**BÁO CÁO KẾT QUẢ NGHIÊN CỨU BÀI BÁO PHƯƠNG PHÁP THIẾT KẾ THỰC DỤNG VÀ HIỆU QUẢ CHO CÁC BỘ KHUẾCH ĐẠI CÔNG SUẤT (PA) RF HOẠT ĐỘNG TRÊN BĂNG THÔNG CỰC RỘNG VỚI HIỆU SUẤT CAO**

**I. NHỮNG ĐIỂM CHÍNH TRONG BÀI BÁO**

Điểm chính của bài báo: Cốt lõi của bài báo nằm ở thuật toán tổng hợp mạng phối hợp. Thay vì dựa vào các phương pháp tối ưu hóa phức tạp đòi hỏi điều kiện ban đầu chính xác, phương pháp được đề xuất sử dụng một thuật toán tìm kiếm lặp trực tiếp. Phương pháp này bắt đầu bằng việc xác định các vùng trở kháng tối ưu phụ thuộc vào tần số, thay vì các điểm trở kháng tối ưu cố định, sau đó tổng hợp một mạng dạng hình thang phân tán để đạt được các vùng trở kháng này.

**1. Tổng quan**

**\* Thách thức trong bộ công suất hiện đại**

Thiết kế bộ khuếch đại công suất RF là một bài toán tối ưu hóa đa mục tiêu, trong đó các chỉ số hiệu năng chính (KPIs) như băng thông hoạt động, hiệu suất cộng thêm công suất (PAE), công suất đầu ra và độ tuyến tính thường có mối liên hệ chặt chẽ với nhau.

Ví dụ, để đạt được hiệu suất rất cao, các kỹ thuật như điều chỉnh hài (harmonic tuning) thường được sử dụng trong các lớp khuếch đại như Class-F hoặc Class-E. Các kỹ thuật này định hình dạng sóng điện áp và dòng điện tại cực máng của tranzistor để giảm thiểu sự chồng lấn, qua đó giảm tổn hao công suất. Tuy nhiên, việc điều khiển chính xác các thành phần hài (thường là làm cho chúng hở mạch hoặc ngắn mạch) về bản chất là một kỹ thuật băng hẹp. Ngược lại, để đạt được băng thông rộng, các mạng phối hợp trở kháng thường phải chấp nhận sự không hoàn hảo trong việc phối hợp tại bất kỳ tần số đơn lẻ nào, dẫn đến suy giảm hiệu suất và công suất đầu ra.

**\* Hạn chế chính của các phương pháp thiết kế PA băng rộng truyền thống:**

* **Phối hợp Trở kháng Cố định:** Phương pháp thông thường là xác định một trở kháng tải tối ưu duy nhất (thường tại tần số trung tâm của dải băng) và sau đó thiết kế một mạng phối hợp kiểu bộ lọc để duy trì trở kháng này trên một băng thông rộng. Kỹ thuật này có thể hoạt động hiệu quả đối với các băng thông vừa phải, nơi trở kháng tối ưu của tranzistor không thay đổi nhiều. Tuy nhiên, đối với các ứng dụng đa quãng tám (multi-octave), trở kháng tối ưu của tranzistor có sự phân tán đáng kể theo tần số, như được minh họa trong Hình

1 của bài báo. Việc cố gắng phối hợp một trở kháng cố định sẽ dẫn đến sự suy giảm hiệu suất nghiêm trọng ở các cạnh của dải băng.

* **Sự phụ thuộc vào các Công cụ Tối ưu hóa CAD:** Các công cụ thiết kế có sự hỗ trợ của máy tính (CAD) là không thể thiếu, nhưng nhiều trình tối ưu hóa dựa trên gradient của chúng có một điểm yếu cố hữu: chúng yêu cầu một bộ giá trị phần tử ban đầu hợp lý để bắt đầu quá trình tìm kiếm. Nếu không có điểm khởi đầu tốt, các thuật toán này rất dễ bị mắc kẹt trong các cực tiểu cục bộ và không thể tìm ra giải pháp toàn cục thỏa đáng. Điều này đặt gánh nặng lên kinh nghiệm của nhà thiết kế và làm cho quá trình thiết kế kém hiệu quả và không đáng tin cậy.

A diagram of a frequency increase

AI-generated content may be incorrect.

**\* Phương pháp tiếp cận của bài báo:**

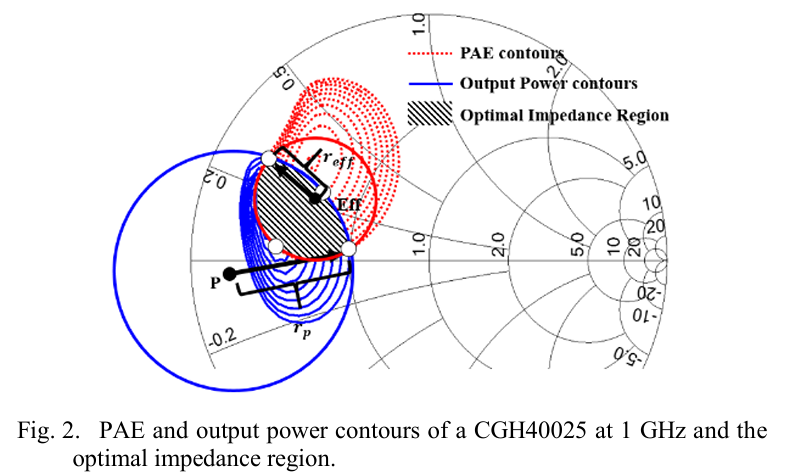
Thay vì theo đuổi một điểm "tối ưu" lý thuyết duy nhất, phương pháp này hướng tới việc tìm kiếm một giải pháp "đủ tốt" trên toàn bộ dải thông. Mục tiêu không phải là đạt được một điểm cụ thể, mà là đảm bảo trở kháng của mạng phối hợp nằm bên trong một vùng tối ưu – là vùng thỏa mãn các chỉ tiêu đầu ra về hiệu suất, công suất đầu ra tại mỗi tần số. Điều này thừa nhận rằng việc đạt được một sự phối hợp hoàn hảo ở tất cả các tần số là rất khó. Thay vì tối ưu hóa cho một điểm duy nhất và khó có thể tìm kiếm, việc đảm bảo hiệu suất không bao giờ giảm xuống dưới một ngưỡng cao là một lựa chọn thông minh và dễ thực hiện hơn.

**2. Phân tích phương pháp**

Có thể phân tích thành 2 bước chính như sau:

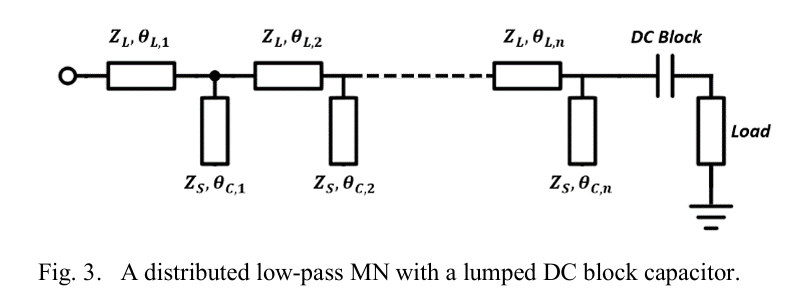
***Bước 1: Xác định vùng trở kháng mục tiêu***

Tiến hành chạy mô phỏng load-pull để xác định các đường bao (contours) trên biểu đồ Smith, nơi các chỉ số hiệu năng quan trọng như PAE và công suất đầu ra (Pout) đạt được các giá trị trong vùng thỏa mãn – là *vùng chung* hoặc phần giao nhau của các đường bao hiệu suất cao này. Cách tiếp cận này xây dựng một sự đánh đổi hiệu suất một cách tường minh ngay từ trong mục tiêu thiết kế. Toàn bộ dải băng hoạt động được chia thành nhiều phần tần số, và một vùng trở kháng mục tiêu được xác định cho mỗi phần. Điều này cho phép thiết kế thích ứng với sự phân tán đáng kể của trở kháng tối ưu của tranzistor theo tần số.



***Bước 2: Xây dựng một mạng phối hợp***

- Cấu trúc mạng: Bài báo chọn một cấu trúc mạng thang thông thấp phân tán, bao gồm các đường truyền mắc nối tiếp (trở kháng cao, hoạt động như cuộn cảm) và các đoạn dây chập hở mạch mắc song song (trở kháng thấp, hoạt động như tụ điện). Cấu trúc này rất phù hợp cho các ứng dụng băng rộng, dễ dàng chế tạo bằng công nghệ vi dải và có đặc tính lọc thông thấp vốn có, giúp triệt tiêu các hài không mong muốn. Bài báo đã chọn một mạng bậc 4 cho việc triển khai thực tế.



**- Thuật toán tổng hợp: Thuật toán tìm kiếm trực tiếp và lặp được đề xuất.**

1. **Xác định Không gian Tham số:** Các biến cần tối ưu là độ dài điện của các đoạn đường truyền (θL​ cho các phần tử nối tiếp và θC​ cho các phần tử song song). Để đơn giản hóa bài toán, các trở kháng đặc tính của chúng được cố định (ZL​=80Ω, ZC​=25Ω).
2. **Rời rạc hóa và Lấy mẫu:** Không gian liên tục của các độ dài điện (từ 0° đến 180°) được rời rạc hóa thành một số lượng mẫu hữu hạn (m=6 trong bài báo). Điều này biến một không gian tìm kiếm vô hạn thành một không gian hữu hạn, có thể quản lý được.

A line with numbers and letters

AI-generated content may be incorrect.

Đối với bài báo, chọn lượng mẫu m=6, với 8 tham số theta\_L, theta\_C, thì tổng cộng có tổ hợp.

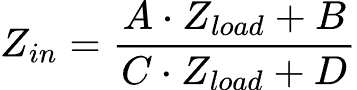
1. **Tìm kiếm Lặp:** Thuật toán lặp lại việc tìm kiếm qua các tổ hợp của các độ dài đã được lấy mẫu. Đối với mỗi tổ hợp (một "tập con"), trở kháng đầu vào của mạng phối hợp được tính toán trên toàn dải băng tần bằng cách sử dụng công thức ma trận ABCD.

A math equations and formulas

AI-generated content may be incorrect.

Ma trận này được cấu thành từ tích (phép nhân ma trận) của các ma trận ABCD của từng thành phần riêng lẻ trong mạch, theo đúng thứ tự chúng được mắc nối tiếp từ đầu vào đến đầu ra.

\begin{cases}
V_{in} = A(Z_t I_{out}) + B I_{out} = (A Z_t + B) I_{out} \\
I_{in} = C(Z_t I_{out}) + D I_{out} = (C Z_t + D) I_{out}
\end{cases}



1. **Hàm chi phí và Lựa chọn:** "Chất lượng" của mỗi tổ hợp được đánh giá bằng một hàm chi phí, đo khoảng cách Euclid ngắn nhất từ điểm trở kháng được tính toán đến ranh giới của vùng trở kháng mục tiêu trên biểu đồ Smith.

A mathematical equation with numbers and symbols

AI-generated content may be incorrect.

1. **Tinh chỉnh:** Tập con có giá trị hàm chi phí (Λ(m)) nhỏ nhất được chọn. Không gian tìm kiếm sau đó được tái định tâm và lấy mẫu lại xung quanh điểm "tốt nhất" mới này, và quá trình được lặp lại cho đến khi quỹ đạo trở kháng nằm trong các vùng mục tiêu trên toàn dải thông.

Hay nói một cách đơn giản, chúng ta sẽ chia lưới độ dài điện và tần số trong dải thông. Ở bước đầu tiên, tìm kiếm thô, chúng ta sẽ tính chi phí cho tất cả các tổ hợp có thể có khi chia lưới độ dài điện, ứng với mỗi tần số, tức là với mỗi bộ tổ hợp và mỗi tần số ta sẽ xác định được giá trị Zin, từ đó ta sẽ xác định được khoảng cách nhỏ nhất từ điểm đó đến đường bao giới hạn vùng trở kháng tối ưu ứng với tần số đó – được xác định khi thực hiện chạy mô phỏng load-pull cho từng tần số. Sau khi khảo sát hết tất cả các tổ hợp, ta sẽ xác định được tổ hợp mà hàm chi phí là nhỏ nhất, hay tổng khoảng cách từ các điểm Zin đến đường giới hạn vùng trở kháng tối ưu ứng với mỗi tần số chia lưới trong dải thông là nhỏ nhất. Kết thúc bước thứ nhất là tìm kiếm thô.

Ở bài báo này, tác giả cũng giả định rằng, đối với các điểm Zin tính được mà nằm trong khoảng vùng trở kháng tối ưu rồi, thì chi phi của điểm đó được cho bằng 0, hay khoảng cách đến đường giới hạn vùng trở kháng tối ưu bằng 0.

Các bước tiếp theo, là tìm kiếm tinh. Lấy tổ hợp nhận được ở bước tìm kiếm thô, ta tiến hành phóng to vùng tổ hợp nhận được, hay lúc này ta sẽ tìm kiếm ở vùng nhỏ hơn với độ chia nhỏ hơn. Ví dụ ở bước thô ta tìm được tổ hợp nhỏ nhất là 15, thì ở bước tiếp theo ta chỉ cần tìm trong khoảng 0-30 chứ không phải là 0-180, nhưng với độ chia lưới vẫn là m=6. Tiếp tục tìm kiếm tổ hợp sao cho hàm chi phí nhỏ nhất, cho đến khi giá trị chi phí thu được thỏa mãn yêu cầu của bài toán. Lý tưởng thì chi phí phải bằng 0.

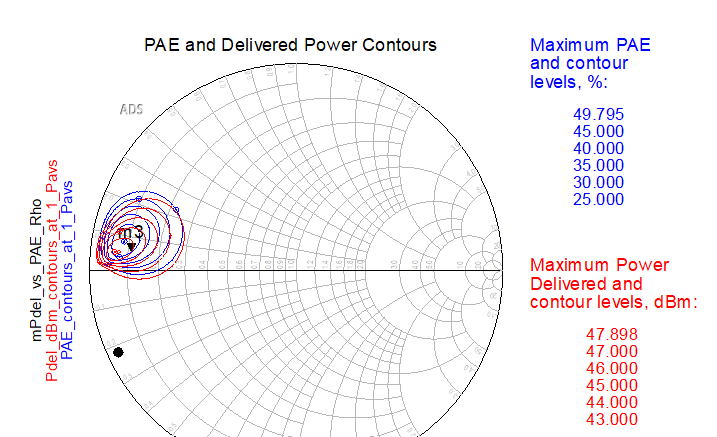
**II. QUÁ TRÌNH TÌM HIỂU**

**1. Xác định vùng trở kháng mục tiêu và tính khoảng cách**

Khi chạy load-pull trong ADS, ta sẽ quan sát được các đường đồng mức của hiệu suất PAE, và công suất đầu ra Pout trên đồ thị Smith, nhưng để dễ dàng tuy chỉnh và xác định vùng trở kháng mục tiêu hiển thị trong ADS là không dễ dàng. Vậy nên, e sẽ trích xuất dữ liệu dữ liệu chứa các tham số này từ ADS ra file .csv để dễ dàng phân tích, xử lý, tìm kiếm vùng trở kháng mục tiêu và vẽ chúng.

Ví dụ dưới đây e xin trình bày bài toán theo phương pháp của tác giả, để đơn giản, thì e chỉ trình bày trước tại 1 tần số, và e lấy ví dụ bài khuếch đại công suất trong bài tập lớn trong kỳ vừa rồi học ạ.

Bước đầu tiên là chạy load-pull sweet, để xác định các đường đồng mức PAE, power theo Pin đầu vào. Nhìn vào đồ thị Smith thu được, ta xác định các đường đồng mức PAE, Power thỏa mãn vùng trở kháng mục tiêu, ví dụ PAE>40%, Power>40dBm. Nhưng nếu để đồ thị như thế nào, thì ta sẽ khó xác định được chính xác đường bao trở kháng mục tiêu, và tìm được khoảng cách nhỏ nhất từ Zin đến đường bao giới hạn đó.



Vì vậy, e sẽ tạo một bảng dữ liệu PAE, Power và xuất nó sang file .csv để dễ dàng thao tác và xử lý dữ liệu.

A screenshot of a computer

AI-generated content may be incorrect. Ảnh trên thể hiện cách xuất dữ liệu PAE, Power từ ADS ra file .csv. Sau khi xuất file .csv ta thu được một file có các cột dữ liệu như sau:

A table with numbers and letters

AI-generated content may be incorrect.

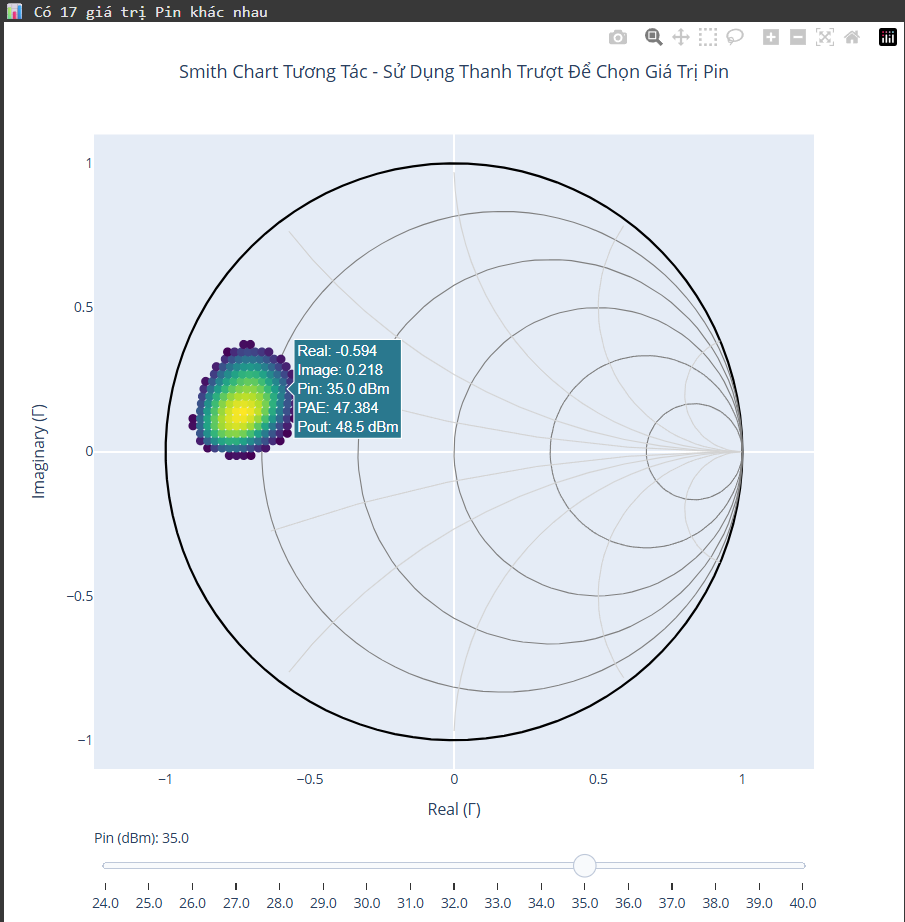
Có 5 cột dữ liệu, trong đó 2 cột đầu là phần ảo, phần thực của hệ số phản xạ gamma, cột thứ 3 là giá trị Pin đầu vào, 2 cột còn lại là hiệu suất PAE và công suất ra Pout.

Khoảng giá trị, số lượng dữ liệu của file .csv phụ thuộc vào các tham số ta cấu hình trong phần tử LoadPull instrument như min max, image, real và độ rộng dải Pin.

A screenshot of a computer

AI-generated content may be incorrect.

Từ dữ liệu này ta có thể viết chương trình xây dựng được vùng trở kháng mục tiêu, tìm đường bao giới hạn của vùng trở kháng mục tiêu đó, tính được khoảng cách nhỏ nhất từ một điểm Zin bất kì đến đường bao giới hạn đó.



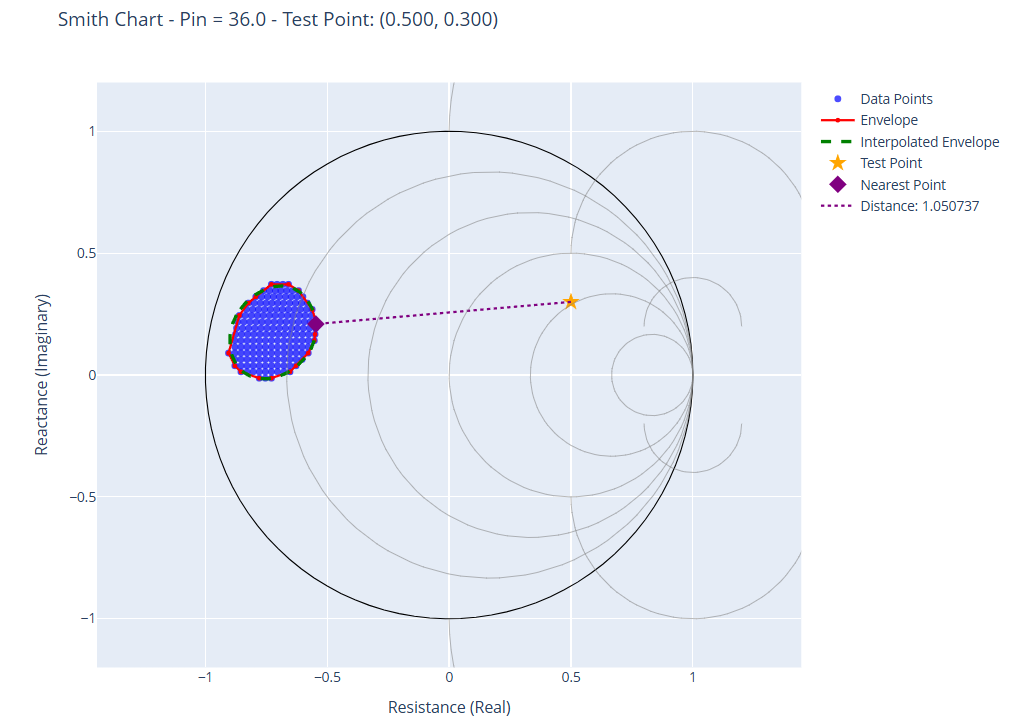
Ảnh trên hiển thi tất cả các điểm dữ liệu sau thỏa mãn các tiêu chí PAE>40%, Pout>40dBm từ file dữ liệu gốc theo giá trị Pin. Thanh trượt Pin, ta có thể dễ dàng tùy chỉnh dễ dàng giá trị Pin đầu vào, ứng với mỗi giá trị Pin đầu vào, ta thu được một tập hợp điểm thỏa mãn các tiêu chí về hiệu suất và công suất đầu ra.

- Sau khi tìm được tập hợp các điểm thỏa mãn, ta tiến hành xác định đường bao giới hạn vùng trở kháng mục tiêu.

+ Sử dụng phương pháp ConvexHull để tạo đường bao đi qua các điểm dữ liệu nằm ngoài cùng.

+ Sử dụng phương pháp Spline để nội suy đường bao được tạo thành ở trên thành một đường cong mượt mà hơn, sát với thực tế hơn.

- Sau khi xác định được đường bao vùng trở kháng mục tiêu, ta tiến hành xây dựng thuật toán tính khoảng cách ngắn nhất từ một điểm test (Zin) bất kì đến đường bào giới hạn. Chú ý rằng, ta phải thêm một bước kiểm tra điều kiện, xem điểm test đó có nằm trong hay ngoài vùng trở kháng mục tiêu, nếu nằm trong thì gán hàm chi phí cho nó bằng 0, nếu nằm ngoài thì sử dụng công thức tính khoảng cách Euclid để xác định giá trị chi phí.

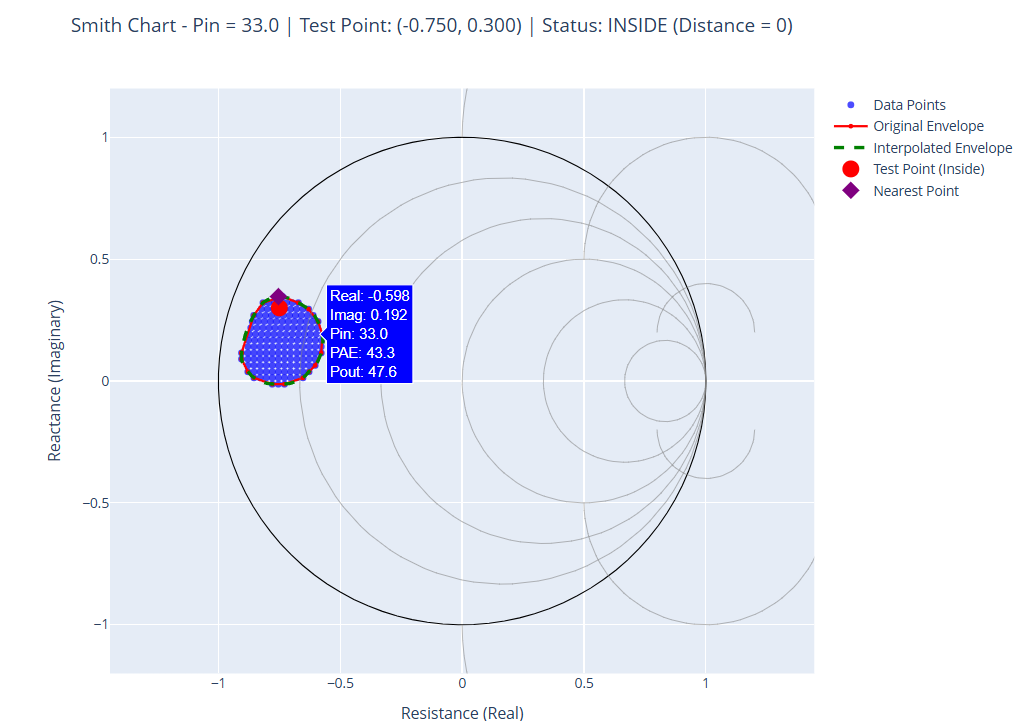


A screenshot of a chart

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Các ảnh trên thể hiện kết quả thu được đường bao giới hạn vùng trở kháng mục tiêu tại Pin=36, và hiển thị 1 ví dụ khoảng cách nhỏ nhất đến đường bao giới hạn tại một điểm test bất kỳ.



Ảnh trên thể hiện khi điểm test nằm trong vùng trở kháng mục tiêu, lúc này thuật toán sẽ gán khoảng cách bằng 0.

**2. Tính trở kháng đầu vào Zin**

Dựa vào công thức ma trận ABCD được xây dựng như trong bài báo, ta xây dựng thuật toán tính giá trị Zin với các tham số đầu vào là các giá trị theta\_L, theta\_C, tần số, trở kháng ZL=80Ohm, ZC=25Ohm (được tác giả chọn từ đầu), gia trị tụ điện =100pF.

Ảnh dưới là hiển thị kết quả thu được sau khi chạy chương trình tính Zin tại 1 bộ tham số cụ thể tại 1 tần số cụ thể.

A white paper with black text

AI-generated content may be incorrect.

**3. Xây dựng thuật toán**

Như phần lý thuyết đã trình bày, thuật toán gồm 2 bước chính, tìm kiếm thô và tìm kiếm tinh.

- Đặt ngưỡng mục tiêu cho hàm chi phí là 1e-06;

- Tần số đầu vào: 2.4GHz;

- Pin=35;

Kết quả thu được khi chạy thuật toán:

A screenshot of a computer

AI-generated content may be incorrect.

**\* Nhận xét:**

- Với bài toán giả sử trên, chỉ tính toán tại 1 tần số nên ngay từ bước đầu tiên, tìm kiếm thô đã tìm được 5420 bộ tham số thỏa mãn chỉ tiêu – tức là nằm trong vùng trở kháng mục tiêu rồi. Thời gian chạy thực hiện ở từng bước là khoảng 310 giây.

- Đối với bài toán thực tế, làm việc với dải thông rộng, lượng tần số làm việc tăng lên, đặt ra nhiều tiêu chí phải thỏa mãn hơn (hàm chi phí phải tối ưu về 0), cũng như khối lượng tính toán, kiểm tra trong từng bước cũng tăng lên đáng kể, điều này dẫn đến sẽ cần nhiều bước thực hiện hơn để tìm được bộ tham số tối ưu, khi số bước thực hiện sẽ tăng lên nhiều thì tổng thời gian tính toán sẽ tăng lên tuyến tính. Đây là một vấn đề đặt ra khi nếu số bước thực hiện tăng lên đáng kể.

- Nếu chạy với số bước thực hiện lớn rồi nhưng vẫn chưa tìm được bộ tham số tối ưu, ta có thể tăng thêm số bậc mạng hình thang lên, hoặc đưa tham số ZL, ZC (có điều kiện ràng buộc chặt chẽ), tăng tỷ lệ chia lưới nhỏ hơn về độ dài điện và tần số để có thể tìm được bộ tham số tối ưu. Nhưng điều này đòi hỏi khối lượng tính toán tăng lên và thời gian chờ tăng lên.

**\* Nhận xét một số điểm hạn chế về phương pháp chia lưới**

- Thuật toán tìm kiếm không hiệu quả, tiêu tốn tài nguyên tính toán lớn nếu tham số và số bước thực hiện lớn: Grid search đơn giản, không tận dụng gradient. Thuật toán này hoạt động bằng cách "vét cạn". Nó chia không gian tham số thành một lưới các điểm rời rạc và tính toán hàm chi phí tại **từng điểm một** trên lưới đó rồi so sánh với nhau để tìm chi phí nhỏ nhất.

Hạn chế (Không tận dụng gradient): Thuật toán này hoàn toàn "mù" về hướng đi. Sau khi tính chi phí tại một điểm, nó không rút ra được thông tin gì về việc điểm kế tiếp nên ở đâu để có chi phí thấp hơn. Nó không biết "độ dốc" (gradient) của bề mặt hàm chi phí. Ngược lại, các thuật toán dựa trên gradient (như Gradient Descent) hoạt động thông minh hơn: tại một điểm, chúng sẽ tính toán hướng dốc nhất và đi một bước theo hướng đó. Điều này giúp chúng hội tụ về điểm tối ưu nhanh hơn rất nhiều, thay vì phải thử mọi điểm một cách vô định.

- Cơ chế thích nghi chưa linh hoạt: Cơ chế "thích ứng" duy nhất của thuật toán này là quy tắc "phóng to" (zoom-in). Sau khi tìm thấy điểm tốt nhất trong một lưới thô, nó sẽ vẽ một "cái hộp" nhỏ hơn xung quanh điểm đó và lặp lại tìm kiếm bên trong cái hộp này. Kích thước của cái hộp và số lượng điểm lấy mẫu bên trong m là cố định và được định nghĩa trước.

Quy tắc này quá cứng nhắc. Giả sử vùng tối ưu thực sự có hình dạng một elip dài và hẹp. Việc "phóng to" bằng một cái hộp vuông sẽ rất không hiệu quả, lãng phí nhiều tính toán vào những khu vực không tiềm năng của cái hộp. Một cơ chế thích ứng thông minh hơn có thể sẽ phân tích sự phân bố của các điểm tốt trong vòng lặp trước để điều chỉnh không chỉ vị trí mà còn cả **hình dạng và kích thước** của vùng tìm kiếm tiếp theo. Các thuật toán như Tối ưu hóa Bayes (**Bayesian Optimization**) làm điều này rất tốt bằng cách xây dựng một mô hình xác suất của hàm mục tiêu và dùng nó để quyết định điểm tiếp theo cần thử nghiệm một cách thông minh.

- Có thể mặc kẹt tại điểm cực tiểu địa phương (local minimum):

Ở vòng tìm kiếm thô đầu tiên, nó tìm ra vùng có vẻ tốt nhất (ví dụ, thung lũng A) và **cam kết** chỉ tìm kiếm trong vùng đó ở các bước tiếp theo. Nó hoàn toàn bỏ qua tất cả các vùng khác.

Điều gì sẽ xảy ra nếu có một thung lũng khác (thung lũng B) sâu hơn nhiều (giải pháp tốt hơn), nhưng ở bước tìm kiếm thô ban đầu, điểm đại diện của nó lại có độ cao chỉ cao hơn một chút so với điểm ở thung lũng A? Thuật toán sẽ loại bỏ thung lũng B ngay từ đầu và không bao giờ có cơ hội tìm ra nó. Nó sẽ bị "mắc kẹt" trong một giải pháp "khá tốt" (cực tiểu cục bộ - local minimum) mà không thể tìm ra giải pháp "tốt nhất" (cực tiểu toàn cục - global minimum).

Các thuật toán tiên tiến hơn như Thuật toán Di truyền (**Genetic Algorithm**) duy trì một "quần thể" gồm nhiều giải pháp đa dạng cùng lúc. Chúng cho phép các giải pháp "lai tạo" với nhau và có những "đột biến" ngẫu nhiên. Chính những đột biến này đôi khi giúp thuật toán "nhảy" ra khỏi một cực tiểu cục bộ và khám phá một vùng không gian hoàn toàn mới, làm tăng đáng kể khả năng tìm ra giải pháp tối ưu toàn cục.

**\* Kết luận:**

- Nếu bài toán có lượng tham số ít và dải tần số không quá rộng, thì sử dụng phương pháp bài báo sẽ đơn giản và hiệu quả.

- Nếu số lượng tham số lớn, dải thông rộng, khối lượng tính toán lớn thì việc sử dụng các mô hình và thuật toán tiên tiến sẽ mang lại hiệu quả hơn.

**III. PHƯƠNG HƯỚNG NGHIÊN CỨU TRONG THỜI GIAN TỚI**

1. Thực hiện bài toán với trường hợp nhiều tần số để đánh giá.

2. Nghiên cứu tìm hiểu các thuật toán tối ưu và tiên tiến hơn như Bayesian, thuật toán di truyền. Từ đó áp dụng vào bài toán, thực hiện chạy, đánh giá, so sánh kết quả với phương pháp ban đầu.

3. Nghiên cứu một số mô hình AI tiên tiến, hiện đại để tăng tốc quá trình đánh giá. Từ đó nghiên cứu cách áp dụng và thực hiện vào bài toán.