

به نام خدا

شماره دانشجویی: ۸۱۰۱۹۸۵۲۹

دانشجو: سید محمد حسینی

پروژه پایانی مدار مخابراتی

استاد درس: دکتر کمره ای

## مقاله انتخابی: Evolving Sinusoidal Oscillators Using Genetic Algorithms

### مقدمه

امروزه کاربرد اسیلاتورها تقریباً فراگیر شده است. از آنجا که فرکانس نوسان و ضریب کیفیت و پهنای باند یک اسیلاتور هنگام طراحی مقادیری محدود است. بنابراین موقعیت هایی پیش می آید که بخواهیم با توجه به مشخص بودن این سه پارامتر، اسیلاتور طراحی کنیم. بنابراین هدف این مقاله طراحی یک اسیلاتور سینوسی تک فرکانسی با فرض اینکه پهنای باند و فرکانس نوسان و ضریب کیفیت مشخص باشند. از آنجا که شماتیک های متفاوتی برای یک اسیلاتور جهت نوسان می توان طراحی کرد ما شماتیک را ثابت گرفته و مدار اسیلاتور زیر (شکل ۱)) را مد نظر قرار می دهیم. با توجه به پیچیده شدن محاسبات و عدم توانایی مشخص کردن سریع المان ها، برای محاسبه مقادیر سلف و خازن و مقاومت از فرمول مستقیم نرفته و درگیر حل دستگاه معادلات درجه بالا نشده و به جای آن از الگوریتم ژنتیک استفاده خواهیم کرد.

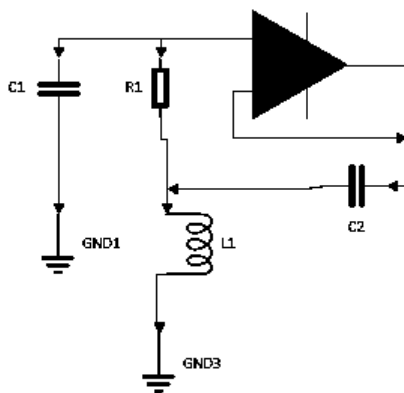


Figure 1 - شکل مدار

در ادامه ابتدا در بخش مسئله پرسش را مطرح کرده و در بخش ژنتیک جزئیات این الگوریتم را توضیح می دهیم و سپس در بخش پیاده سازی جزئیات اعمال الگوریتم ژنتیک ارائه شده و نهایتاً در بخش نتیجه گیری به ارزیابی مسئله می پردازیم.

### مسئله

همانگونه که در مقدمه گفته شد مسئله بدست آوردن پارامترهای مدار بالا با دانستن پهنای باند و ضریب کیفیت و فرکانس نوسان است. در ادامه توضیح مختصری راجع به چگونگی بدست آوردن شروط نوسان در این نمونه اسیلاتورها عنوان شده است.

نمایش نمادین یک اسیلاتور در شکل ۱ نشان داده شده است. در این بلوک،  $\gamma$  ترمینال ورودی جریان امپدانس کم (در حالت ایده آل صفر) است. جریان ورودی به گره  $\gamma$  به ترمینال  $X$  منتقل می شود که یک جریان امپدانس بالا (در حالت ایده آل بی نهایت) است. پورت خروجی ولتاژ تولید شده در ترمینال  $X$  به ترمینال  $Z$  منتقل می شود که یک ترمینال خروجی ولتاژ امپدانس کم (در حالت ایده آل صفر) است. رابطه بین جریان پورت و ولتاژ به صورت زیر بدست می آید:  $v_\gamma = 0$ ,  $i_X = \beta i_\gamma$ ,  $v_Z = \alpha v_X$ . در حالت ایده آل،  $\alpha = 1$  و  $\beta = 1 \pm$  است.

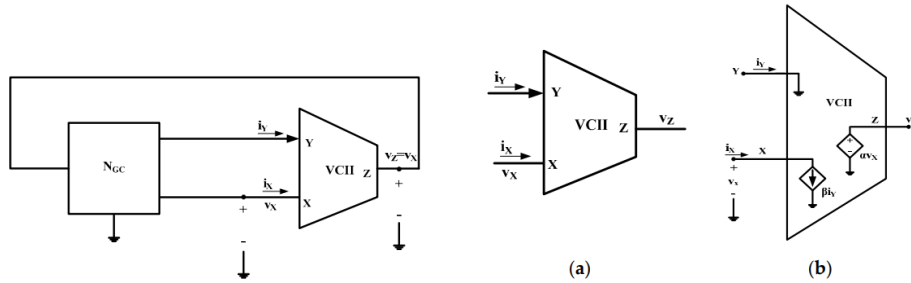


Figure ۲ - شماتیک کلی یک اسپلاتور

معادله مشخصه (CE) کل سیستم را می توان با جایگزینی، در شکل ۳، مانند مدل معادل شکل ۲ b و با در نظر گرفتن یک ورودی فیک در گره Y محاسبه کرد (البته، هیچ سیگنال ورودی در یک مدار نوسان ساز واقعی وجود نخواهد داشت).

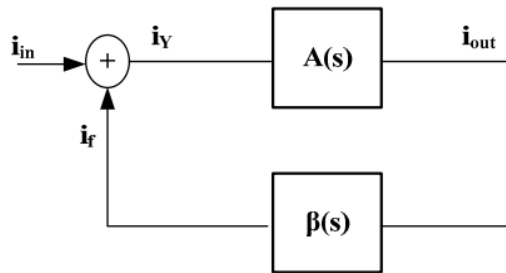


Figure ۳ - شماتیک تابع تبدیل یک اسپلاتور

از این رو، پیکربندی های شکل های ۲ و ۳ را می توان به عنوان یک سیستم بازخورد مثبت دید که تابع انتقال جریان (TF) این گونه بدست می آید.

$$T_I(s) = \frac{i_{out}(s)}{i_{in}(s)} = \frac{A(s)}{1 - A(s)\beta(s)}$$

از آنجا که  $A(s)$  مقادیر مثبت و منفی یک می گیرد و بتا نیز برابر با  $i_f(s)/i_{out}(s)$  است

$$T_I(s) = \frac{i_{out}(s)}{i_{in}(s)} = \frac{\pm 1}{1 \pm \frac{i_f(s)}{i_{out}(s)}}$$

با این حال، از آنجایی که  $i_{out} = -i_x$  و در یک مدار نوسانگر جریان ورودی ( $i_{in} = 0$ ) وجود ندارد، اگر  $i_f = i_y$  پس:

$$T_I(s) = \frac{i_{out}(s)}{i_{in}(s)} = \frac{\pm i_x(s)}{i_x(s) \pm i_y(s)}$$

بنابراین می توانیم شرط وجود (CE) را به صورت زیر استنتاج کنیم.

$$i_x(s) \pm i_y(s) = 0$$

با فرض  $v_y = 0$ ،  $v_z = v_x$ ، توابع ترانسکانداکتانس شبکه غیرفعال در شکل ۲ را می توان با یک عبارت منطقی به صورت زیر بیان کرد:

$$\frac{i_X}{v_Z} = \frac{N_X(s)}{D(s)}, \quad \frac{i_Y}{v_Z} = \frac{N_Y(s)}{D(s)}$$

که در آن  $N_X$  (ها) و  $N_Y$  (ها) به ترتیب در گره های  $X$  و  $Z$  صورت هستند، در حالی که  $D(s)$  یک مخرج مشترک است. با استفاده از این  $CE$  تبدیل می شود:

$$N_X(s) \pm N_Y(s) = 0$$

برای اطمینان از یک نوسان سینوسی خالص،  $CE$  باید یک چند جمله ای مرتبه دوم با ریشه های کاملاً موهومی باشد. این امر مستلزم آن است که فیدبک شبکه حداقل دو خازن داشته باشد. لازم به ذکر است که در شکل ۲، حداقل سه خازن برای ایجاد یک تغییر فاز برای ایجاد یک حلقه بازخورد مثبت مورد نیاز است.  $CE$  شکل ۲ به عنوان یک اسپلاتور با فرض یک شبکه تنها با دو خازن، معادله به شکل زیر خواهد بود:

$$as^2 + bs + c = 0$$

برای شروع نوسان، معیارهای رایج زیر باید رعایت شوند:

$$b = 0, \quad \frac{c}{a} > 0$$

با  $c \neq 0$  و  $a \neq 0$ ، به طوری که با توجه به معیار Barkhausen، قطب های کاملاً موهومی برای تابع انتقال حلقه بسته به دست می آید. فرکانس نوسان:

$$\omega_0 = \sqrt{\frac{c}{a}}$$

حال با توجه به واضح شدن شروط نوسان به سراغ شرح مختصری در مورد الگوریتم ژنتیک می رویم.

## الگوریتم ژنتیک

الگوریتم های ژنتیک دسته ای از الگوریتم های جستجو و بهینه سازی هستند که از اصول انتخاب طبیعی و ژنتیک الهام گرفته شده اند. آن ها از روند تکامل طبیعی تقلید می کنند تا راحل های بهینه برای مسائل پیچیده پیدا کنند. این الگوریتم تکاملی بر روی جمعیتی از تها بالقوه عمل می کند که به صورت کروموزوم یا افراد نمایش داده می شوند. این کروموزوم ها تحت عملیات ژنتیکی مانند جهش (Mutation) و تقاطع (Crossover) قرار می گیرند که منجر به تکامل جمعیت در طول نسل های متوالی می شود. برای پیاده سازی الگوریتم ژنتیک در هوش مصنوعی ابتدا بهتر است که با اجزای آن آشنا شویم، اجزای الگوریتم ژنتیک به صورت فهرست موارد زیر هستند:

- نمونه سازی اولیه (Initialization): فرآیند با ایجاد یک جمعیت اولیه از افراد تصادفی آغاز می شود.
- ارزیابی سازگاری (Fitness Evaluation): سازگاری هر فرد توسط تابعی ارزیابی می شود که عملکرد آن ها را در حل مشکل اندازه گیری می کند.
- انتخاب (Selection): افراد با آمادگی بالاتر احتمال بیشتری برای انتخاب شدن به عنوان والدین برای نسل بعدی دارند.
- تقاطع (Crossover): افراد انتخاب شده تحت crossover قرار می گیرند، جایی که مواد ژنتیکی آن ها برای ایجاد فرزندان ترکیب می شود.
- جهش (Mutation): گاهی اوقات، تغییرات تصادفی در ماده ژنتیکی فرزندان برای حفظ تنوع اثرگذار خواهد بود.
- تکرار (Repeat): فرآیند انتخاب، تقاطع و جهش تا زمانی که یک شرط خاتمه برآورده شود (به عنوان مثال، حداکثر تعداد نسل یا دقت راحل مورد نظر) ادامه می یابد.

در دهه ۱۹۷۰، جان هالند، دانشمندی از دانشگاه میشیگان، مفهوم استفاده از الگوریتم‌های ژنتیک را برای بهینه‌سازی مهندسی معرفی کرد. ایده اساسی پشت این الگوریتم شبیه‌سازی انتقال ویژگی‌های ارثی از طریق ژن‌ها است، دقیقاً شبیه نحوه انتقال صفات انسانی از طریق کروموزوم‌ها. هر ژن در این کروموزوم‌ها نشان دهنده یک ویژگی خاص است. به عنوان مثال، ژن ۱ می‌تواند رنگ چشم، ژن ۲ قد، ژن ۳ رنگ مو و غیره را نشان دهد.

اما در عمل، انتقال کامل کروموزوم‌ها به نسل بعدی اتفاق نمی‌افتد. دو رویداد اولیه به طور همزمان اتفاق می‌افتد. اولین رویداد به عنوان «جهش» *mutation* | شناخته می‌شود، که در آن ژن‌های خاصی دستخوش تغییرات تصادفی می‌شوند. اگرچه تعداد ژن‌های جهش‌یافته معمولاً کم است، اما این تغییرات تصادفی نقش مهمی دارند. به عنوان مثال، ژن مسئول رنگ چشم می‌تواند به طور تصادفی منجر به این شود که فردی در نسل بعدی چشمان سبز داشته باشد، درحالی‌که نسل قبلی عمدتاً دارای چشمان قهوه‌ای بود. جهش تنوع و امکان ظهور صفات جدید را معرفی می‌کند.

رویداد دوم، که بیشتر از جهش رخ می‌دهد، «تقاطع» *crossover* | نامیده می‌شود. در طول تقاطع، دو کروموزوم با یکدیگر ترکیب می‌شوند و قسمت‌هایی را با یکدیگر مبادله می‌کنند. این فرآیند باعث می‌شود که فرزندان ترکیبات متفاوتی از ژن‌ها را در مقایسه با والدین خود به ارث ببرند. تقاطع ترکیب مجدد اطلاعات ژنتیکی را ترویج می‌کند و منجر به فرزندان با ویژگی‌های منحصر به فرد می‌شود.

جهش و تقاطع با هم توانایی الگوریتم ژنتیک را برای کاوش و بهره‌برداری از فضای راحل هدایت می‌کنند. جهش با تغییر تصادفی ژن‌ها، تازگی ایجاد می‌کند و امکان کشف بالقوه صفات جدید و مفید را فراهم خواهد کرد. تقاطع تبادل و باز ترکیب مواد ژنتیکی را تسهیل می‌کند، تنوع و انتشار ویژگی‌های مطلوب را در طول نسل‌ها ارتقا می‌دهد.

با ترکیب این مکانیسم‌های الهام گرفته از تکامل طبیعی، الگوریتم‌های ژنتیک می‌توانند به طور مؤثر فضاهای حل پیچیده را جستجو و بهینه کرده و امکان کشف راحل‌های بهینه یا نزدیک به بهینه را برای طیف وسیعی از مسائل فراهم کنند. امروزه الگوریتم ژنتیک در هوش مصنوعی ترکیب شده است و توانایی‌های بسیار حیرت انگیزی دارد.

## پیاده‌سازی

همانطور که پیش از این نیز اشاره شد، الگوریتم ژنتیک از اصل بقا برزنده‌ترین‌ها در طبیعت، برای ایجاد فرایند جستجو و متعاقباً، جستجو در فضای جواب مسأله استفاده می‌کند. بنابراین، الگوریتم ژنتیک برای حل مسائل بهینه‌سازی (بهینه‌سازی یا کمینه‌سازی) بسیار مناسب خواهد بود. به طور کلی، یک «تابع برزندگی» (*Fitness Function*) یا  $F(i)$ ، در ابتدا با استفاده از تابع هدف فرموله می‌شود و در عملیات ژنتیکی متوالی در نسل‌های الگوریتم ژنتیک مورد استفاده قرار می‌گیرد.

از دیدگاه زیست‌شناسی، برزندگی یک مقدار کیفی (*Qualitative Value*) است که بازده تولید مثل کروموزوم‌ها را می‌سنجد. در الگوریتم ژنتیک، از تابع برزندگی برای محاسبه شانس (یا احتمال) تولید مثل موجودیت‌ها یا کروموزوم‌های موجود در جمعیت نیز استفاده می‌شود؛ به عبارت دیگر، به عنوان معیاری برای مشخص کردن خوب بودن کروموزوم‌ها یا جواب‌های کاندید مسأله مورد استفاده قرار می‌گیرد (و معمولاً این معیار باید بهینه شود).

در الگوریتم ژنتیک، کروموزوم‌ها یا موجودیت‌هایی که بیشترین برزندگی را دارند، نسبت به دیگر کروموزوم‌های موجود در جمعیت، شانس بیشتری برای ترکیب و جهش (عملگرهای ژنتیکی) خواهند داشت. عملکرد صحیح برخی از عملگرهای ژنتیکی منوط به تعریف توابع برزندگی نامنفی (*Non-Negative*) است؛ با این حال، دیگر عملگرهای ژنتیکی، پیش‌شرط نامنفی بودن تابع برزندگی را برای انجام عملیات خود تعریف نمی‌کنند. در مسائل بهینه‌سازی، می‌توان تابع برزندگی را معادل تابع هدف در گرفت. به عبارت دیگر:

$$F(i)=O(i) \rightarrow \text{Objectivefunction}(i)=\text{Fitnessfunction}(i)$$

در مسائل «کمینه‌سازی» (*Minimization*) «»، برای تولید مقادیر نامنفی در تمامی حالات و جهت منعکس کردن برزندگی نسبی رشته‌های متناظر با کروموزوم‌ها یا موجودیت‌های جمعیت، بسیار حیاتی است که تابع هدف اصلی مسأله به تابع برزندگی نگاشت شود. برای انجام چنین نگاشتی، روش‌های متفاوتی وجود دارد. در ادامه، یک روش شایع و پرکاربرد جهت نگاشت تابع هدف به تابع برزندگی در مسائل کمینه‌سازی، معرفی شده‌اند.

$$F(x) = \frac{1}{1 + f(x)}$$

در این رابطه،  $F(x)$  تابع برازندگی و  $f(x)$  تابع هدف مسأله کمینه‌سازی تعریف شده است. این تبدیل، مکان جواب کمینه در فضای جواب‌های مسأله را تغییر نمی‌دهد ولی یک مسأله کمینه‌سازی را به یک مسأله بیشینه‌سازی معادل آن تبدیل می‌کند. حال به توجه به گفته‌های بالا و بخش ابتدایی تابع تبدیل مدار شکل ۱ و نتیجتاً fitness function ان عبارت است از:

$$H(\omega) = \frac{C_1 C_2 R_1^2 \omega^2}{(R_1^2 + \omega C_1^2)(L_1 C_2 \omega^2 - R_1^2)}$$

و بنابراین

$$fitness\ score = \frac{1}{|\omega - desired\ frequency| + |BW - desired\ bandwidth| + |H - 1|}$$

شکل کلی الگوریتم در فلوچارت زیر ارائه شده‌است.

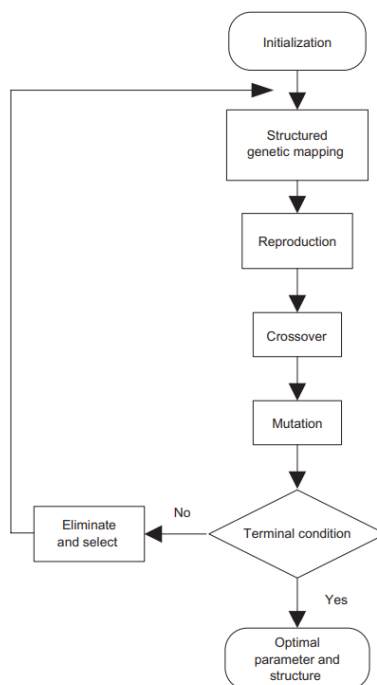


Figure ۴ - الگوریتم ژنتیک در این مسئله

## نتیجه گیری

نتایج حاصل از عملکرد الگوریتم با پارامترهای زیر نشان می‌دهد که الگوریتم موفقیت آمیز عمل کرده و در مدت زمان کوتاهی به جواب همگرا شده‌است. نتایج به همراه پارامترهای الگوریتم در زیر نمایش داده شده است.

population_size	mutation_rate	num_generations	desired_frequency	desired_bandwidth	desired_Q
10	0.1	100	1000 Hz	200 Hz	10

	1	2	3
1	'C1'	7.6996e-13	
2	'L1'	3.9701e-07	
3	'C2'	2.7394e-13	
4	'R1'	38.2346	
5	'C1'	6.7429e-13	
6	'L1'	4.3056e-07	
7	'C2'	4.5274e-13	
8	'R1'	610.8572	

دوپاسخ بالا جواب های نهایی الگوریتم است که در شروط مسئله نیز صدق میکنند و در تصویر زیر خروجی این دو مدار با یک سیگنال سینوسی مقایسه شده است.

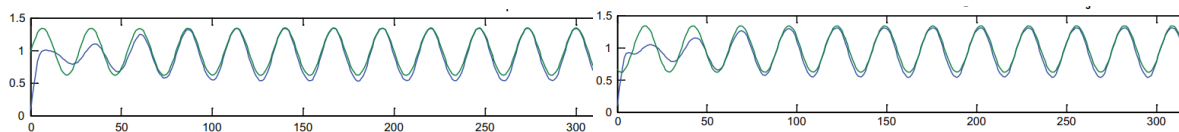


Figure ۵- سیگنال خروجی دو مدار منتخب الگوریتم