Feasible Region Approximation Using Convex Polytopes

Sachin S. Sapatnekar
Department of EE/CprE
Iowa State University
Ames, IA 50011.

Pravin M. Vaidya
Department of Computer Science
University of Illinois
Urbana, IL 61801.

S. M. Kang
Coordinated Science Laboratory
University of Illinois
Urbana, IL 61801.

Abstract

A new technique for polytope approximation of the feasible region for a design is presented. This method is computationally less expensive than the simplicial approximation method [1]. Results on several circuits are presented, and it is shown that the quality of the polytope approximation is substantially better than an ellipsoidal approximation.

I. Introduction

While designing a circuit according to certain specifications, it is useful to define a *feasible region* as the set of parameter values for which the design specifications are satisfied. It is a commonly-made assumption [1, 2] that the feasible region is a convex body; while in general, this is not true, it is usually true that the feasible region is "nearly convex."

An example of a situation in which the concept of the feasible region is useful is the problem of design centering. Realising that process variations may cause parameter values to drift from their nominal values, this problem seeks to select the nominal values of design parameters in such a way that the behavior of the circuit remains within specifications with the greatest probability. In other words, the aim of design centering is to ensure that the manufacturing yield is maximized. An approach that is commonly used to solve this problem [1, 2] approximates the feasible region by a convex body, and takes the design center to be its exact or estimated center.

The procedure in this paper can be used for the approximation of *any* convex (or "nearly convex") body.

Two types of convex bodies have been used most commonly for feasible region approximation:
(i) An *ellipsoid*, which is a body defined by

$$\{\mathbf{y} \mid (\mathbf{y}-\mathbf{y}_c)^T \mathcal{B}(\mathbf{y}-\mathbf{y}_c) \le \mathrm{r}^2\}, \mathcal{B} \in \mathbf{R}^{n\times n}, \mathbf{y}, \mathbf{y}_c \in \mathbf{R}^n. \quad (1)$$

Ellipsoids are inherently limited in approximating asymmetric regions, and regions with linear edges.
(ii) A *polytope*, which is defined as an intersection of half-spaces, is given by

$$\mathcal{P} = \{\mathbf{x} \mid A\mathbf{x} \ge \mathbf{b}\}, A \in \mathbf{R}^{m\times n}, \mathbf{b} \in \mathbf{R}^m. \quad (2)$$

A polytope can be thought of as the convex hull of its vertices. Hence, it is capable of representing any convex body in the limiting case where the number of vertices (and hence, hyperplanes) is infinite. A satisfactory approximation to the convex body can be obtained using a finite number of hyperplanes.

In this work, we address the problem of feasible region approximation by convex polytopes. We present a new approach that is computationally cheaper than the existing approach in [1], and illustrate how polytope approximation is superior to ellipsoidal approximation.

II. Approximation of the Feasible Region

A. Introduction

The feasible region $\mathcal{F} \subset \mathbf{R}^n$, where n is the number of design parameters, is formally defined as the set of points in the design parameter space for which the circuit satisfies all specifications on its behavior. It is often assumed that $\mathcal{F}$ is a convex body. In this work, we preserve that assumption in defining the approach to be taken, and use properties of convex sets to create an approximation to $\mathcal{F}$. In our implementation though, we show how we may make allowances for the incorrectness of this assumption.

The simplicial approximation method [1] provides a method for approximating a feasible region by a polytope. Each iteration of this method involves the solution of $f+1$ linear programs (where f is the number of faces of the polytope), a line search, and updating a convex hull. Apart from the fact that each linear program solution is expensive, the number of linear programs to be solved in the entire procedure is large (specifically, $k(f+1)$, where k is the number of iterations). Moreover, the procedure of updating the convex hull is also computationally expensive. This is because updating a convex hull consists of identifying hyperplanes to be removed and finding the equations of the new hyperplanes. Obtaining the equation of a hyperplane, given the vertices, is an $O(n^3)$ operation; this needs to be carried out for each new hyperplane of the polytope while updating the convex hull in an iteration.

In the algorithm presented here, an initial polytope is updated by performing a set of line searches. After up to $2n$ line searches, a new polytope center has to be computed, which is an $O(n^{2.5})$ operation. Thus, the algorithm that we present is computationally less complex than simplicial approximation.

B. Construction of the Approximating Polytope

Our algorithm is based on the following well-known theorem on convex sets [3] that is stated below.

Supporting Hyperplane Theorem : Let C be a convex set and let $\mathbf{y}$ be a boundary point of C. Then there exists a hyperplane containing $\mathbf{y}$ and containing C in one of its closed half-spaces.

Definition : A tangent plane at a point $\mathbf{z}_0$ on the boundary of a region is given by

$$[\nabla g(\mathbf{z}_0)]^T \mathbf{z} = [\nabla g(\mathbf{z}_0)]^T \mathbf{z}_0 \tag{3}$$

where $g(\mathbf{z}) \geq 0$ is an active constraint at point $\mathbf{z}_0$, and $\nabla g(.)$ is the gradient of the function $g(.)$.

For a point $\mathbf{z}_0$ on the boundary of a convex set, C, a tangent plane at $\mathbf{z}_0$ satisfies the supporting hyperplane property. In particular, for the constraint $g(\mathbf{z}) \geq 0$, C is contained in the half-space

$$[\nabla g(\mathbf{z}_0)]^T \mathbf{z} \geq [\nabla g(\mathbf{z}_0)]^T \mathbf{z}_0. \tag{4}$$

The aim of the algorithm is to approximate the feasible region, $\mathcal{F} \subset \mathbf{R}^n$ by a polytope

$$\mathcal{P} = \{\mathbf{z} \mid A\mathbf{z} \geq \mathbf{b}\}, A \in \mathbf{R}^{m \times n}, \mathbf{b} \in \mathbf{R}^m, \tag{5}$$

formed by the intersection of m half-spaces in $\mathbf{R}^n$.

The algorithm begins with an initial feasible point, $\mathbf{z}_0 \in \mathcal{F} \subset \mathbf{R}^n$. An n-dimensional box, $\{\mathbf{z} \in \mathbf{R}^n \mid z_{min} \leq z_i \leq z_{max}\}$, containing $\mathcal{F}$ is chosen as the initial polytope $\mathcal{P}_0$. In each iteration, n orthogonal search directions, $\mathbf{d}_1, \mathbf{d}_2 \cdots \mathbf{d}_n$ are chosen. A binary search is conducted from $\mathbf{z}_0$ to identify a boundary point $\mathbf{z}_{b,i}$ of $\mathcal{F}$, for each direction $\mathbf{d}_i$. If $\mathbf{z}_{b,i}$ is relatively deep in the interior of $\mathcal{P}$, then the tangent plane to $\mathcal{F}$ at $\mathbf{z}_{b,i}$ is added to the set of constraining hyperplanes in Equation (5). In practice, since the feasible region is not strictly convex, it is useful to shift the tangent plane away by a distance factor δ. In other words, if $\mathbf{c}$ is the gradient of the active constraint at the boundary point, the hyperplane added is

$$\mathbf{c}^T \mathbf{z} \geq \mathbf{c}^T \mathbf{z}_{b,i}(1-\delta) \tag{6}$$

where δ is typically of the order of 0.01.

A similar procedure is carried out along the direction $-\mathbf{d}_i$. Once all of the hyperplanes have been generated, the center of the new polytope is calculated, using a method described in [4]. Then $\mathbf{z}_0$ is reset to be this center, and the above process is repeated.

The pseudo-code describing the algorithm is as follows:

```
k = 0
Initialize $\mathcal{P}_0$ and $\mathbf{z}_0$
While (# planes added in the last iteration $\neq$ 0) {
    $\mathcal{P}_{k+1} = \mathcal{P}_k$
    Choose orthogonal directions $\mathbf{d}_1, \mathbf{d}_2 \cdots \mathbf{d}_n$
    For i = 1 to n {
        Perform a binary search along direction $\mathbf{d}_i$
            from $\mathbf{z}_0$ to find a boundary point $\mathbf{z}_{b,i}$ of $\mathcal{F}$
        If $\mathbf{z}_{b,i}$ exists, and $\mathbf{z}_{b,i} \in \mathcal{P}_{k+1}$, and
            distance($\mathbf{z}_{b,i}$,boundary of $\mathcal{P}_{k+1}$) > $\epsilon$ {
            Add a hyperplane to the polytope
            $\mathcal{P}_{k+1} = \mathcal{P}_{k+1} \cup$ new hyperplane
        }
        Perform a binary search along direction $-\mathbf{d}_i$
            from $\mathbf{z}_0$ to find a boundary point $\mathbf{z}_{b,i}$ of $\mathcal{F}$
        If $\mathbf{z}_{b,i}$ exists, and $\mathbf{z}_{b,i} \in \mathcal{P}_{k+1}$, and
            distance($\mathbf{z}_{b,i}$,boundary of $\mathcal{P}_{k+1}$) > $\epsilon$ {
            Add a hyperplane to the polytope
            $\mathcal{P}_{k+1} = \mathcal{P}_{k+1} \cup$ new hyperplane
        }
    }
    Set $\mathbf{z}_0$ = center of the updated polytope, $\mathcal{P}_{k+1}$
    k = k + 1;
}
```

The procedure requires a technique for calculating the gradient of an active constraint at a boundary point. As in [2], one of several methods may be employed for this purpose. If the exact functional form of the constraint is known, an expression for the gradient may be derived. If not, methods such as the adjoint network technique [5] or finite differences may be used to calculate the gradient.

III. Experimental Results

A. A Numerical Example

A numerical example in two dimensions is presented here to illustrate our technique. The feasible region is represented by the ellipse

$$\mathcal{F} = \left\{ \mathbf{z} \mid (\mathbf{z}-\mathbf{t})^T \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} (\mathbf{z}-\mathbf{t}) \leq 1 \right\} \tag{7}$$

with the ellipse center, $\mathbf{t}$, at $[4,4]^T$. The approximating polytope, $\mathcal{P}$, shown in Fig. 1, can be seen to be a good approximation of the ellipse.

B. A Tunable Active Filter

The techniques described in Section II were used to approximate the feasible region for a tunable active filter shown in Fig. 2. The one pole role-off model used for the operational amplifier presumes a dc gain of 2×10^5, and a 3-dB bandwidth of 12π rad. The designable parameters are considered to be $[G_1, G_4, C_1, C_2]$. The transfer function for this filter is given by

$$\mathbf{F} = \left| \frac{v_2}{v_g} \right|. \tag{8}$$

An initial feasible point [80.2 $\mu\mho$, 5.42 m$\mho$, 0.72856 μF, 0.72856 μF] is provided, as in [2].

Fig. 1 : Approximation of an ellipsoid by a polytope.

Fig. 2 : A tunable active filter.

The specifications for the filter are as follows:

$$\begin{array}{rclllll} & & \mathbf{F} & \leq & 0.5 & \text{at} & 90 \text{ Hz} \\ & & \mathbf{F} & \geq & 0.5 & \text{at} & 92 \text{ Hz} \\ 1.0 & \leq & \mathbf{F} & \leq & 1.21 & \text{at} & 100 \text{ Hz} \\ & & \mathbf{F} & \geq & 0.5 & \text{at} & 108 \text{ Hz} \\ & & \mathbf{F} & \leq & 0.5 & \text{at} & 110 \text{ Hz} \end{array} \qquad (9)$$

Yield estimation for various values of parameter variances about the point [83.13 $\mu\mho$, 5.423 m$\mho$, 0.716047 μF, 0.769181 μF] was carried out. The results in [2] for the same circuit, for yield estimation about the same point, show that the percentage yield using the approximating ellipsoid as the feasible region is as much as 12 points away from the actual yield.

For various sets of parameter variances (expressed as a percentage of the nominal values above), Table 1 shows the yield using the approximating polytope as the feasible region in comparison with the actual circuit yield. The yield is calculated on the basis on a Monte Carlo simulation of 500 points. It can be seen that the results of our technique are within about 3 percentage points of the actual yield, which is a marked improvement over [2].

Table 1 : Yield for a Tunable Active Filter

Parameter Variances	Actual Yield	Polytope Yield
[1.5, 1.5, 1.5, 1.5]	46.6 %	43.4 %
[1.5, 1.5, 2.0, 0.5]	47.2 %	45.2 %
[1.5, 1.8, 1.2, 0.6]	52.8 %	51.2 %
[1.5, 1.5, 1.5, 0.5]	53.6 %	53.4 %
[1.0, 1.2, 1.0, 0.6]	69.2 %	68.2 %
[0.5, 0.8, 1.0, 0.8]	75.0 %	74.0 %
[1.0, 0.4, 1.0, 0.6]	84.2 %	84.4 %
[0.5, 0.8, 0.7, 0.6]	86.4 %	86.0 %

Fig. 3 (a) : A high-pass filter.
Fig. 3 (b) : Band diagram for the high pass filter.

C. A High Pass Filter

This is an example of a high-pass filter [2], whose circuit diagram and specifications are shown in Fig. 3. The frequencies of interest are $\{170, 350, 440, 630, 680, 990, 1800\}$ Hz, which correspond to seven constraints. For this example, the reference frequency, ω_0, is 990 Hz.

The parameter $\bar{I}_{loss}$ is defined as follows:

$$I_{loss}(j\omega) \;\triangleq\; 20\, log_{10} \left| \frac{V_1}{V_2}(j\omega) \right| \quad \text{dB} \qquad (10)$$

$$\bar{I}_{loss}(j\omega) \;\triangleq\; I_{loss}(j\omega) - I_{loss}(j\omega_0) \quad \text{dB}. \qquad (11)$$

The first experiment took the design parameters as $[C_1, C_3, C_4, C_5]$. The initial feasible point was taken to be $[11.1\text{nF}, 12.9\text{nF}, 34.3\text{nF}, 97.3\text{nF}]$, as in [2]. Table 2 shows the results of yield estimation about the point $[10.37\text{nF}, 13.28\text{nF}, 34.63\text{nF}, 87.84\text{nF}]$. The yield figures for the approximating polytope are seen to be close to the actual yield, in contrast with [2], where they are off by as many as 26 percentage points.

$[C_1, C_2, C_3, C_4, C_5, L_1, L_2]$ were used as design parameters for the second experiment. The initial feasible point was taken, as in [2], as [11.65 nF,10.47 nF,13.99 nF,39.93 nF,99.4 nF,3.988 H,2.685 H]. The yield figures for this more complex circuit are also almost always within a few percentage points of the actual yield, as against [2], where the discrepancy is as much as 34 percentage points.

The interpretations of the columns of Tables 2 and 3 are the same as those in Table 1.

IV. Conclusion

In this paper, a new approach to feasible region approximation has been presented, using a procedure that is computationally less expensive than the existing method.

Experimental results show that in contrast to the results of ellipsoidal approximation in [2], this technique provides a good approximation to the feasible region for the *same* example circuits, with yield estimates taken about the *same* points.

The three examples in [2] are all simulated here, with the following results :

a. For the tunable active filter, our results were within about 3 percentage points of the actual yield, as against a discrepancy of upto 12 percentage points in [2].

b. For the high pass filter, with four parameters, our technique calculates the yield within an accuracy of 5 percentage points, whereas [2] shows errors of up to 26 percentage points.

c. For the high pass filter, with seven design parameters, the yield estimates are accurate to within 4 percentage points, as against 34 points in [2].

Thus, we have illustrated that polytope approximation is better than ellipsoidal approximation since the highly symmetric ellipsoidal shape is inherently incapable of approximating less symmetric bodies. Also, as shown in Section II, our method is computationally less expensive than existing approaches.

Table 2 : Yield for a High Pass Filter (4 Parameters)

Parameter Variances	Actual Yield	Polytope Yield
[15, 15, 15, 15]	55.4 %	54.8 %
[15, 18, 18, 12]	57.6 %	54.4 %
[15, 10, 15, 10]	65.0 %	64.8 %
[10, 15, 10, 15]	71.8 %	66.4 %
[8, 12, 10, 10]	79.8 %	79.0 %
[5, 15, 10, 5]	80.2 %	78.8 %
[10, 10, 10, 10]	81.4 %	81.6 %
[9, 9, 9, 9]	86.4 %	88.4 %
[10, 5, 5, 15]	92.6 %	89.8 %

Table 3 : Yield for a High Pass Filter (7 Parameters)

Parameter Variances	Actual Yield	Polytope Yield
[10, 10, 10, 10, 10, 10, 10]	29.8 %	32.4 %
[5, 10, 5, 10, 5, 10, 5]	41.0 %	44.2 %
[8, 8, 8, 8, 8, 8, 8]	44.0 %	48.2 %
[5, 5, 5, 5, 10, 10, 10]	45.2 %	47.8 %
[10, 10, 10, 10, 5, 5, 5]	48.8 %	50.0 %
[9, 10, 8, 10, 5, 4, 6]	50.6 %	52.6 %
[8, 8, 8, 8, 5, 5, 5]	60.6 %	62.4 %
[4, 4, 8, 8, 8, 4, 4]	71.8 %	74.2 %
[5, 5, 5, 5, 5, 5, 5]	72.6 %	76.6 %
[4, 4, 4, 5, 3, 4, 4]	83.2 %	84.4 %
[2, 2, 2, 5, 2, 4, 4]	86.4 %	86.6 %
[4, 2, 4, 5, 3, 2, 4]	94.1 %	92.8 %

References

[1] S. W. Director and G. D. Hachtel, "The simplicial approximation approach to design centering," *IEEE Trans. Circuits and Systems*, vol. CAS-24, no. 7, pp. 363–372, 1977.

[2] H. L. Abdel-Malek and A.-K. S. O. Hassan, "The ellipsoidal technique for design centering and region approximation," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 1006–1014, Aug. 1991.

[3] D. G. Luenberger, *Linear and Nonlinear Programming.* Addison-Wesley, 1984.

[4] S. S. Sapatnekar, P. M. Vaidya, and V. B. Rao, "A convex programming approach to transistor sizing for CMOS circuits," *Proc. Int. Conf. on Computer-Aided Design*, pp. 482–485, Nov. 1991.

[5] L. O. Chua and P. M. Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques.* Prentice-Hall, 1975.