



Constraint Modules: An Introduction

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation	Mitros, Piotr. "Constraint Modules: An Introduction." 2008 IEEE International Symposium on Circuits and Systems. Seattle, WA, USA, 2008. 117-120. © Copyright 2011 IEEE
As Published	http://dx.doi.org/10.1109/ISCAS.2008.4541368
Publisher	Institute of Electrical and Electronics Engineers
Version	Final published version
Accessed	Tue Aug 12 09:45:58 EDT 2014
Citable Link	http://hdl.handle.net/1721.1/67460
Terms of Use	Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.
Detailed Terms	

Constraint Modules: An Introduction

Piotr Mitros

MIT Dept. of Electrical Engineering and Computer Science
Cambridge, MA 02139
Email: pmitros@mit.edu

Abstract—We describe a methodology for solving convex constraint problems using analog circuits. We demonstrate how this methodology can be used to design circuits that solve function-fitting problems through iterated gradient descent. In particular, we show 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 primarily local properties of the circuit to show stability. Such stability arguments can be scaled to much more complex systems than traditional stability criteria.

I. OVERVIEW

We present 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 is applied to designing circuits that solve several classes of problems. In particular, it is applied to designing a compact circuit that can build a model of a memoryless nonlinear system, and predistort to linearize this system.

The methodology is based on using active transformers with the traditional technique for solving systems of linear equations with transformers [3] [5]. With active rather than passive transformers, stability can no longer be guaranteed by passivity. If the implementation of an active transformer deviates from the ideal model, the resulting circuit may no longer be passive, and potentially no longer be stable. As a result, a new stability criterion is necessary. The stability criterion developed is equivalent to that in Chua's canonical nonlinear programming circuit [1] [2].

This work differs from Chua's in two significant ways. First, rather than just using the resulting systems for the solution of static mathematical programs, we exploit the dynamic properties of these modules to build interesting time-varying systems. Second, Chua's circuits were monolithic. In contrast, we treat constraint modules as components useful for the construction of complex circuits. Individual constraint modules could be made into discrete components, or used as cells in a standard cell layout. These, in turn, could be used for automated synthesis of complex circuits.

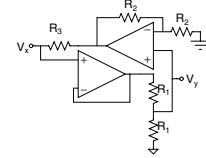


Fig. 1. An active transformer holding the constraint $V_x = 2V_y$.

II. STATIC SYSTEMS

We assign each constraint module an objective function that is minimized when the constraint is met. For instance, a 2:1 transformer holding the constraint $V_x = 2V_y$ could be assigned the least-squares objective function $L(V_x, V_y) = (V_x - 2V_y)^2$. Currents are output onto adjacent nodes in proportion to how much a change in that node's voltage affects the objective function. In this case, the output currents would be $I_x = -\frac{\partial L}{\partial V_x} = -2V_x + 4V_y$ and $I_y = -\frac{\partial L}{\partial V_y} = 4V_x - 8V_y$. A circuit implementation of this is shown in Fig. 1. Since $I_y = -2I_x$, this is consistent with the currents expected from a 2:1 transformer. Indeed, this methodology is closely related to incremental passivity. All incrementally passive memoryless one-ports (in circuit design terminology, static 2-terminal elements not capable of power gain) are valid constraint modules. It is, however, possible to design valid constraint modules that are neither passive nor incrementally passive. In contrast to passive circuits, constraint modules connected to inductors can oscillate.

Stability follows by a Lyapunov-type argument. Let the global objective function be the sum of the objective functions of all of the constraint modules in the circuit. Assume each node i has capacitance C_i . Let N_i be the set of all constraint modules around the i^{th} node. Then, the change in voltage on the i^{th} node is:

$$\frac{dV_i}{dt} = -\frac{1}{C_i} \sum_{j \in N_i} \frac{\partial L_j}{\partial V_i}$$

The effect of the change in that voltage on the overall objective function is just the sum of the effects of that voltage on the individual objective functions:

$$\frac{\partial L_{N_i}}{\partial V_i} = \sum_{j \in N_i} \frac{\partial L_j}{\partial V_i}$$

The change in the sum of those objective functions over time,

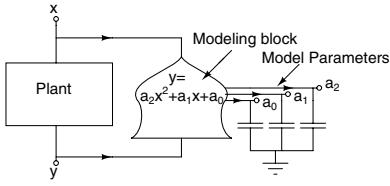


Fig. 2. Using a constraint module to obtain the parameters of a quadratic model for the plant.

as a result of the change in the voltage on that node is:

$$\frac{dL_{N_i}}{dt} = \left(\frac{dV_i}{dt} \right) \left(\frac{\partial L_{N_i}}{\partial V_i} \right) = -\frac{1}{C_i} \left(\sum_{j \in N_i} \frac{\partial L_j}{\partial V_i} \right)^2 \leq 0 \quad (1)$$

The global objective function is just the sum of all of the local objective functions. The global objective function is, therefore, decreasing. By LaSalle's Theorem, the system will converge to a minimum of the overall objective function, and remain within that minimum.

Since low-frequency stability does not depend on the capacitances C_i , high frequency gain may be made arbitrarily small by increasing C_i without affecting low-frequency stability. High-frequency stability may then be shown by an argument analogous to the small gain theorem. The necessary size of the capacitances C_i is a local property — it may be conservatively determined based solely on the properties of the constraint modules adjacent to node i .

Robust stability (where the circuit implementation does not match the model perfectly) may be shown by finding a region B' where Eq. 1 no longer holds (usually, a small plateau around the minima), and determining the supremum value of the global objective function L within that region: $v = \sup_{\mathbf{x} \in B'} L(\mathbf{x})$. It can also be shown that the system will stay within the region $B = \{\mathbf{x} | L(\mathbf{x}) < v\}$. If B is sufficiently small such that the system is essentially linear within it, global stability follows by scale invariance of linear systems.

III. DYNAMIC SYSTEMS

Take a constraint module implementing the constraint $y = a_2x^2 + a_1x + a_0$ with the objective function $L(\mathbf{a}, \mathbf{x}, y) = (a_2x^2 + a_1x + a_0 - y)^2$ shown in Fig. 2. Assume that voltages are only connected to ports x and y , while the a_i nodes are left floating (resulting in an unconstrained system). The stability criterion from section II guarantees convergence to the hyperplane that solves the equation, but the system may drift within that hyperplane in a potentially unstable way.

Now, assume the system is shown three possible input values: (x_0, y_0) with duty cycle $\tau = \frac{1}{2}$, (x_1, y_1) with duty cycle $\tau = \frac{1}{4}$, and finally, (x_2, y_2) with duty cycle $\tau = \frac{1}{4}$. This system is equivalent to three identical systems, each with the same objective function as the original, connected permanently to (x_0, y_0) , (x_1, y_1) , and (x_2, y_2) , respectively, but being switched to the nodes a_1 , a_2 , and a_3 with the same duty cycles. In the limit case of high switching speed, this, in turn, is equivalent to having three constraint modules with

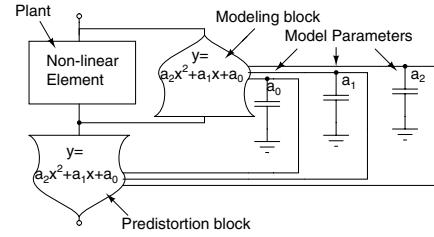


Fig. 3. A simple linearizer consisting of a modeling block and a matched predistortion block.

currents (and therefore objective functions) scaled by the duty cycle permanently connected to the nodes a_1 , a_2 , and a_3 . In other words, it is equivalent to a system with the overall objective function:

$$L(\mathbf{a}, \mathbf{x}, \mathbf{y}) = \begin{aligned} &\frac{1}{2}(a_2x_0^2 + a_1x_0 + a_0 - y_0)^2 + \\ &\frac{1}{4}(a_2x_1^2 + a_1x_1 + a_0 - y_1)^2 + \\ &\frac{1}{4}(a_2x_2^2 + a_1x_2 + a_0 - y_2)^2 \end{aligned}$$

This is, in particular, applicable to the problems of modeling and linearization. In the above case, the circuit will monitor a plant, and generate the best-fit model parameters a_i based on the data points generated by the plant, weighed by the duty cycles of those data points. Given this modeling block, it is straightforward to linearize by driving the input to the plant with a matched constraint module, as shown in Fig. 3.

In general, this technique allows very simple circuits to solve fairly complex constraint problems. If the constraint equations have similar forms, instead of requiring a separate constraint module for each equation, this allows us to use a single module switching rapidly between implementing different constraints. If this is implemented by explicitly serializing parallel samples, the control logic may be complex, but as in the modeling case, it can often be implemented implicitly.

IV. GENERAL LINEAR REGRESSION

One useful class of models is defined by using least squares to minimize the constraint:

$$\sum a_i f(\mathbf{x}) = g(\mathbf{x})$$

This class of models always results in a convex objective function. As a result, the circuit will always converge. This includes a broad range of models, including Taylor expansions, Fourier series, Chebyshev polynomials, discretizations, and wavelet expansions.

For models of this form, if the model space can perfectly match the plant, the fast switching requirement is no longer necessary. Each point \mathbf{x} defines a hyperplane over the space of possible values of \mathbf{a} . This hyperplane passes through the plant's actual model, \mathbf{a}_{sys} . The system projects the current estimate $\hat{\mathbf{a}}$ towards that plane. Therefore, $\hat{\mathbf{a}}$ can only move in the direction of the actual value \mathbf{a}_{sys} , and so converges to \mathbf{a}_{sys} .

If the model space cannot perfectly match the plant, the circuit will still converge to within a ball around the ideal model. The size of this ball is roughly inversely proportional to

a measure of the level of signal diversity, and proportional to a measure of the level to which the model space can approximate the plant. A formalization of the concept of signal diversity and of the level to which the model space can approximate the plant is given in [4].

V. COMPILING

In many applications, circuits developed under this methodology are unnecessarily complex, and can be compiled into a more efficient form. In particular, many constraint modules will buffer the voltages on the nodes around them. Multiple modules can share a common buffer. Modules containing voltage-mode logic will frequently have a voltage-to-current converter on the output. These can be shared. Many modules will compute the same functions of adjoining voltages. Sharing computation among modules can reduce circuit size while improving matching between modules. In addition, there is often a large number of circuit-specific simplifications.

We will briefly present an example of one compiled circuit. Take a system that is trying to linearize a nonlinear plant by characterizing it and predistorting with a model of the form:

$$v_i = \sum_i a_i f_i(v_o)$$

Using a least-squares objective function, this constraint module gives the resulting dynamics:

$$\begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \vdots \\ \dot{c}_n \end{bmatrix} = \begin{bmatrix} f_1(v_o) \\ f_2(v_o) \\ \vdots \\ f_n(v_o) \end{bmatrix} \cdot \left(v_i - \sum_i a_i f_i(v_o) \right) \quad (2)$$

This consists of a direction term $\langle f_1(v_o), \dots, f_n(v_o) \rangle$, and a scaling term $(v_i - \sum_i a_i f_i(v_o))$. If the model is monotonic¹, then $\text{sign}(v_i - \sum_i a_i f_i(v_o)) = \text{sign}(v_d - v_o)$. The argument in section IV does not depend on the rate of descent, so we approximate the ideal system dynamics from Eq. 2 as:

$$\begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \vdots \\ \dot{c}_n \end{bmatrix} = \begin{bmatrix} f_1(v_o) \\ f_2(v_o) \\ \vdots \\ f_n(v_o) \end{bmatrix} \cdot (v_d - v_o) \quad (3)$$

The direction of descent does not change. Since the rate of descent is now dependent on operating point, the argument in section III no longer applies. Some points are weighed more than other points, and the circuit does not necessarily generate least-squares optimal models. The change in weight is approximately equal to the slope of the model at that point (a proof is given in [4]). If the model is close to affine, the effect of the approximation will be minimal. Alternatively, if the model space can fit the system perfectly, the system will still reach the ideal model. If the model space can only model the system poorly, and the model is strongly nonlinear this

¹If the model space contains nonmonotonic models, but the correct model is known to be monotonic, this can be guaranteed by using an additional constraint module to hold the system within the space of monotonic models.

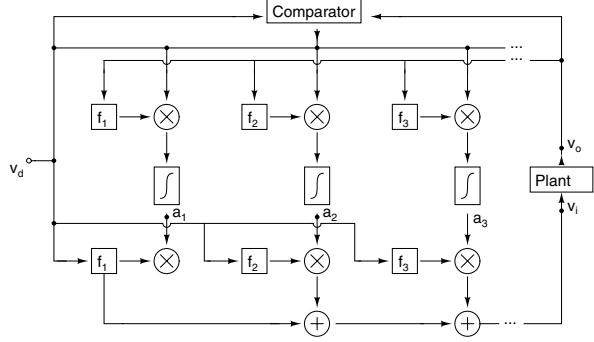


Fig. 4. A block diagram of the linearizer circuit.

approximation will potentially introduce a significant error in the resulting model.

A block diagram of the resulting circuit is shown in Fig. 4. Here, for each term $a_i f_i(v_o)$, the circuit has a structure consisting of a two instances of the basis function, two multipliers, and an integrator. The system relies on the top multiplier having very good behavior only around the origin. Given adequate signal diversity, small errors in large-signal behavior of the multipliers do not have a significant effect on the performance of the system. If the basis functions have significant errors, but the two sets of functions match well, the effect on overall system performance will usually also be minimal. This will change the model space to have a different set of basis functions, but the system will still find a good model within this new model space.

Aside from compactness, the compiled circuit has some significant advantages over the original. The sign of the scaling term is calculated from the overall error in the output, $v_d - v_o$, rather than the error in the modeling portion of the circuit. As a result, the global feedback overcomes many matching issues of the original circuit.

Next, the circuit is robust in the following sense: if any of the columns fails in a static way (in other words, the output is a memoryless function of the input), the remaining columns will simply model the failed column as if it were part of the plant. They will build a combined model of the nonlinearity and the failed column, and compensate for the failure. This is mostly relevant to systems with a very large number of columns. One of the goals of this research is to investigate methodologies that will allow analog circuits to take advantage of increased numbers of transistors in the way that digital circuits do. The local stability criterion, together with the partial analog fault tolerance, are two tools towards accomplishing this goal.

VI. EXPERIMENTAL RESULTS

We built a number of test systems to demonstrate the functionality of the constraint modules. Here, we will only present results for the compiled linearization circuit implementing the model $v_i = a_0 + a_1 v_o + a_2 v_o^2$. While the linearizer circuit requires 5 operational amplifiers and 6 multipliers the test circuit used 10 operational amplifiers so that we could more

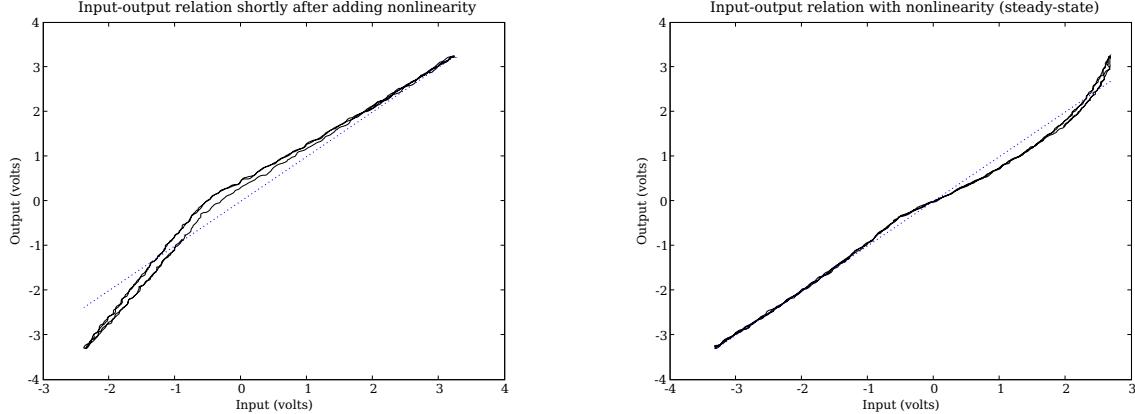


Fig. 6. Input-output relationship of the compiled linearization circuit with a simple diode non-linearity (shown in Fig. 5). The circuit was driven with a 350Hz, 6.5V_{PP} triangle wave. The left graph shows the input-out relationship shortly after the nonlinearity was added, whereas the right graph shows the relationship after the system has converged to steady-state and compensated for the nonlinearity. In the steady-state, distortion is reduced by 65%.

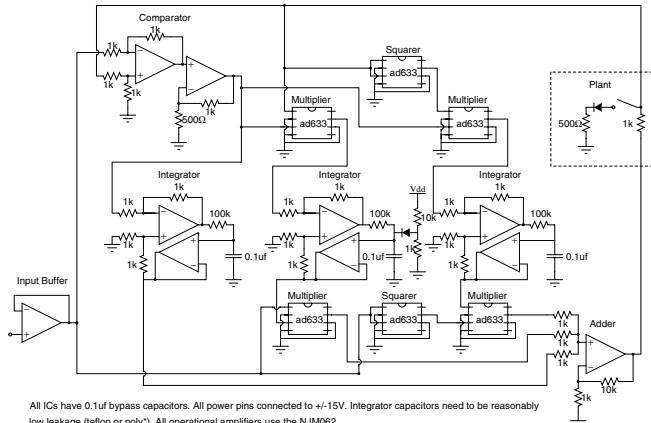


Fig. 5. Detail of the linearizer circuit for the model $v_i = a_1v_d^2 + a_2v_d + a_3$.

easily measure its performance and modify circuit parameters. The linearizer circuit models a voltage divider with a diode. This plant was chosen to show worst-case behavior — here, the model can fit the system poorly, and the resulting model is fairly nonlinear. The circuit detail is shown in Fig. 5. Experiment results are shown in Fig. 6. The modeling and predistortion circuit reduced offset to less than 8mV, gain error to less than 0.3%, and nonlinear distortion by 65%.

VII. APPLICATIONS

We foresee a number of applications for this system. Feedforward linearization can be used inside of a feedback loop without significantly affecting feedback linearization. If individual stages of an operational amplifier were designed in such a way that bias currents could control level of distortion, a modeling stage could characterize the levels of gain and distortion, and the output could be used (possibly through another optimization module) to adjust those bias currents to reduce distortion while maximizing gain. Here, the modeling

stage only needs to be accurate at signal speeds, and not at the full bandwidth of the feedback loop.

Next, this gives a compact way to linearize systems with time delays, as in Cartesian feedback of RF amplifiers. As presented, it does not give a way to control systems with complex dynamics, although there are several possible extensions that may allow this. The static system gives a compact way to solve constraint optimization problems, such as those that occur when trying to calculate optimal bias current levels based on measured functions of temperature and process variations. It may also be applicable to building analog soft coders.

VIII. CONCLUSION

In this paper, we demonstrated a methodology for the design of complex analog systems. This methodology can be directly applied to build models of systems, to predistort and linearize nonlinearities, and to solve some classes of systems of equations. The methodology scales well to the design of complex systems. In many cases, the resulting circuits are compact, and can implement significant computation at substantially lower cost than a digital solution. They can therefore be used as subcomponents for simple analog blocks.

ACKNOWLEDGMENT

I would like to thank Tom Knight and Gerry Sussman for their help, support, advice, and supervision. Without them, this project would have been impossible.

REFERENCES

- [1] L. Chua and L. Gui-Nian. Nonlinear programming without computation. *IEEE Transactions on Circuits and Systems*, 31(2):182-188, February 1984.
- [2] M. Kennedy and L. Chua. Neural networks for nonlinear programming. *IEEE Transactions on Circuits and Systems*, 35(5):554-562, May 1988.
- [3] R. Mallock. An electrical calculating machine. *Proceedings of the Royal Society*, 140:457-483, 1933.
- [4] P. Mitros. *Constraint satisfaction modules: A methodology for analog circuit design*. Ph.D. Thesis. MIT. 2007.
- [5] 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.