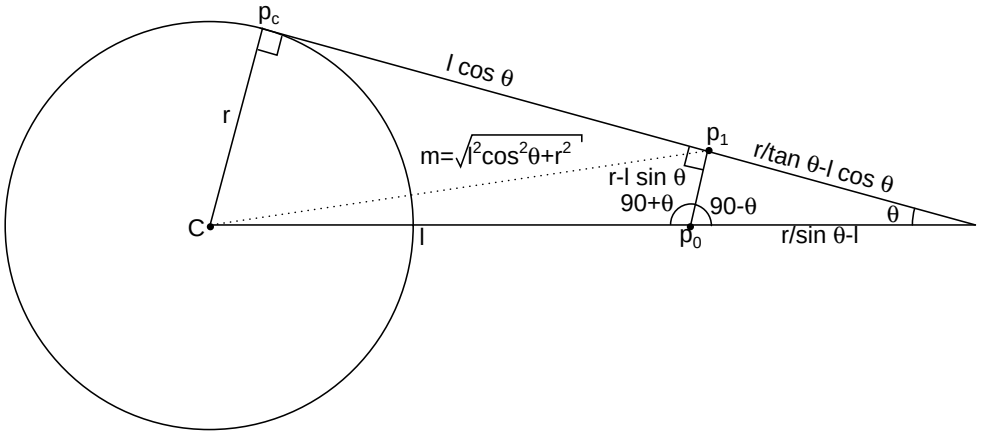


Figure 3-4: This is an illustration of a projection of an estimate onto the hyperplane generated by a point on the surface of a ball around a function.

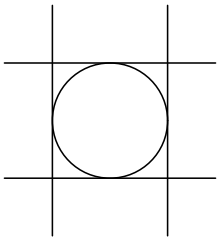


Figure 3-5: This illustrates projecting onto 4 lines around a ball of radius r. Notice how in the discrete-time scenario, or worst-case continuous time scenario, the system would end up on a point $r\sqrt{2}$ from the center. In contrast, in the typical continuous time scenario, where all four lines are seen equally frequently, the system would wobble around the center. In the limit case of high speed/large capacitors, it would give an optimal estimate of the model parameters.

Taking the worst-case,

$$E\left(r\left|\sin\theta\right| - l\sin^2\theta\right) < 0$$

Which gives the event horizon:

$$\sin\theta = \frac{r}{l}$$

Note that this is a worst-case scenario analysis. In many practical scenarios, r will also vary. In addition, taking the absolute value significantly increases error, since in many cases, this assumes the system is moving in the wrong direction, whereas it is moving in the right one. Figure 3-5 shows how this can lead to the estimate being significantly over-conservative. Here, if the system were switching quickly between the four hyperplanes, it would generate a perfect estimate. In contrast, by the criterion given, the estimate would be bounded to have an error of at most $r\sqrt{2}$.

3.1.3 Robust Discrete Time Stability

In the discrete time case, the circuit is presented with discrete inputs, and at each time step, the estimate is projected onto the hyperplane defined by those inputs (as opposed to projecting part way, as in the continuous time case). This section will show that the error bound in the discrete time case is the same as that shown in the continuous time.

Define l to be the distance to the initial approximation p_0. Let m be the distance to the new approximation p_1, after the current round of gradient descent. Then,

$$m = \sqrt{l^2 \cos^2 \theta + r^2}$$

$$m^2 = l^2(1 - \sin^2 \theta) + r^2$$

$$m^2 - l^2 = l^2 \sin^2 \theta + r^2$$

Taking the expected value,

$$E(m^2 - l^2) = E(r^2 - l^2 \sin^2 \theta)$$

This gives the maximum expected magnitude of the drift. For convergence, the new point must be closer than the old point, so $E(m^2 - l^2) < 0$. This implies:

$$E(r^2 - l^2 \sin^2 \theta) < 0$$

This gives same event horizon as in the continuous time case:

$$|\sin \theta| = \frac{r}{l}$$

If $|\sin \theta| > \frac{r}{l}$, then the system moves towards the center. If $|\sin \theta| < \frac{r}{l}$, the system moves away from the center.

Put differently, let us assume that Mallory, a hostile opponent, had control over the input to a system that Alice was trying to model. Mallory wants to force Alice's model to diverge as far as possible from the real system. If Mallory can control θ to within ϵ, he can still only make the system have a maximum error of $\frac{r}{\sin \epsilon}$. If Alice designed the overall system, she can foil Mallory by making the input slightly noisy, weakening Mallory's control of θ.

At the same time, even with perfect control, the rate of divergence falls off rapidly with how far it is from the optimum. The maximum divergence per step would fall

off as $\sqrt{r^2 + l^2} - l$, and after k steps, the worst-case distance would be $\sqrt{kr^2 + l_0^2}$

Note that if there is noise in the system, the system needs $E(|\sin\theta|) \gg \frac{r^2}{l^2}$ to counteract for any drift in the values of $c_1 ... c_n$ stored on the capacitors.

In practice, although the bounds are the same, the continuous time case results in better convergence. Specifically, in steps moving away from convergence move much more quickly away from the center towards the end of the step. In contrast, convergent steps move much more quickly towards the center in the beginning of the step. An intuitively explanation of this is shown in Figure 3-5.

A good algorithm would perform the fast gradient descent far from the optimum, and slower gradient descent near the optimum. Further analysis of the optimal rates of descent is a potential area of future research.

3.2 Parallel Equivalent Model

In the limit case of slow integration rate, it is possible to transform the dynamic model into an equivalent static model. Practically, near the limit case, it is possible to do the same transformation, but treating error due to finite switching speed as noise. This transformation is a powerful tool for showing optimality of the solution, for determining convergence and stability, and for analyzing rate of convergence and system dynamics.

As an example, take the modeling system shown in Figure 3-6. This system is shown three training inputs, $(x_0, y_0), (x_1, y_1), (x_2, y_2)$, with duty cycles of $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$, respectively. This is trivially equivalent to three systems, each connected to the capacitors with the respective duty cycle, as shown in Figure 3-7. In the limit case of fast switching speed, this is equivalent to having all three blocks connected continuously, but with weaker objective functions, as shown in Figure 3-8. This is identical to the static constraint case.

As shown in Chapter 2, this finds the parameters that minimize the least-squares

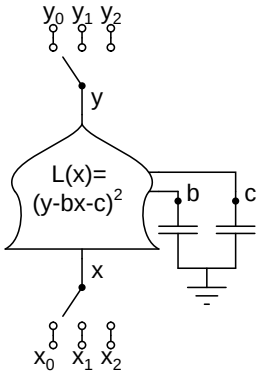


Figure 3-6: Shown is a basic modeling system, with the training parameters switching between three sets of inputs, (x_0, y_0), (x_1, y_1), and (x_2, y_2).

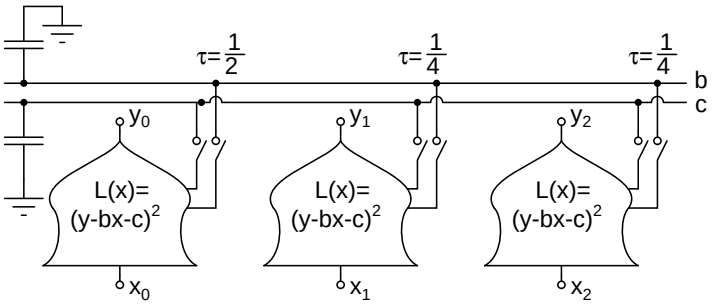


Figure 3-7: If the original modeling system is shown three inputs, as shown in Figure 3-6 it is equivalent to three identical modeling systems, each permanently connected to an input, with switches connecting the constraint blocks to the appropriate capacitors.

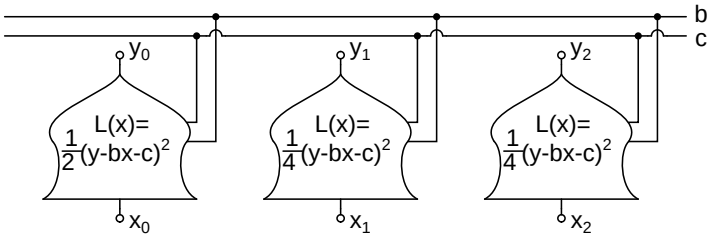


Figure 3-8: In the limit case of fast switching, remove the switches from the system shown in Figure 3-7, and instead scale back the currents from the constraint blocks based on the duty cycle with which the blocks are connected to the different inputs.

objective function for the general linear regression model:

$$\mathbf{c}_{out} = \arg\min_{\mathbf{C}} \mathrm{E}\left(\sum c_i f_i(\mathbf{x}) - g(\mathbf{x})\right)^2$$

Thus far, stability was only shown in the case of general linear regression. This analysis gives a way to show stability for a broader range of models. Once the dynamic system is transformed into the parallel static system, it is possible to do a static convexity analysis on the overall objective function. If the stability conditions from Chapter 2 are met, convergence in the limit case of slow integration follows.

In the non-limit case, the analysis follows that of the robust low frequency stability, presented in Section 2.2.1. Following the example, take three steps:

$$\langle b_{t+1}, c_{t+1}\rangle = \langle b_t, c_t\rangle + \mathbf{s}(x_0, y_0, b_t, c_t, \tau_0)$$

$$\langle b_{t+2}, c_{t+2}\rangle = \langle b_{t+1}, c_{t+1}\rangle + \mathbf{s}(x_1, y_1, b_{t+1}, c_{t+1}, \tau_1)$$

$$\langle b_{t+3}, c_{t+3}\rangle = \langle b_{t+2}, c_{t+2}\rangle + \mathbf{s}(x_2, y_2, b_{t+2}, c_{t+2}, \tau_2)$$

Where $\mathbf{s}(x, y, b, c, \tau)$ is the step that results from showing the input $\langle x, y\rangle$ to the system with the current estimate $\langle b, c\rangle$ for time τ. Let $\langle b', c'\rangle$ be the evolution over the same time period in the ideal limit case. Define the error $e = L(\langle b_{t+3}, c_{t+3}\rangle) - L(\langle b', c'\rangle)$, and let the total time be τ. If the error accumulating over time is less than the ideal rate of decrease of the objective function L, the system will be converging:

$$\frac{e}{\tau} < \frac{dL}{dt}$$

Following the logic of Section 2.2.1, this can be used to define a region B_1 to which the system will converge.

This type of analysis also gives a technique for estimating rate of convergence, since the static/parallel model can give a good estimate of overall system dynamics. As with stability, this assumes that the the capacitors are large enough that the error from the static model is small. In cases where the capacitors are smaller, and the

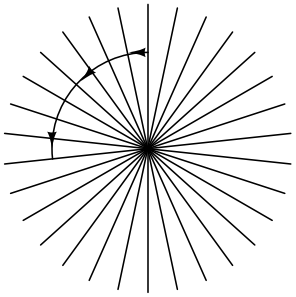


Figure 3-9: Shown is a set of lines around a point. Here, each line corresponds to an input to the modeling system. The point in the middle corresponds to the exact model of the system being modeled. If the system were successively projecting onto lines of slowly changing angle, as shown by the arc, the system would take arbitrarily long to reach the center. If, on the other hand, it were to project onto two perpendicular lines in succession, it would reach the center in two steps, and stay there.

above model does not hold, the rate of convergence is very dependent on the order in which inputs are seen. Figure 3-9 demonstrates this. Depending on order of inputs, the system may either converge in 2 steps (if they are orthogonal), or never (if they change in small increments)[4]. As a result, it is much more difficult to characterize system dynamics.

[4]This may or may not be a problem. Indeed, if the recent past of the input is a good indicator for the near future, this may be a great benefit. In the extreme case of an input slowly varying over the full range, but quickly moving around a local point, the system will continually maintain a good approximation of the nonlinearity around the current point. As a result, the system will be able to compensate for it very well with a very simple approximation.

Chapter 4

Linearization

This chapter discusses how to use the framework for linearization. Basic linearization is a very straightforward extension of the modeling framework. First, the modeling block from Chapter 3 derives the model parameters for the nonlinearity. Then, a matched block predistorts the input signal based on those parameters. This is shown in Figure 4-1. This basic circuit is somewhat inefficient — it consists of two identical blocks, with many redundant parts. This section will show how to compile the original circuit into a more efficient implementation with a much lower parts count, show some interesting properties of this circuit, and show experimental results for how well this circuit works.

The quality of the predistortion depends on the level to which the blocks match[1] and the level to which the function they represent can approximate the nonlinearity. Note that even if the model functions are very poorly poorly implemented, so long as the functions match, and can still reasonably approximate the plant, the system will still work well — it will just use a slightly different model than was intended. In many cases, matching can be guaranteed by integrating the modeling block and the

[1]For some types of blocks, matching is more difficult to achieve than in traditional circuit design. The circuit shown in Figure 4-1 relies on matching between the modeling block with information flowing from $(x, y) \longrightarrow (a, b, c)$ and the predistortion block with information flowing $(a, b, c, y) \longrightarrow x$. The active parts of the topologies for the two information flows may be significantly different. If the topology is implemented in a symmetric way (e.g., a diode to represent an inequality), this is not a problem. If it is implemented with independent paths for both error currents (e.g., the active transform shown in Figure 2-4), two paths with different topologies must match).

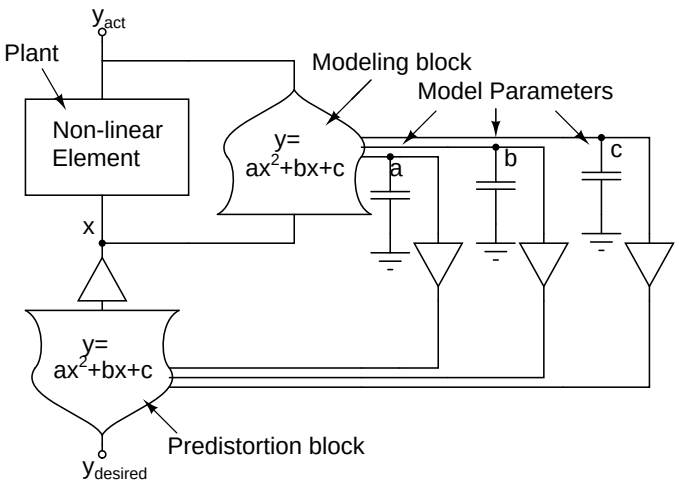


Figure 4-1: Compensating for a nonlinearity.

predistortion block.

Due to the limited frequency response of the optimization blocks, this is not immediately practical for high-speed applications. The performance limitations are not fundamental, as there are plausible techniques for making this class of circuits very fast, but this remains an area of future work. Note that in contrast to the feed-forward section, in most cases, the modeling section does not need to be particularly fast. If the system is inside of a feedback loop, the feed-forward predistortion needs to be fast enough so as not to significantly effect frequency response as required for stability. In contrast, the modeling section only needs to operate at signal speeds. Similarly, in the case of a pure feed-forward system (rather than one inside of a feedback loop), limited frequency response in the feed-forward section will translate to limited frequency response of the overall system. In contrast, limited frequency response in the modeling section will, at worse, translate to a small amount of modeling error, and in many cases, the error will tend to average out.

Note that since models may be non-monotonic, the predistortion block may have local minima. In this case, the predistortion block is not guaranteed to converge to the proper minimum. There are a number of solutions to this — most simply, using a monotonic model guarantees that this problem will not occur. If the model's monotonicity depends on model parameters, a constraint block can be used to force

the model parameters to remain in the space where the model is monotonic. Finally, starting the system near the global minimum can guarantee convergence in a broad, but not complete, range of cases. This can often be implemented with a conventional feed-forward path around the predistortion block that holds the output of the predistortion block near the proper solution, but still permits the predistortion block freedom to move within the known error region.

4.1 Circuit Compilation Overview

The circuit presented, while fairly powerful, is more expensive in hardware than necessary. This section will show that in some cases this cost can be dramatically reduced, while improving performance. Take a model of the form:

$$x = \sum_i c_i f_i(y)$$

Assume that the system modeled is monotonic. The circuit is trying to find the model parameters $\langle \hat{c}_1, ..., \hat{c}_n \rangle$. Take the objective function:

$$\min \left(x - \sum_i c_i f_i \left(y_{act} \right) \right)^2$$

So,

$$\frac{d}{dc_j} = 2 f_j \left(y_{act} \right) \left(x - \sum_i c_i f_i (y_{act}) \right)$$

From here on, drop the 2, since it is an arbitrary scaling factor that sets convergence speed. The first simplification is to treat ports known to be inputs or outputs as non-bidirectional ports. The feed-forward section (the "inverse model") can calculate $x = \sum_i c_i f_i \left(y_{des} \right)$ directly, using traditional input/output blocks. Since x and y are inputs, the function fitting section, needs to treat only the c_i ports as bidirectional.

The next set of simplification relies on approximating system dynamics. The system will lose some of the guarantees of optimality and performance in the process.

The system, by the methodology, would follow the dynamics:

$$\begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \vdots \\ \dot{c}_n \end{bmatrix} = \begin{bmatrix} f_1\left(y_{act}\right)\left(x - \sum_i c_i f_i(y_{act})\right) \\ f_2\left(y_{act}\right)\left(x - \sum_i c_i f_i(y_{act})\right) \\ \vdots \\ f_n\left(y_{act}\right)\left(x - \sum_i c_i f_i(y_{act})\right) \end{bmatrix} = \begin{bmatrix} f_1\left(y_{act}\right) \\ f_2\left(y_{act}\right) \\ \vdots \\ f_n\left(y_{act}\right) \end{bmatrix} \cdot \left(x - \sum_i c_i f_i(y_{act})\right)$$

This consists of two terms: the direction term $\langle f_i\left(y_{act}\right), \cdots, f_n\left(y_{act}\right)\rangle$, and the scaling term $\left(x - \sum_i c_i f_i(y_{act})\right)$. The direction term is much simpler than the scaling term. The most obvious simplification is to calculate the scaling term once, rather than for every term, and reuse the result. This still requires the circuit to make the complex calculation once[2]. This calculation is, however, unnecessary.

Notice that the proofs in Sections 3.1.2 and 3.1.3 rely solely only on direction of descent, and not rate of descent. As a result, one can approximate the scaling term with any function that does not change sign. In other words, for any function g that is restricted to the first and third quadrant, the system can descend along:

$$\begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \vdots \\ \dot{c}_n \end{bmatrix} = \begin{bmatrix} f_1\left(y_{act}\right) \\ f_2\left(y_{act}\right) \\ \vdots \\ f_n\left(y_{act}\right) \end{bmatrix} \cdot g\left(x - \sum_i c_i f_i(y_{act})\right)$$

The circuit can now be simplified by looking for appropriate functions that have the same sign as $\left(x - \sum_i c_i f_i(y_{act})\right)$, but are easier to calculate.

Assume the model is monotonic. If the system is monotonic (or nearly so), this requirement is reasonable — even if the space of possible models allows for non-monotonic models, the model parameters may be restricted to the space that generates monotonic models with an appropriate constraint block. In this case:

$$\text{sign}\left(x - \sum_i c_i f_i\left(y_{act}\right)\right) = \text{sign}\left(\sum_i c_i f_i\left(y_{des}\right) - \sum_i c_i f_i\left(y_{act}\right)\right) = \text{sign}(y_{des} - y_{act})$$

[2]This calculation, while complex, is not as bad as it first appears, since the $f_i\left(y_{act}\right)$ terms are already calculated for the direction term.

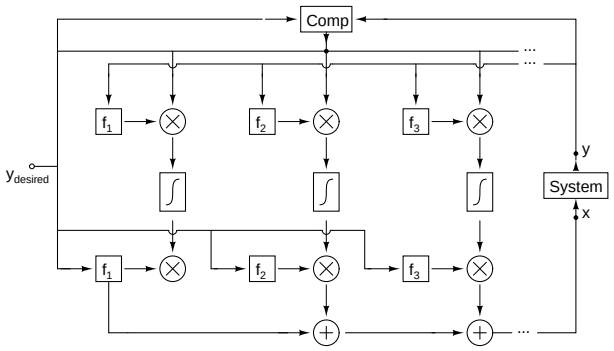


Figure 4-2: Block diagram of function-fitting and linearization circuit.

And so, the system can descend along the trajectories:

$$\frac{d}{dc_i} = f_i\left(y_{act}\right) g\left(y_{des} - y_{act}\right)$$

Here, g is, again, any function that remains in the first and third quadrant.

With these simplifications, the circuit is much simpler than the original constraint block. A block diagram of the resulting circuit is shown in Figure 4-2.

4.2 Optimality

While changing the rate of descent has no effect on the stability proofs of Sections 3.1.2 and 3.1.3, it does affect the parallel model of Section 3.2, and therefore, optimality. Since the rate of descent may be scaled differently at some points than at others, some points will have more significant weight than others. To give an intuitive understanding of how optimality is affected, take a simple modeling system that is trying to build the model $x = c$. Assume that 50% of the time, this system sees the point $x = 0$, and the remaining 50% of the time, it sees the point $x = 1$. Given equal rates of descent at both points, the system will converge to the model parameter $c = 0.5$. If, on the other hand, rate of descent is 100 times greater for the point $x = 0$ than for $x = 1$, assuming least squares, it will converge to the model $c \approx 0.1$.

This section will explain and quantify this problem. It will show that if the model is approximately affine, this approximation has a minimal effect, and if the model is

fairly non-affine, the level of this effect will be bounded by the ratio of the minimum slope of the model to the maximum slope of the model.

Assume that the model is strictly monotonic. For convenience, define the functions h_{mod} and h_{sys}:

$$x = h_{mod}(y_{des}) = \sum_i c_i f_i(y_{des})$$

$$y_{act} = h_{sys}(x)$$

Here, h_{mod} is the function of the predistortion block, while h_{sys} is the function of the nonlinearity itself.

Take the system instantaneously observing a single data-point. Based on the values $(h_{mod}(y_{des}), y_{des})$ and $(h_{mod}(y_{act}), y_{act})$ approximate h_{mod} as an affine function:

$$\hat{h}_{mod}(y) \approx ay + b$$

In this proof, this approximation will only be used at the points from which it was derived, and will therefore give exact answers. Rephrase the scaling functions using this notation. The simplified scaling function becomes:

$$y_{des} - y_{act} = y_{des} - h_{sys}(x) = y_{des} - h_{sys}(h_{mod}(y_{des}))$$

While the original scaling function is:

$$x - \sum_i c_i f_i(y_{act}) = x - h_{mod}(y_{act}) = h_{mod}(y_{des}) - h_{mod}(h_{sys}(h_{mod}(y_{des}))) =$$

$$\hat{h}_{mod}(y_{des}) - \hat{h}_{mod}(h_{sys}(h_{mod}(y_{des}))) =$$

$$a(y_{des}) + b - a(h_{sys}(h_{mod}(y_{des}))) - b = a(y_{des} - (h_{sys}h_{mod}(y_{des})))$$

As can be seen, the two are identical, except for the scaling factor a, which is approximately the slope of the line connecting $(h_{mod}(y_{des}), y_{des})$ and $(h_{mod}(y_{act}), y_{act})$. By the mean value theorem, a is bounded between the minimum and the maximum of the slope of the model. Since, in most cases, $y_{des} \approx y_{act}$, a is very close the the actual

slope of the model at either point, so this is a fairly tight bound.

This means that if the model has slope of one (the system has a potentially significant offset, but only a small amount of gain error or nonlinearity), this simplification has no effect on system dynamics. If the model is approximately affine (the system is compensating for potentially significant offset and gain error, but a small nonlinearity), this simplification will affect rate of convergence, but not the equilibrium point. If the model is highly nonlinear, the system will still converge, but the weight it gives to the points it sees will be scaled by the slope of the model at those points.

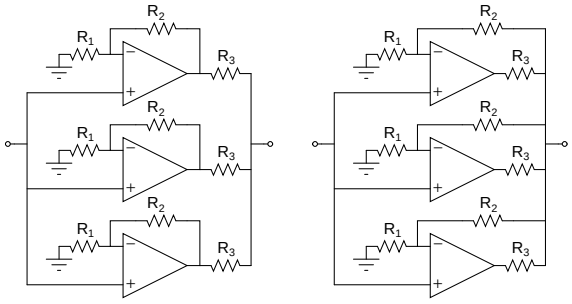
4.3 Robustness

As presented, the circuit is surprisingly robust to component failures. Specifically, if any of the columns fails in a non-time-varying manner (in other words, the output to the averaging circuit is only a function of the current value of the desired output of the system), the other columns will model how its failure affects the output, and compensate for it. This is somewhat limited in utility, as it will not compensate for:

- A noisy column.
- An oscillating column.
- Degenerate cases in which the column adapts in response to other columns attempting to compensate for it.
- Failures outside of the columns (e.g., in the summing circuit or the comparator circuit). This can be alleviated, but not eliminated, by using a separate comparator for each block.

Nevertheless, it will handle a significant range of failures. This technique is not unique to this system. It is also possible to use a similar technique of robustness-by-taking-the-median in traditional feedback design, as shown in Figure 4-3.

As processes scale into the deep sub-micron range, quantization of dopants becomes increasingly important. Even at $0.1\mu m$, a channel may have 150 ± 20 atoms



(a) Mean (b) Median

Figure 4-3: In the traditional circuit (a), robustness is achieved by taking the mean. Each operational amplifier attempts to hold the output voltage at a fixed level. If one of them fails, for instance, taking the output to a rail, the output will move $\frac{1}{3}$ of the way to the rail. In contrast, the voting circuit (b) will take the value of the median operational amplifier. Here, if the bottom operational amplifier were to fail, the other two could overwhelm it, and the primary effect of the failure would be a potential reduction in headroom.

of dopant. As processes scale to smaller devices, it is likely that some devices may entirely fail. In the classical case of large-scale devices with no device failures, and small, Gaussian variations, averaging provides the best estimate. In the discrete case, in contrast, the median will avoid the failure case, and indeed, given an adequately large sample size, will with high probability give the exact desired device. Therefore, it is likely that this type of robustness will be increasingly important in the future.

4.4 Circuit Implementation

The linearization was tested with a simple test circuit which modeled a nonlinearity with a basic Taylor approximation:

$$x = c_3 y^2 + c_2 y + c_1$$

The system tries to find the model parameters $\langle \hat{c_1}, \hat{c_2}, \hat{c_3} \rangle$ by minimizing the objective function:

$$\min(x - c_3 y_{act}^2 - c_2 y_{act} - c_1)^2$$

As described in Section 4.1, in the feed-forward section, the circuit calculates $x = ay_{des}^2 + c_2 y_{des} + c_1$ directly, using traditional input/output blocks.

In the function fitting section, without the simplifications, the circuit would calculate the currents:

$$\frac{d}{dc_3} = y_{act}^2(x - ay_{act}^2 - c_2 y_{act} - c_1)$$

$$\frac{d}{dc_2} = y_{act}(x - ay_{act}^2 - c_2 y_{act} - c_1)$$

$$\frac{d}{dc_1} = (x - ay_{act}^2 - c_2 y_{act} - c_1)$$

In the simplified version, the circuit simply calculates:

$$\frac{d}{dc_3} = (y_{act} - y_{des}) \cdot y_{act}^2$$

$$\frac{d}{dc_2} = (y_{act} - y_{des}) \cdot y_{act}$$

$$\frac{d}{dc_1} = (y_{act} - y_{des})$$

This reduces the requirements down to 4 multipliers, 2 squarers, 2 adders, and 3 integrators, so the system has only 11 substantial parts (operational amplifiers and multipliers). A block diagram of the circuit is shown in Figure 4-4. The detailed implementation is shown in Figure 4-5. This implementation uses five additional operational amplifiers over the advertised 11 parts. These are included for testing purposes. A real-world implementation would implement the differential amplifier and the integrators using one operational amplifier each, rather than two. In addition, the input buffer, and possibly the output buffer, could be omitted. A photo of the circuit is shown in Figure 4-6.

This circuit includes a diode connected to the linear term. The approximations made assumed that the model was monotonic. This diode resistor network guarantees that the linear term stays positive. Although it is possible for the square term to make the model non-monotonic, this does not occur in practice.

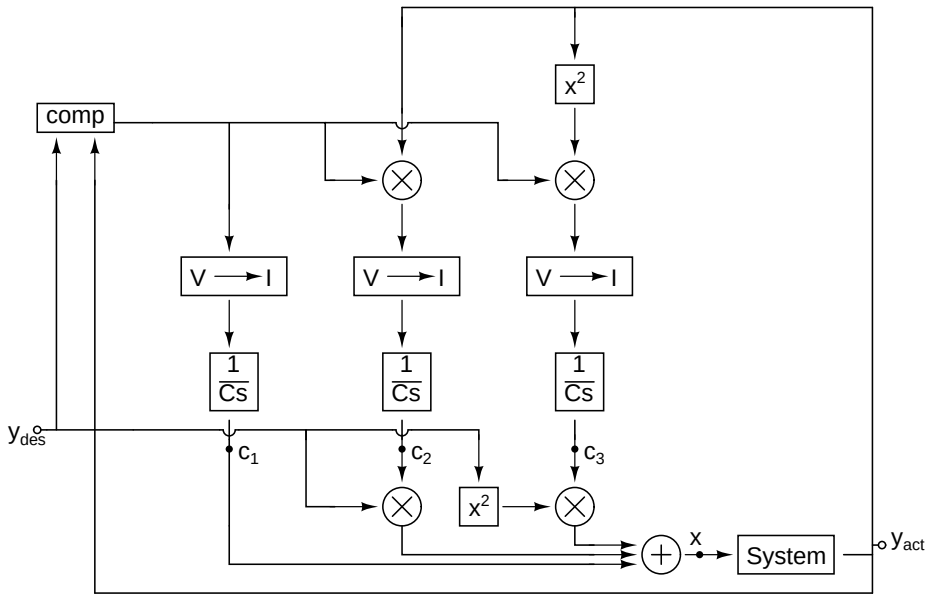


Figure 4-4: Shown is a block diagram of the test circuit for the simplified modeling and linearization circuit.

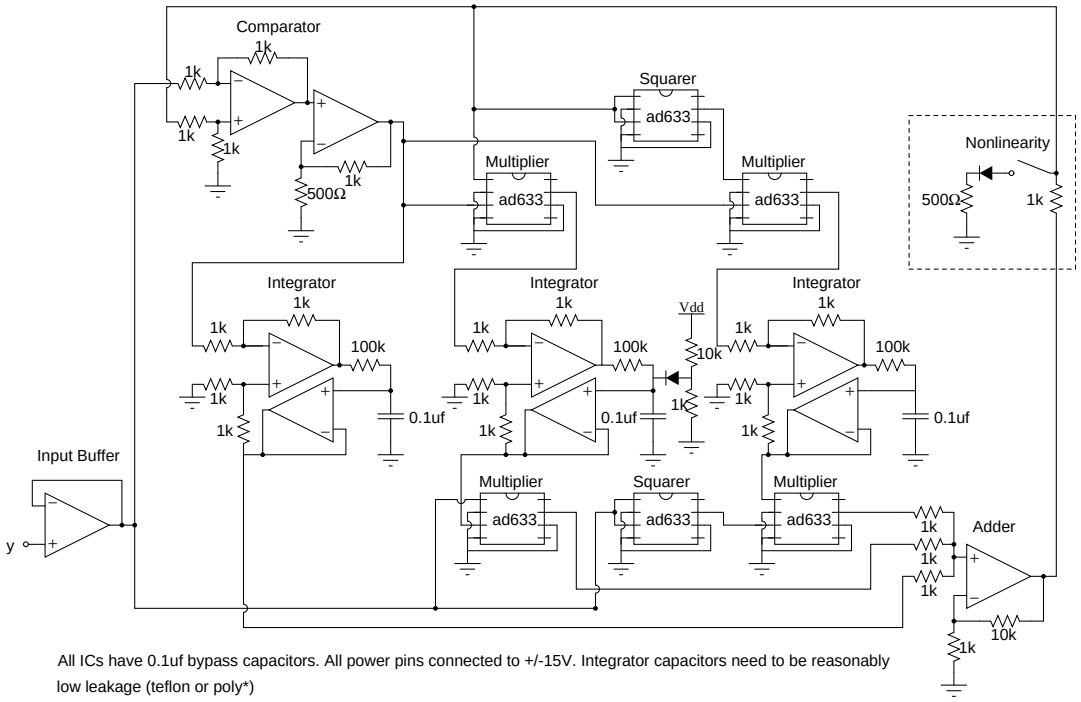


Figure 4-5: Shown is the detail circuit schematic of the test circuit for simplified modeling and linearization. All operational amplifiers are the NJM062.

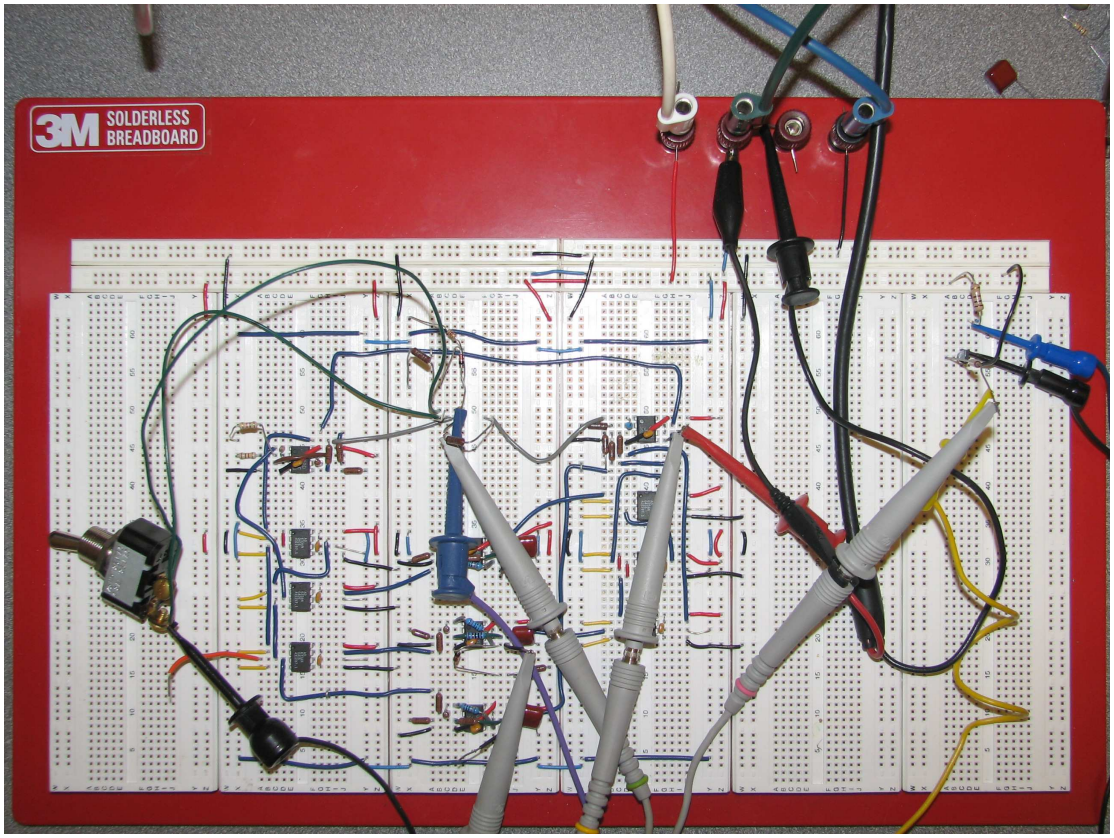


Figure 4-6: Shown is a photo of the test circuit for the simplified modeling and linearization.

4.4.1 Properties of the Circuit

Notice that if the nonlinear blocks are inaccurate, the system will still build the best possible model, simply on a different basis (assuming good matching). If they are slightly mismatched, or if the multipliers have large-signal error, this mismatch or error will simply introduce a small amount of error in θ, so given enough input diversity, and a model space that is capable of approximating the system closely, the system will still converge to the correct answer[3]. For accuracy, the system is primarily relying on the multipliers having correct behavior near the origin (outputting the correct sign based on the sign of the input). Large signal error simply introduces a corresponding error in θ.

As shown, the circuit has a significant limitation. The AD633 guarantees an error of less than 2 percent of full scale, which is about 300mV. In this application, maximum full scale error is not a significant limitation — what is important is good performance around the origin. The AD633 does not necessarily behave very well for small signals. This may lead to a significant DC error in the values of a, b, c. This error can, to some extent, be limited this by increasing the gain of the comparator. This works well, but if the gain is too large, the system may run into clipping in the error signal. If the error signal clips, the system may run into stable modes of the circuit where it is operating entirely incorrectly. It is possible to eliminate these modes by starting near a reasonable value (this is a secondary purpose of the diode/resistor network on the capacitor for the linear term). It is possible to entirely eliminate this problem if the gain of the comparator is less than $\frac{1}{2}$, but at that point, the system runs into the limitations of the AD633. This should not be an issue in integrated implementations because it is comparatively easy to build a multiplier with good small-signal behavior (e.g., a Gilbert cell).

[3]The original matched pair of constraints circuits did not have this property. Rather, it comes from using y_{act} to compute the sign of the direction of descent.

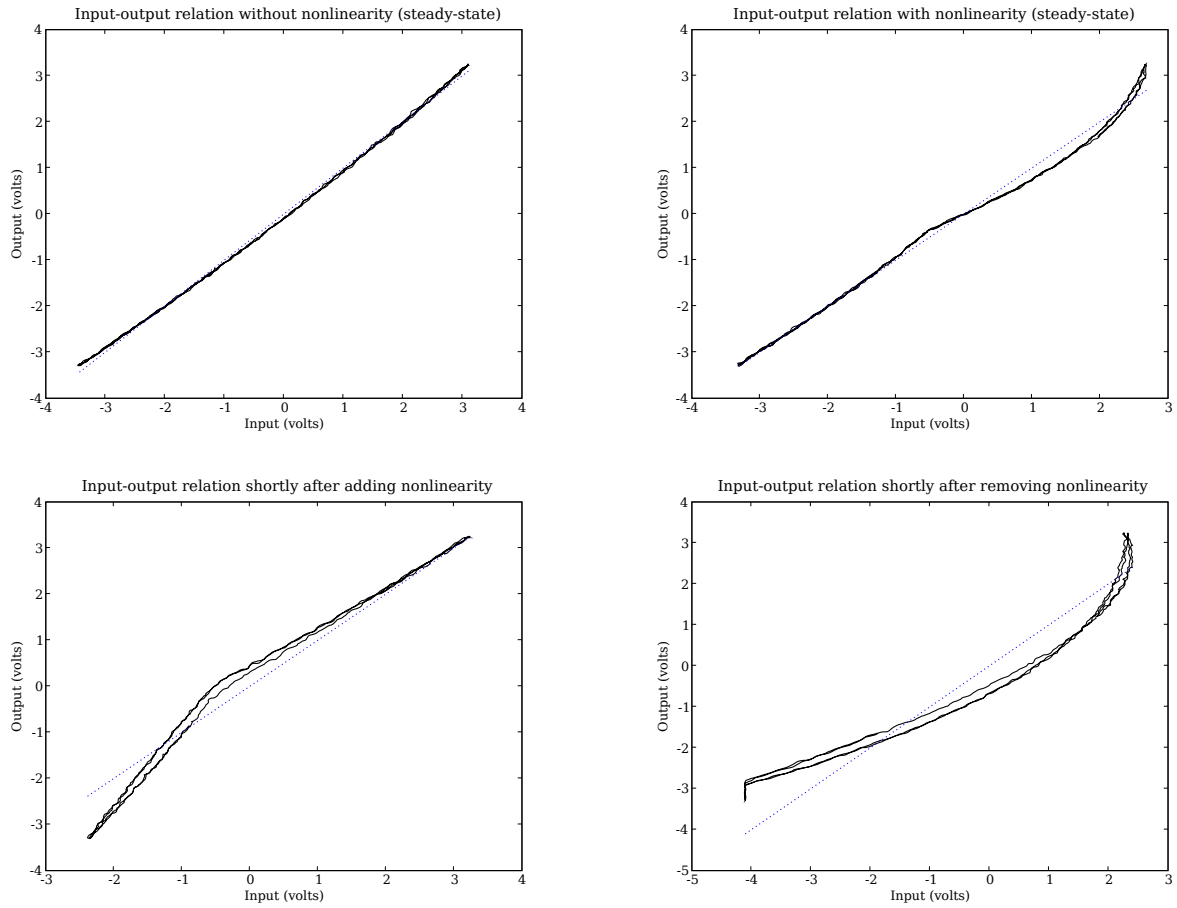


Figure 4-7: Input-output relationship of simplified circuit with a simple diode nonlinearity. Circuit was driven with a 350Hz, $6.5V_{PP}$ triangle wave

4.4.2 Experimental Results

The circuit was tested with a 6.5V, 355Hz triangle wave. Results are shown in Figure 4-7. In the plots immediately following the addition or removal of the nonlinearity, the circuit is adapting to the new transfer function. As a result, the lines do not entirely overlap.

Convergence time is approximately 5 seconds, primarily due to the large node capacitors.

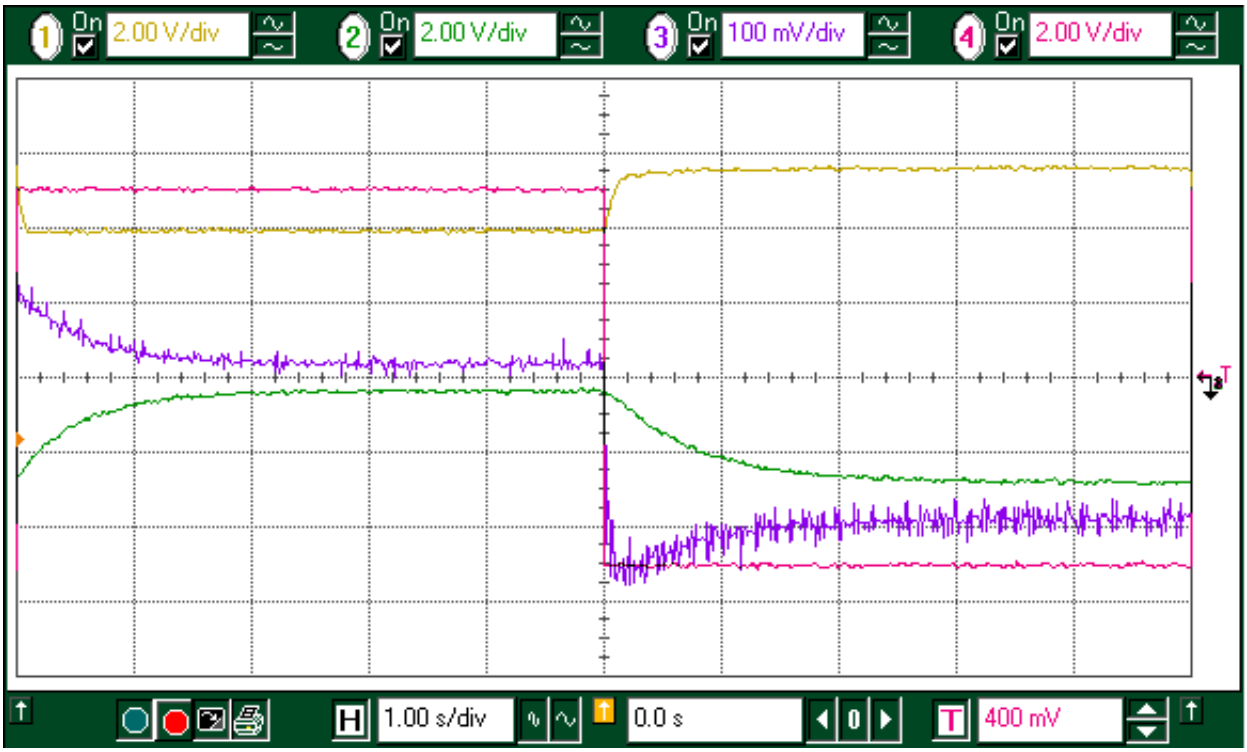


Figure 4-8: Scope traces showing system dynamics of the linearizing system with the addition and removal of the non-linearity. Channels 1-3 show the time evolution of the capacitor voltages. Channel 4 shows the status of the non-linearity.

Chapter 5

Conclusion

This dissertation explored the design and some of the uses of a methodology for the design of bidirectional analog circuits. There are several possible future steps:

- Developing similar methodologies
- Developing more high-level applications for this methodology
- Applying this methodology to specific applications
- Further developing theory

The first task was discussed in the introduction — the methodology explored in the dissertation is one of many possible methodologies for building complex systems. Three other major ones are in use today: digital abstraction, passivity, and incremental passivity. Many others are possible. Exploring these is the broadest area of future work.

Next, this dissertation showed how the methodology could be applied to a small number of high-level tasks: solving systems of equations, function fitting, and linearization. It is likely that many other high-level applications remain. In particular, the behavior of systems was only explored in two cases — the limit cases of slow signal speeds (in which case, the system would solve static mathematical programming problems), and fast signal speeds (in which case, the system would solve dynamic

mathematical programming problems, typically for modeling[1]). It has not been explored for systems with intermediate switching speeds. It has also not been explored for slow switching speeds with underconstrained, non-linear systems (where iterated gradient descent does not necessarily converge). Furthermore, the methodology could be extended in several possible ways. First, it could be extended to to hold dynamic constraints. For instance, an oscillator could be expressed with the constraints[2]:

$$x^2 + y^2 = 1$$

$$\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 = 1$$

$$\left(\frac{d^2x}{dt^2}\right)^2 + \left(\frac{d^2y}{dt^2}\right)^2 < 10$$

Furthermore, the applications presented in this thesis (as well as any extensions) need to be applied to real-world problems — improving specific circuits. This chapter will explore several of these.

There are a number of possible theoretical extensions to this thesis — using parallelism to improve performance, tighter stability criteria, controlling and modeling systems with memory, as well as a number of others. Several of these will also be touched on in the conclusion.

5.1 Improved Operational Amplifier

The linearization presented in this dissertation is independent of traditional feedback techniques. In an operational amplifier, the individual stages could be linearized through feed-forward linearization without significantly affecting feedback linearization. The techniques would, invariably, reduce bandwidth. Nevertheless, properly implemented, this reduction may not be significant.

Consider a conventional operational amplifier, as shown in Figure 5-1. In most

[1] Non-modeling applications are an area remaining to be explored.

[2] The last of these constraints is simply to stop the system from arbitrarily switching directions, and would probably be more concisely expressed in a circuit implementation.

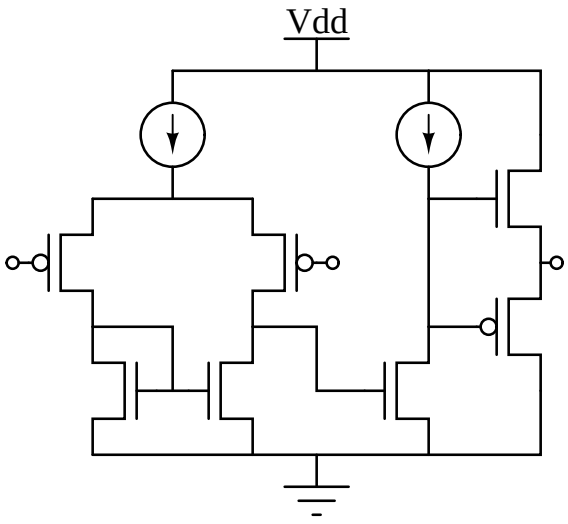


Figure 5-1: A conventional operational amplifier.

cases, the devices of the operational amplifier are much greater than minimum size. It is possible to use small size buffers to monitor the voltages on nodes of the operational amplifier without significantly increasing capacitive loading. As a result, monitoring each stage with a modeling block, as shown in Figure 5-2 can be implemented with only a small reduction in bandwidth.

This monitoring gives an estimate of the gain, offset, level of distortion and type of distortion of each stage. Note that the modeling blocks only need to operate at signal speeds, not at the full bandwidth of the operational amplifier. Conservatively, the next step is to construct a constraint block (or possibly, a feed-forward calculation) that will compute bias currents to minimize distortion based on the measured levels of gain and distortion, as well as estimates of temperature, doping level, oxide thickness, and other device-to-device variations. This would involve minimal changes to the operational amplifier design.

More aggressively, the stages could be modified in such a way that the bias currents could have more significant control over distortion. As mentioned, traditional circuit design typically uses devices much bigger than minimum dimensions, placing the same device many times in parallel, as shown in Figure 5-3. Instead, many different stages may be placed in parallel, as shown in Figure 5-4. Here, controlling the bias

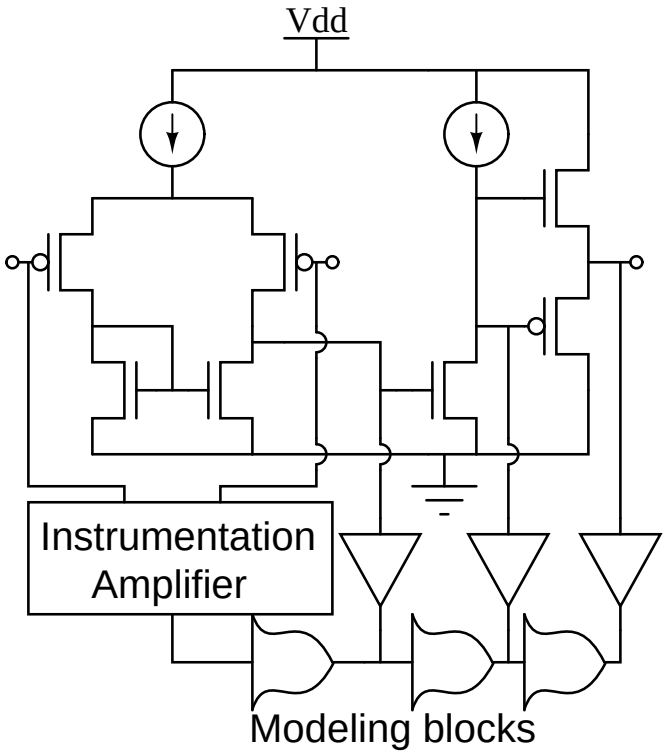


Figure 5-2: Modeling blocks can be added to a conventional amplifier to monitor the distortion of each stage. Note that the modeling blocks only need to operate at signal speeds, not at the full bandwidth of the operational amplifier. Also, note that they are monitoring open-loop, rather than closed-loop distortion of each stage, and so do not need to be especially accurate.

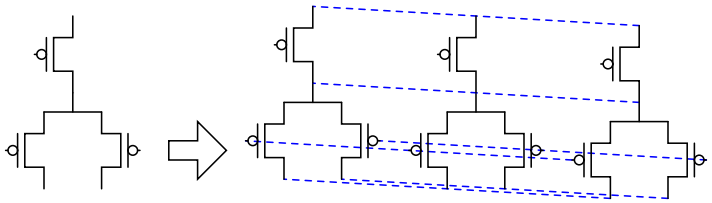


Figure 5-3: The standard way to improve matching of is to simply include many of each transistor in parallel, as shown with the differential pair above.

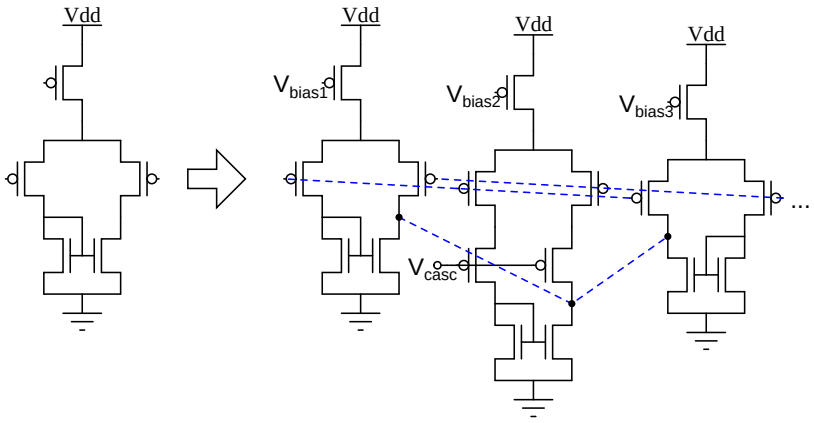


Figure 5-4: This circuit, instead of using the same differential pair in parallel as in Figure 5-3, uses different differential pair designs. While most of the differential pairs are in the conventional configuration, several may be backwards. If the different blocks have different levels and types of distortion, combining them in different levels by controlling the bias currents may allow for the creation of a super-linear differential pair.

currents lets different topologies have different levels of influence on the output. If some reduction in gain can be tolerated, then high distortion, low gain stages may be added in reverse as well. These could remove significant amounts of distortion, while only slightly reducing gain. Then, the auxiliary constraint circuit could create an optimal combination of topologies that would give much lower distortion than any individual topology.

Even more aggressively, a second modeling circuit, with a much slower timescale, may be able to infer the relationship between the bias levels, the distortion, the gain, the frequency response, and other properties. Then, a matched constraint circuit would be able to minimize distortion, maximize gain, and maximize bandwidth. A

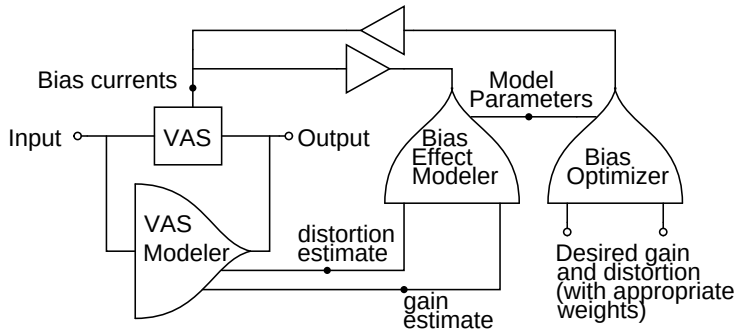


Figure 5-5: This is a possible design for a super-linearized stage. Here, the VAS modeler block builds a model of the VAS stage, used to compute gain and distortion. The bias effect modeler monitors the bias currents, gain, and distortion, and builds a model for how the three relate around the current bias point. Finally, the bias optimizer adjusts the bias currents to minimize distortion and maximize gain, based on the model parameters from the bias effect modeler.

block diagram of this design is shown in Figure 5-5. Designing this system poses some difficult engineering problems because the circuit that models the relationship between bias currents and distortion needs to operate over a very long timescale — much greater than that of the one that characterizes distortion and gain. In addition, the circuit will tend to hold the bias currents fairly constant, whereas the modeling block needs some signal diversity to accurately model the effects of the bias currents on circuit operation. Injecting noise would solve this problem, but may lead to unreasonable levels of noise on the output of the amplifier. Nevertheless, most likely, these issues could be resolved. This circuit is an example of the type of complex system that this methodology may allow.

5.1.1 Improved Matching

When simply scaling components to improve matching, error due to mismatch scales as $\frac{1}{\sqrt{n}}$, where n is the number of components (or the area of the device). Since the system now has control over which components are in the system, a natural extension is to use them to improve matching. Ideally, this would be done by smoothly adjusting bias levels to stages mismatched in both directions, such that the overall system was balanced at matching. Alternatively, this could be achieved by switching bias currents

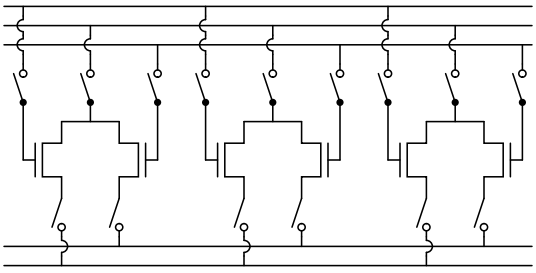


Figure 5-6: If the system switched in appropriate pairs of devices to minimize mismatch, it could achieve much faster order of growth of matching with increased die area, but with a high (but linear) overhead.

on and off, as shown in Figure 5-6.

Now, instead of matching on the order of $\frac{1}{\sqrt{n}}$, with even a very simple switching algorithm, matching error scales on the order of $\frac{1}{n}$. Assume there is one variable for which matching matters, and that it has a Gaussian distribution. Ignore the outlying elements, and only take the m transistors that are closest to being matched. Add one pair of transistors from this set. See in which direction the system is mismatched, and add a pair of transistors mismatched in the opposite direction. Repeat this process until there are no devices remaining. At the end of the process, the mismatch will be at most that of a single pair of transistors, and since the number of transistors is proportional to n, this gives mismatch scaling on the order of $\frac{1}{n}$.

By adding transistors from the opposite quadrant from where the system is currently operating, this generalizes to the case of matching multiple parameters. This generalization loses a significant constant factor in the matching (a function of the number of parameters we are matching), but still maintains $\frac{1}{n}$ scaling.

This algorithm is very simple. Increasing algorithmic complexity (and therefore, constant overhead) allows for much better matching. If the transistors are first ordered by decreasing mismatch, overall mismatch is proportional to the mismatch of the last pair added. Assuming that the probability density around zero mismatch is non-zero, and that the density function is continuous, this gives mismatch error scaling of $\frac{1}{n^2}$.

An optimal set of transistors would give mismatch dramatically lower than even

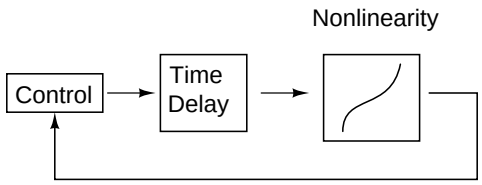


Figure 5-7: A system with a time delay

this, but algorithms for determining an optimal set are computationally complex[3].

5.2 Controlling Systems With a Time Delay

Systems with a time delay, such as the one shown in Figure 5-7, are difficult to control with a traditional feedback loops. This limitation is fairly fundamental — the loop response time is limited by the delay. In many cases, the primary reason for the control system is to compensate for an unknown nonlinearity rather than disturbance rejection. In these cases, a modeling method, such as the one presented in this thesis, may offer a superior alternative to traditional control systems.

5.2.1 Cartesian Feedback

In RF amplifiers, there is a tradeoff between linearity and power efficiency. Switching amplifiers have high power efficiency, but introduce significant distortion. Linear amplifier topologies, in contrast, have poor power efficiency. Cartesian feedback[2] is a technique for reducing the distortion of a non-linear, power-efficient RF amplifiers. In Cartesian feedback, an RF transmitter is given input signals for the in-phase and quadrature components (typically called I and Q). These signals are then modulated, stepped up to RF, and amplified to drive an antenna. The high-amplitude antenna signal is then stepped down and demodulated. A feedback loop compares the desired I and Q components to the actual demodulated ones, and adjusts the input to the amplifier to compensate for error. This system is shown in Figure 5-8

[3]This is very similar to the knapsack problem, and the bin packing problem. Both are computationally difficult, but reasonable approximation algorithms exist for both.

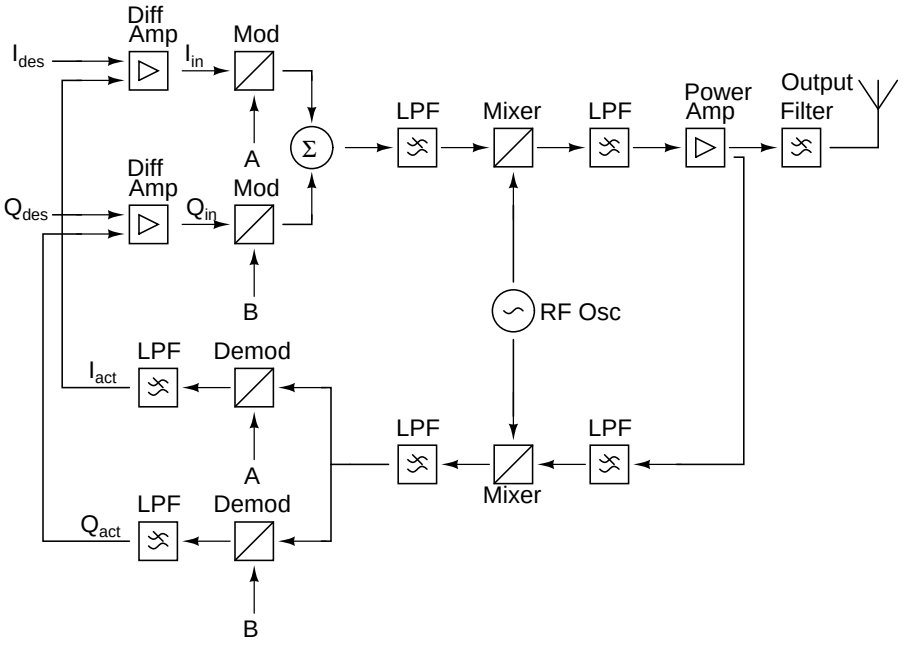


Figure 5-8: This is a block diagram of a Cartesian feedback system. The output from a nonlinear RF amplifier is demodulated, and compared to the desired output. A feedback loop adjusts the input to compensate for error.

Cartesian feedback is fundamentally limited by control systems response time. The modulator and demodulator introduce a substantial delays, which significantly limits the design of the control systems. When a new symbol is transmitted, it takes a substantial amount of time for the control system to adapt. A number of papers attempt to solve this problem either by dynamically building a model of the nonlinearity instead of traditional controls, or by using a periodic calibration cycle that figures out the appropriate values of I_{in} and Q_{in} to give the desired output values I_{des} and Q_{des}, and storing it in a digital memory[8].

The methodology in this thesis may offer a compact way to build a circuit capable of building a model of the nonlinear amplifier in a Cartesian feedback system, and predistorting to compensate for the nonlinearity.

5.3 Other Applications

There is a broad range of other applications for the work in this thesis. Virtually all analog circuits can improve in performance if bias currents are calculated based on measures of temperature, MOSFET threshold voltage, oxide thickness, and other parameters of the circuit. This sort of calculation is very efficiently expressed as a constrained optimization problem. It is likely that circuits developed using this methodology could solve these optimization problem using much less die area than a digital solution. It is not known how it would compare to a closed-form approximation, but most likely, it would give better results, but use more die area.

The constraint circuit may also be directly applicable to solving Bayesian belief networks, and in particular, to building an analog soft coder. This should generate the same answers as the message passing protocol. Message passing is widely used in industry, and gives a useful approximate answers, but does not handle complex correlations needed for more sophisticated classes of statistical inference applications. Nevertheless, due to the complexity of full statistical inference, it is a frequently used approximation.

The basic transformer circuit may have other applications in isolation. Since it does not need to maintain power equality, it can translate microvolts and picoamps into kilovolts and megaamps. It may give a logical way for a small circuit to control a power circuit, or to couple two motors together for remote control with tactile feedback, as with a synchro motor or a selsyn, or for haptic applications. Given the low-cost of a digital solution, it is not clear whether this would be a useful application.

5.4 Future Theoretical Work

In addition, there is a large number of ways to potentially extend the theory of this work. First, there may be ways to integrate it with digital circuits to improve the performance of those digital circuits. Second, there is a number of ways to improve circuit performance by sacrificing compactness and simplicity. Parallel modeling cir-

cuits can achieve higher performance by observing several samples at once. More complex compensation schemes can also improve convergence time, and reduce noise. On the other hand, the circuits may be further simplified by sacrificing performance. Finally, and perhaps most importantly, the modeling and linearization system need to be generalized to systems with memory. This section will discuss possible techniques for achieving the above.

5.4.1 Integration with Digital Techniques

The circuits developed in this thesis were intended to be used as building blocks for analog circuits. Nevertheless, they may also be applicable to solving complex constraint propagation problems directly through analog, rather than through the use of digital computation. It may also be used for initializing a digital solver near the optimum, so that the digital circuit only needs to operate over a region where the system is well approximated by a quadratic form[4]. Many non-linear convex optimization problems are exceedingly slow with digital computation. If a problem has a large number of dimensions, and the minimum is at the bottom of a long, winding valley, the solution is often computationally infeasible. A direct analog solution may approximately solve these problems using a more reasonable amount of time, power, and complexity. Douglas, *et al.* demonstrates that performance of equation solvers could be improved by the use of an analog solver to compute a starting point for a digital solver[13][25].

Pursuing this work requires a comparison of the cost, speed, and power usage of digital and analog solutions. If the analog solutions were found to be competitive with digital, the next step would most likely consist of the design of a reconfigurable analog computer consisting of programmable constraint blocks in an FPGA-like architecture. In the unlikely case this approach could be successful, it would be widely applicable to a wide variety of computational problems.

In addition, modern digital processors perform many probabilistic machine learn-

[4]Digital solvers can solve optimization problems over quadratic forms exactly in a constant number of steps, where the constant depends on the dimensionality of the problem.

ing functions. The processor pipeline typically includes a branch prediction engine. Caches use sophisticated algorithms to predict which cache lines are most likely to be needed in the future. These often yield incorrect answers, and they are evaluated in terms of cost, speed, and proportion of correct to incorrect guesses. In contrast to traditional digital logic, this logic does not need to be deterministic. As a result, it may be possible to reimplement these learning algorithms in analog, achieving better performance, speed, and cost than traditional digital circuits.

5.4.2 Improved Performance

The modeling circuit is designed to be compact enough to be useable in many places where digital logic would be size or cost prohibitive. Some of the design decisions made to achieve this goal limit circuit performance. There are a number of ways to achieve greater performance at the expense of increased cost and complexity.

Spatial Parallelism

The modeling circuit is limited in performance in part because it only examines one data point at a time, and has no memory beyond the current estimate as stored on the capacitance on the parameter lines. The use of parallelism can overcome this limitation, and substantially improve convergence speed of the modeling circuit. The circuit shown in Figure 5-9 implements a parallel modeler that uses a large number of learning blocks connected together, each connected to a different input. If the samples are generated serially, the system can look at multiple samples by passing the signal through a delay, and having blocks look at samples from the past, as shown in Figure 5.4.2. The delay can be implemented as a real delay, or as a clocked sample-and-hold.

In the discrete time case, if a system without parallelism projects into an m-dimensional hyperplane[5] (where m is the number of parameters in the function fit), an

[5]This section uses the terminology and notation introduced in Section 3.1, and in particular, Figure 3-4. While this notation, and therefore the arguments in this section, are specific to general linear regression, analogous arguments exist for most nonlinear systems

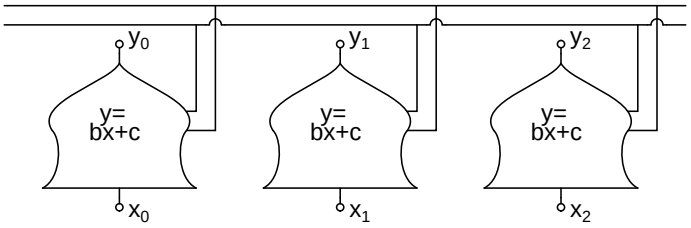


Figure 5-9: If the system is monitoring several identical (or similar) systems, it can attach a modeling block to each of them, but use a common model for all of them. Combining information from multiple, matched systems in this way can improve convergence speed and accuracy.

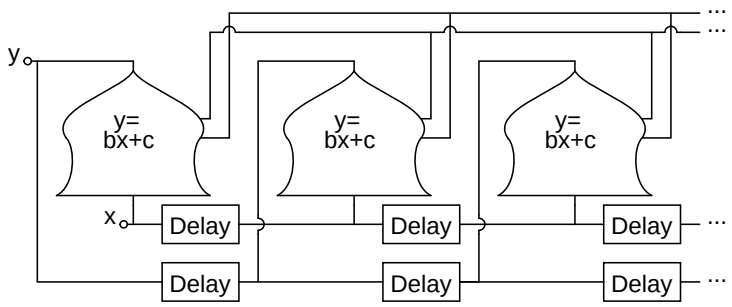


Figure 5-10: Spatial parallelism improving convergence speed in a single circuit.

n-way parallel system would project into an $m-n$ dimensional hyperplane (assuming $m > n$). The expected angle between a line and a random j dimensional hyperplane drops off very quickly as a function of j. Reducing j dramatically increases the expected angle, and hence, dramatically improves the rate of convergence.

If $m \leq n$, the system will be fully constrained or overconstrained, and as a result, it will be able to arrive at the least-squares estimate of the parameters in just one step.

Similarly, in the continuous time case, reducing dimensionality improves performance. A conventional serial modeling system has a performance tradeoff with integration speed. A system with a slow rate of integration moves slowly, and so converges slowly. A system with a fast rate of integration tends stays close to the hyperplane of the current input, so θ is always small, and so the system moves in a direction nearly perpendicular to the center. This also gives low convergence speed. Using parallelism, the system can combine information from multiple samples with a large θ between them, so the system can move both quickly, and in a good direction.

It is also possible to combine temporal and spatial parallelism in more clever ways. Take the case of Cartesian feedback in a MIMO[6] system. A MIMO system typically has a number of matched RF amplifiers. Since these amplifiers are nearly identical, the system ought to be able to exploit calibration information from one to calibrate the remaining amplifiers more quickly and accurately. There may, however, still be minor mismatches. As a result, the system should weigh the data points from the local amplifier more than those of the remote ones. The system shown in Figure 5-11 implements this using a soft equals constraint. Here, if a single transmitter is transmitting the same symbol over and over, the information from the remaining ones will prevent the state from drifting. On the other hand, depending on the hardness of the soft-equals constraint, local information can arbitrarily strongly override remote information. The soft-equals constraint may be implemented with a resistor, as explained in Appendix C[7].

[6]Multiple Input Multiple Output is a technique for using an array of antennas and multipath to improve performance of RF systems.

[7]For more interesting dynamics, the soft-equals block may also be implemented as an RC low-pass

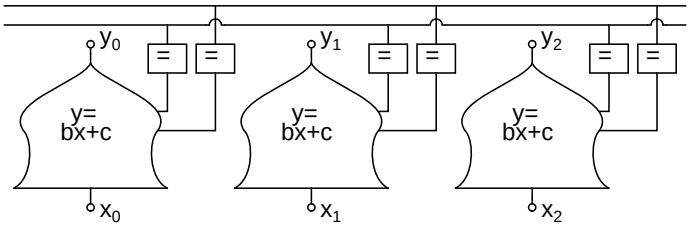


Figure 5-11: Using a soft-equals to combine information from related circuits, such that the information from the local circuit is weighed more heavily than from parallel circuits.

Formalizing the tradeoffs given by parallelism, as well as engineering systems to exploit parallelism is an area of future work.

Nonlinear Compensation

The previous chapters dealt with basic gradient descent where the rate of descent was proportional to the slope of the objective function. This need not always be the case. Dynamically controlling the rate of integration could allow for greater convergence speeds in both the static and dynamic cases, while maintaining stability, and improving accuracy.

Section 2.2.1 used a maximum bound on error currents. In practice, error currents will depend on the current state of the system, and in particular, on the level of non-error currents. Similarly, in Section 2.2.3, the size of capacitors needed to achieve high frequency stability depended on the gain of the blocks. In a nonlinear system, the gain will vary with the operating point. In the static system described in Chapter 2, gain is set to achieve stability at the most conservative operating point of the system. In a system with dynamic gain control, it could be set based on the current operating point, thereby achieving substantially greater convergence speed.

Dynamic gain control can also help improve accuracy. For a nonlinear system (or even linear systems where the inputs have non-Gaussian distributions), the optimal control is nonlinear. The theory for this is best developed in the context of phase locked loops for FSK receivers. Here, the noise signal is Gaussian, while the signal

filter, with any number of resistors connected in series, with capacitors to ground at the junctions.

consists of discrete symbols. When the feedback loop is close to equilibrium, the feedback loop has a slow integration time constant to minimize noise susceptibility. When it is far from equilibrium, it has a short time constant so it can quickly slew to the next symbol. This methodology includes nonlinear constraints, and in modeling applications, is targeted at systems where the input has a non-Gaussian distribution. The optimum compensation is, therefore, in general nonlinear.

Implementing nonlinear compensation involves either adjusting the current levels, or adjusting the capacitance values. Adjusting the capacitance values without affecting the voltages on the capacitors is difficult, so this would typically involve scaling current levels. In addition, to compute the appropriate current scalings, the system would most likely also need to monitor current levels.

Creating a coherent theory for nonlinear compensation is an area of future research. It has the promise to improve convergence speed, reduce noise levels, and improve accuracy, at the expense of increased circuit cost and complexity. Since most systems have non-Gaussian inputs, doing this in a more general way would have a broad range of applications.

5.4.3 Further Simplifications

Further simplifications are possible. For instance, it may be reasonable to approximate $f_i(y_{act}) \approx f_i(y_{des})$ in order to remove the duplicate f_i blocks. This introduces an additional error ψ in the direction of the vector $\langle \dot{c}_1, \dot{c}_2, \cdots, \dot{c}_n \rangle$. If the $|\psi| < |\theta|$ in the perfect model assumption, or $|\psi| < |\theta| + \left|\arcsin\frac{r}{l}\right|$ in the imperfect model assumption, the system will still converge. Otherwise, it may not converge. This gives a further simplified system shown in Figure 5-12. This only approximates the original dynamics well if the model is initially quite good, and the system is trying to improve it further. For instance, if the basic system is fairly affine, and the system is trying to model a small amount of nonlinearity, it can be constructed such that all possible models are nearly linear, and so initially fairly good.

More radically, each multiplier may be replaced with a switch, as shown in Figure 5-13. Assume a very large number of columns. The integrator calculates whether

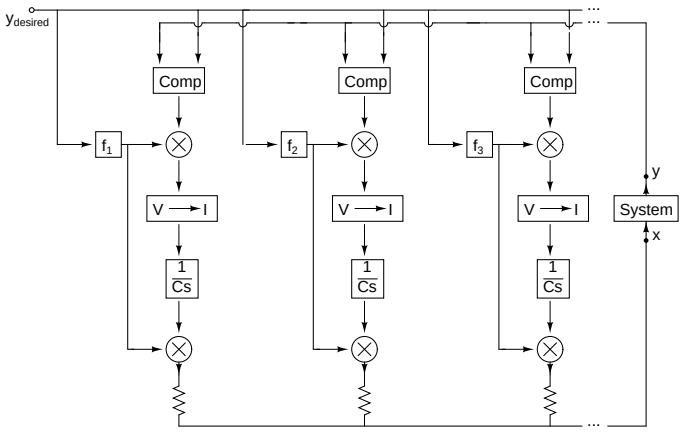


Figure 5-12: Simplified learning system assuming $f_i(y_{des}) = f_i(y)$.

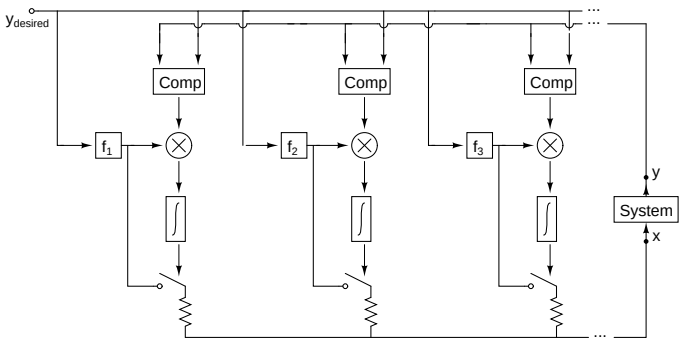


Figure 5-13: Simplified learning using switches and assuming $f_i(y_{des}) = f_i(y)$.

each individual unit is helping or hurting the performance of the system overall, and either switches it in or out based on the result.

As shown, this system is not very useful — it will trivially sit in local minima. It may also have modes of behavior where large numbers of switches switch simultaneously, causing unstable behavior. For instance, if the gain is too low, the integrators for several switches may switch at the same time to raise gain. At that point, the gain may be too high, at which point, again, many of the switches may switch back. Nevertheless, using a switch (or a hybrid in between a switch and a multiplier) would lead to a somewhat simpler circuit. By using a very large number of individually imprecise columns (with very small transistors), and using clever tricks (e.g., injecting noise) to avoid local minima and awkward modes, the system may be able to achieve formidable performance while relying on very small, poor devices.

5.4.4 Generalizing to Dynamic Systems

As presented, the method is restricted to modeling and controlling memoryless nonlinear systems. There are a number of possible ways to generalize the theory to modeling and controlling systems with memory. Exploring these is an area of future work, and this section will outline several possible approaches.

Linearization as a Special Case of LTI Controls

It is straightforward to see that the system presented is, in a sense, a general case of traditional LTI controls. Take one of the simplest possible models: $y = \epsilon x + c$. This is shown in Figure 5.4.4. This behaves identically to a simple dominant-pole compensated linear feedback loop, as shown in Figure 5.4.4.

Replacing the capacitor in Figure 5.4.4 with an arbitrary filter allows the system to emulate an arbitrary LTI controller. This suggests that, for systems where the dynamics are well understood, but that have an unknown nonlinearity, the methodology may provide a powerful framework for nonlinear control system design. Replacing the capacitors in the circuit shown in Figure 1-4 with arbitrary filters, as shown in Fig-

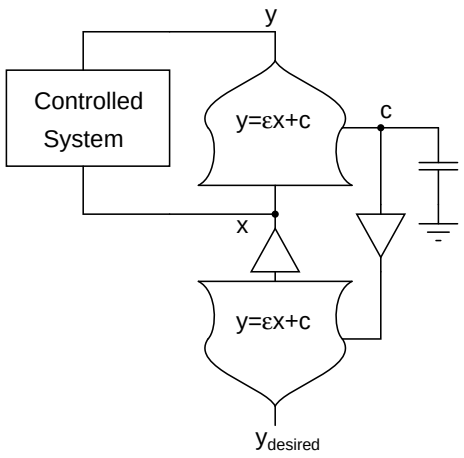


Figure 5-14: This is a simple linearization block that is equivalent to a dominant-pole compensated LTI controller.

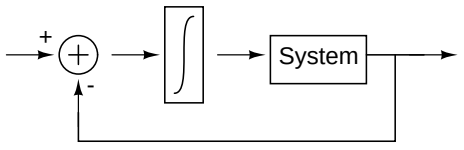


Figure 5-15: This is a block diagram of a dominant pole controller for an LTI system.

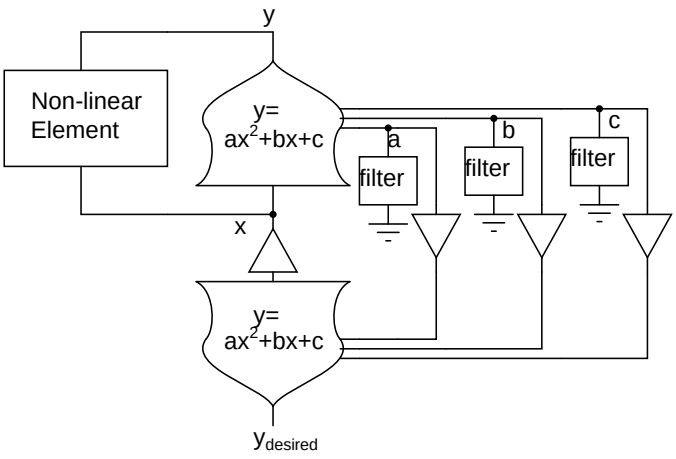


Figure 5-16: This is a possible controller for nonlinear systems with memory.

ure 5.4.4 may allow the design of controllers that can handle systems with memory in the same way as traditional LTI controllers do, while compensating for nonlinearities in the same way as the controllers for memoryless systems described in this dissertation.

The mathematics for understanding the dynamics of such a system are complex, and have not yet been explored.

Kalman filters

The methodology may also be suitable to building a Kalman-like filter. If the transfer function of the system is known at design time, a modeling system may be able to project the current estimate part way, based on the current data point. The most naive implementation of this concept would not maintain the ideality of the Kalman filter — the system would not maintain an estimate of the error, and would not adjust how far to project based on that estimate. This implementation would still keep track of a good estimate of system state, and would have the advantage of being very compact. It is likely that an implementation of a full Kalman filter would also be possible, although at increased complexity.

Determining a Transfer Function

For a limited set of systems, the constraint solving circuit may be able to determine a transfer function explicitly. Consider an discrete-time system with input $\mathbf{u}$, output $\mathbf{y}$, no hidden state variables, and a transfer function of the form:

$$\mathbf{u}_t = \sum_i C_i \mathbf{y}_{t-1} + \sum_i D_i \mathbf{u}_{t-1}$$

By observing $\mathbf{u}_t$ and $\mathbf{y}_t$, the constraint solver can reconstruct C_i and D_i. Once these are reconstructed, this can be used to control the system. A similar result can be shown for simple continuous-time systems. Consider an LTI system of the form:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

Where $\mathbf{x}$ is the state vector, A and B are matrices, and $\mathbf{u}$ is the input. If the system could monitor $\mathbf{x}$ and $\mathbf{u}$, it could trivially determine the matrices A and B.

In most cases, however, the systems have hidden state variables. Consider the general case of continuous time systems of the from:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

$$y = C\mathbf{x} + D\mathbf{u}$$

Similarly, consider discrete time systems of the form:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{u}_t$$

$$y = C\mathbf{x}_t + D\mathbf{u}_t$$

Here, a modeling block could monitor the visible outputs, and try to learn the values of the matrices A, B, C, and D, as well as the compute and track the current value of $\mathbf{u}$.

Most likely, in the general case, this system would tend towards a local minimum,

and may not be able to adequately track the desired variables. There may, however, be cases where the system can stay near the global minimum, in particular if some properties of the matrices A, B, C, and D are known, and a constraint block can hold those within those bounds.

Characterizing the classes of models for which this type of technique may converge, and developing design techniques to select an appropriate model for a given problem domain is an area of future work. Using this to design optimal control systems is also an area of future work.

5.4.5 Dynamic Constraints

The system described in this paper is limited to the design of static constraints. The constraints can neither change over time, nor are they useful for holding velocities of voltage. It may be possible to generalize this methodology to dynamic systems by replacing the traditional static objective function $L(x_1, x_2, ..., x_n)$ with a dynamic objective function $L(x_1, x_2, ..., x_n, \dot{x}_1, \dot{x}_2, ..., \dot{x}_n, t)$ As stated in the introduction to this chapter, this may allow for the creation of oscillators and other dynamic systems. Furthermore, it may be possible to design systems where the constraints themselves are change over time, most likely in a manner controlled by an external circuit. Theory and applications for both types of dynamic constraints are left as an area of future work.

5.5 Summary

This dissertation presented a methodology for analog circuit design that:

- Is based on constraints.
- Relies on a bidirectional flow of information.
- Uses a local, rather than global, stability criterion.

Like digital circuit design, passive circuit design, and incrementally passive circuit design, it offers a technique for designing complex, stable circuits. It has been applied to the problems of solving systems of equations, building models, and linearization.

Circuits designed under this methodology are moderately compact, and may be further compiled into a more efficient form. Compiled, even systems performing complex functions may be compact. For instance, a system that can model a nonlinearity by a quadratic Taylor approximation and predistort to compensate for the nonlinearity can be built with less than a dozen major blocks (operational amplifiers and Gilbert multiplier cells). This specific compilation generalizes to building a broad range of complex models, and as the complexity of the model scales, the circuit develops some interesting properties. Most significantly, the compiled circuit has a level of fault tolerance, where if blocks fail in a broad set of ways, the remaining blocks will act to compensate for that failure.

This methodology is still in an early stage. Although the theory has been substantially developed, a number of tasks still remain. Most importantly, it needs to be applied to specific applications. In addition, there are a number of possible theoretical extensions. It is also the hope that other methodologies for the design of complex, interconnected analog systems may be created in the future.

Appendix A

Cast of Characters

The notation in parts of this paper is, in some cases, not consistent with that of the relevant domain. The difficulty with achieving consistency is that this thesis spans multiple domains — circuits, control systems, machine learning, and mathematical programming. Each domain has different notations and conventions. As a result, it was impossible to be consistent with the conventions of all domains. This appendix is included so that readers can quickly reference the meaning of common variables throughout this document.

A, B, A_i, B_i	$\cdots$	High frequency error currents
B_1	$\cdots$	Region of convergence (level curve of L around B_1')
B_1'	$\cdots$	Region where the objective function is not Lyapunov
B_2	$\cdots$	Region of linearity around minimum
C	$\cdots$	Capacitance
C_j	$\cdots$	Capacitance on node j
c_i	$\cdots$	Training parameter i
I	$\cdots$	Current
L	$\cdots$	Objective function (usually, global)
L_i	$\cdots$	Objective function of ith constraint block
$L_{\mathcal{N},j}$	$\cdots$	Sum of objective functions around node j
S	$\cdots$	Maximum value of objective function in stable region: $\sup_{B_1} L$
t	$\cdots$	Time
V	$\cdots$	Voltage
V_i	$\cdots$	Voltage on node i
ϕ_j	$\cdots$	$\sum_{i \in \mathcal{N}_j} \frac{dL_i}{dV_j}$
σ_j	$\cdots$	Error current onto jth node
σ	$\cdots$	Maximum error current: $\sup_j \sigma_j$
x, y, z	$\cdots$	General-purpose variables

Appendix B

LaSalle's Theorem

LaSalle's Theorem is a basic result in control systems, closely related to Lyapunov stability. Since readers with an electrical engineering background may not be familiar with LaSalle stability, this appendix summarizes the theorem. A more complete, well-written introduction to Lyapunov stability arguments, including LaSalle's Theorem, is found in Khalil[22]. Khalil states LaSalle's theorem as:

Consider the autonomous system:

$$\dot{x} = f(x) \tag{B.1}$$

A set Ω is said to be a *positively invariant set* with respect to (B.1) if

$$x(0) \in \Omega \Rightarrow x(t) \in \Omega, \ \forall t \geq 0$$

Let $\Omega \subset D$ be a compact set that is positively invariant with respect to (B.1). Let $V : D \longrightarrow R$ be a continuously variable function such that $\dot{V}(x) \leq 0$ in Ω. Let E be the set of points in Ω where $\dot{V}(x) = 0$. Let M be the largest invariant set in E. Then every solution starting in Ω approaches M as $t \longrightarrow \infty$.

This result is closely related to Lyapunov's Theorem. The primary difference is that Lyapunov's Theorem concerns itself with stability around a single equilibrium point. LaSalle's Theorem, in contrast, shows convergence into set of points. This property is necessary for handling the underconstrained case of the static constraint system. It is also necessary for the proof of robust stability in the static case, as shown in Section 2.2.1.

Furthermore, some formulations of Lyapunov stability, including the formulation in Khalil, only show stability at the equilibrium point — a local criterion. The stability proofs in this document require global convergence. Global convergence requires either a global version of Lyapunov's Theorem[10], or LaSalle's Theorem.

Appendix C

Sample Circuit Implementations

To give an idea of the procedure that goes into designing constraint blocks, this section shows the design of a number of basic constraint blocks.

C.1 Squarer

A squarer has the constraint:

$$x = y^2$$

With least squares, this gives the objective function:

$$V(x, y) = (x - y^2)^2 = x^2 - 2xy^2 + y^4$$

The derivatives are:

$$\frac{d}{dx} \Longrightarrow 2x - 2y^2$$

$$\frac{d}{dy} \Longrightarrow 4y^3 - 4xy = 4y(y^2 - x)$$

This circuit implementation of this connects a $\frac{1}{2}R$ resistor between a voltage equal to y^2 and the node x, and a $\frac{1}{4}R$ resistor between a voltage equal to $y(y^2 - x + 1)$ and the node y, as shown in figure C-1.

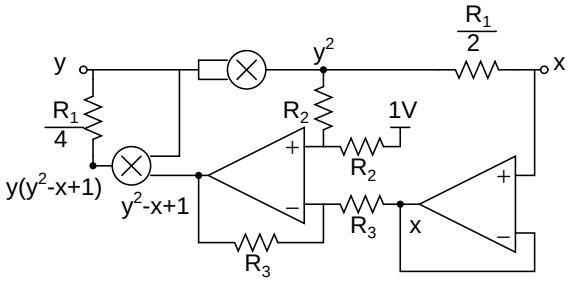


Figure C-1: A squarer circuit implementing the constraint $x = y^2$.

C.2 Exponentiator

To take the exponential, take the constraint:

$$y = e^x$$

Convert this into the objective function:

$$\min(y - e^x)^2 = \min y^2 - 2ye^x + e^{2x}$$

This gives the error currents:

$$\frac{d}{dx} \Longrightarrow 2e^{2x} - 2ye^x$$

$$\frac{d}{dy} \Longrightarrow 2y - 2e^x$$

C.3 Multiplier

To multiply two voltages, take the constraint:

$$x = yz$$

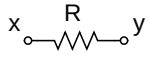


Figure C-2: Soft equality may be implemented as a resistor.

This can be phrased as the optimization problem:

$$\min(x - yz)^2 = \min x^2 - 2xyz + y^2z^2$$

Which gives the currents:

$$\frac{d}{dx} \Longrightarrow 2x - 2yz$$

$$\frac{d}{dy} \Longrightarrow 2yz^2 - 2xz$$

$$\frac{d}{dz} \Longrightarrow 2y^2z - 2xy$$

C.4 Soft Equals

While a wire can be used to hold equality, to enforce a softer equality constraint, start with the relation:

$$x = y$$

Take the optimization problem

$$\min(x - y)^2 = \min x^2 - 2xy - y^2$$

Which gives the derivatives:

$$\frac{d}{dx} \Longrightarrow 2x - 2y$$

$$\frac{d}{dy} \Longrightarrow 2y - 2x$$

This can be implemented with a simple resistor, as shown in figure C-2

Figure C-3: Two diodes used to hold two voltage to be close to each other, but not necessarily identical.

C.5 Approximately Equals

In some cases, it is useful to hold two voltages close to, but not necessarily equal, to each other:

$$|x - y| < c$$

A plausible objective function that approximates this is:

$$\min B\left(e^{C(x-y)} + e^{C(y-x)}\right)$$

Which gives the currents:

$$\frac{d}{dx} \Longrightarrow A\left(e^{C(x-y)} - e^{C(y-x)}\right)$$

$$\frac{d}{dy} \Longrightarrow A\left(e^{C(y-x)} - e^{C(x-y)}\right)$$

Where $A = BC$. The circuit for this is simply two diodes, as shown in figure C-3

This block can be useful in order to force a nearly-monotonic function to move beyond a local minimum.

C.6 Less-Than or Equals

Less than or equals is almost identical to the above block. Take the constraint:

$$x \leq y$$

Approximate this as:

$$\min Be^{C(x-y)}$$

Figure C-4: A diode behaves as a less-than-or-equals constraint.

Which gives the currents:

$$\frac{d}{dx} = Ae^{C(x-y)}$$

$$\frac{d}{dx} = -Ae^{C(y-x)}$$

Where $A = BC$. This can be implemented as a simple diode, as shown in figure C-4. It may be better implemented by a diode in series with voltage source, or as an active "ideal" diode.

Appendix D

Creative Commons Attribution 3.0 License

License

1. **Definitions**

(a) **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

(b) **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

(c) **"Distribute"** means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

(d) **"Licensor"** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

(e) **"Original Author"** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no

individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

(f) **"Work"** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

(g) **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

(h) **"Publicly Perform"** means to perform public recitations of the Work and

to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

(i) **"Reproduce"** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. **Fair Dealing Rights.** Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. **License Grant.** Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

   (a) to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

   (b) to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

(c) to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

(d) to Distribute and Publicly Perform Adaptations.

(e) For the avoidance of doubt:

i. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. **Restrictions.** The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

(a) You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Iden-

tifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

(b) If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the

Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

(c) Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. **Representations, Warranties and Disclaimer**

   UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. **Limitation on Liability.** EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. **Termination**

   (a) This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

   (b) Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under

different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. **Miscellaneous**

(a) Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

(b) Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

(c) If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

(d) No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

(e) This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

(f) The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the

Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at HTTP://CREATIVECOMMONS.ORG/

Bibliography

[1] T. Bonaccio, T. Meng, E. Perea, R. Pitts, C. Sodinii, and J. Wieser. Will Moore's Law rule in the land of analog? In *DAC '04: Proceedings of the 41st Annual Conference on Design Automation*, page 633, New York, NY, USA, 2004. ACM Press. Moderator-R. Rutenbar.

[2] A. Brown and V. Petrovic. Phase delay compensation in HF Cartesian-loop transmitters. *Fourth International Conference on HF Radio Systems and Techniques*, pages 200–204, April 1988.

[3] L. Chua. Global optimization: a naive approach. *IEEE Transactions on Circuits and Systems*, 37(7):966 – 969, July 1990.

[4] L. Chua, C. Desoer, and E. Kuh. *Linear and Nonlinear Circuits*, section 5.5, pages 267–278. McGraw-Hill, 1987.

[5] L. Chua and D. Green. A qualitative analysis of the behavior of dynamic nonlinear networks: Stability of autonomous networks. *IEEE Transactions on Circuits and Systems*, 23(6):355 – 379, June 1976.

[6] L. Chua and L. Gui-Nian. Nonlinear programming without computation. *IEEE Transactions on Circuits and Systems*, 31(2):182 – 188, February 1984.

[7] L. Chua, G. Lin, and J. Lum. The $(p+q)$-port transformer. *International Journal of Circuit Theory and Applications*, 10:335–359, 1982.

[8] S. Chung, J. Holloway, and J. Dawson. Open-loop digital predistortion using Cartesian feedback for adaptive RF power amplifier linearization. *IEEE MTT-S International Microwave Symposium*, June 2007.

[9] J. Cruz and M. Van Valkenburg. *Signals in Linear Circuits*, section 6.3, pages 111–113. Houghton Mifflin, 1974.

[10] M. Dahleh, M. Dahleh, and G. Verghese. *Lectures on Dynamic Systems and Control*, chapter 13. Prepublication draft, 2003.

[11] J. Dennis. *Mathematical programming and electrical networks*. Ph.D. dissertation, Massachusetts Institute of Technology, Dept. of Electical Engineering, September 1958.

[12] C. Desoer and E. Kuh. *Basic Circuit Theory*, section 19.3, 19.6, pages 802–807, 816–822. McGraw-Hill, 1969.

[13] C. Douglas, J. Mandel, and W. Miranker. *Fast Hybrid Solution of Algebraic Systems*. RC Report 13615. IBM TJ Watson Research Center, Yorktown Hts, NY, March 1988.

[14] M. Forti, P. Nistri, and M. Quincampoix. Generalized neural network for nonsmooth nonlinear programming problems. *IEEE Transactions on Circuits and Systems*, 51(9):1741 – 1754, September 2004.

[15] A. Hajimiri. Distributed integrated circuits: An alternative approach to high-frequency design. *IEEE Communications Magazine*, 40(2):168–173, February 2002.

[16] Jayadeva and B. Bhaumik. Comments on 'a naive approach to global optimization' by L.O. Chua. *IEEE Transactions on Circuits and Systems*, 38(7):818 – 820, July 1991.

[17] T. Jelonek, J. Reilly, and Q. Wu. Real-time analog global optimization with constraints: Application to the direction of arrival estimation problem. *IEEE Transactions on Circuits and Systems*, 42(4):233–244, April 1995.

[18] T. Jelonek, J. Reilly, and Q. Wu. Real-time analog global optimization with constraints: Applications to the direction of arrival estimation problem. *IEEE Transactions on Signals and Systems*, 42(4):233–244, 1995.

[19] M. Kennedy and L. Chua. Unifying the Tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of Chua and Lin. *IEEE Transactions on Circuits and Systems*, 34(2):210 – 214, February 1987.

[20] M. Kennedy and L. Chua. Nonlinear networks for nonlinear programming. *IEEE Transactions on Circuits and Systems*, 35(5):554–562, May 1988.

[21] H. Khalil. *Nonlinear Systems*, chapter 6, pages 227–262. Pearson Education, Upper Saddle River, New Jersey, third edition, 2000.

[22] H. Khalil. *Nonlinear Systems*, chapter 4, pages 111–194, esp. 128. Pearson Education, Upper Saddle River, New Jersey, third edition, 2000.

[23] S. Lee, B. Kim, and K. Lee. A novel high-speed ring oscillator for multiphase clock generation using negative skewed delay scheme. *IEEE Journal of Solid-State Circuits*, 32(2):289–291, February 1997.

[24] R. Mallock. An electrical calculating machine. *Proceedings of the Royal Society*, 140:457–483, 1933.

[25] J. Mandel and W. Miranker. *New Techniques for Fast Hybrid Solutions of Systems of Equations*. RC Report 14157. IBM TJ Watson Research Center, Yorktown Hts, NY, November 1988.

[26] M. Minsky and S. Papert. *Perceptrons - An Introduction to Computational Geometry: Expanded Edition*. MIT Press, Cambridge, MA, 1987.

[27] W. Percival. British patent specification, no 460,562. July 1936.

[28] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*, chapter 20.5, pages 736–748. Pearson Education, Upper Saddle River, NJ, second edition, 2003.

[29] M. Seidel and T. Knight. A passive switched capacitor network for solving systems of linear equations. *European Conference on Circuit Theory and Design*, August 1995.

[30] Q. Song and R. Leland. An optimal control model of neural network for constraint optimization problems. *Optimal Control Applications and Methods*, 19:371–376, 1998.

[31] M. Vidyasgar. *Nonlinear Systems Analysis*, chapter 6.2.2, pages 350–358. Society for Industrial and Applied Mathematics, Philadelphia, 2002.

[32] P. Winston. *Artificial Intelligence*, chapter 22-24, pages 443–504. Addison Wesley, Reading, MA, third edition, 1992.

[33] J. Wyatt, L. Chua, J. Gannett, I. Goknar, and D. Green. Energy concepts in the state-space theory of nonlinear n-ports: Part I – passivity. *IEEE Transactions on Circuits and Systems*, 28(1):48–61, January 1981.