

# MOZA Projekt 2

## OA2-B

Jakub Póltorak

22 maja 2022

# Etap 1

## Spis treści

<b>1</b>	<b>Opis projektu</b>	<b>1</b>
<b>2</b>	<b>Zadanie optymalizacji</b>	<b>1</b>
2.1	Zmienne optymalizowane . . . . .	1
2.2	Optymalizacja, funkcja celu i ograniczenia . . . . .	2
2.3	Wybór punktu startowego . . . . .	2
2.4	Wybór algorytmu, skalowanie . . . . .	2
2.4.1	Metoda . . . . .	2
2.4.2	Skalowanie . . . . .	3
2.5	Dokładność algorytmu i gładkość funkcji . . . . .	3
2.6	Uruchamianie kodu Matlaba . . . . .	3

## 1 Opis projektu

Projekt polega na optymalizacji układu dostarczonego przez zamawiającego tak, aby osiągnąć jak najmniejszą powierzchnię i pobór mocy z zasilania przez wzmacniacz.

Układ musi spełniać swoje zadanie (czyli w tym wypadku wzmacniać i generować odpowiedź o odpowiedniej jakości). Dodatkowo należy przeprowadzić analizę Pareto. Optymalizowany układ przedstawiono na poniższym rysunku.

Rysunek 1: Układ wzmacniacza.

## 2 Zadanie optymalizacji

### 2.1 Zmienne optymalizowane

Zadanie polega na optymalizacji powierzchni i mocy (obie jak najmniejsze). Optymalizować można wymiary tranzystorów M1...M7 (szerokość i długość bramek) oraz napięcie VBIAS. Po kilku ręcznych symulacjach układu jako zmienne optymalizowane wybrano jedynie wymiary tranzystorów.

Inne podejście zakładało optymalizację stosunku  $W/L$ , jednak czas optymalizacji wszystkich wymiarów nie jest uciążliwy.

Do funkcji celu dodano dodatkowo czas narastania/opadania w celu uniknięcia sytuacji, gdy optymalizator zmniejsza czasy tak mocno, że odpowiedź układu nie spełnia swojego zadania.

## 2.2 Optymalizacja, funkcja celu i ograniczenia

Poszukujemy minimum funkcji celu:

$$\min[f_1(x), f_2(x), f_3(x)]$$

zależnej od wektora  $x$ :

$$\mathbf{x}_{W_{1-7} \in (20\mu m, 500\mu m), L \in (2\mu m, 100\mu m)} = [W_{1-4} \quad W_5 \quad W_6 \quad W_7 \quad L],$$

gdzie:

$x$  - wektor zmiennych optymalizowanych. Zmienne  $W_{1...7}$  to szerokości bramek, zmienna  $L$  to długość bramki. Wymiary w  $\mu m$ .

$f_1(x)$  - funkcja opisująca powierzchnię całkowitą układu (w  $nm^2$ ),

$f_2(x)$  - funkcja opisująca moc pobieraną z zasilania (wyrażona w mW).

$f_3(x)$  - funkcja  $\max(t_r, t_f)$ , która oblicza czas narastania/opadania (w ns).

Dodatkowo utworzone zostały ograniczenia:

$g_1(x) : f_1(x) < 5,5nm^2$  - ograniczenie pola powierzchni,

$g_2(x) : f_2(x) < 10mW$  - ograniczenie mocy,

$g_3(x) : f_3(x) < 100ns$  - ograniczenie czasu narastania/opadania.

## 2.3 Wybór punktu startowego

Punkt startowy został wybrany metodą prób i błędów.

Przy wyborze punktu startowego kierowano się tym, żeby układ wykonywał swoje zadanie. Ostatecznie wybrano punkt startowy  $x_0$ :

$$x_0 = [50 \quad 30 \quad 40 \quad 30 \quad 15],$$

Odpowiedź układu w punkcie startowym przedstawia poniższy rysunek:

Rysunek 2: Odpowiedź układu w punkcie  $x_0$ .

Jak widać układ wzmacnia sygnał. Otrzymane wyniki nie są jednak optymalne.

## 2.4 Wybór algorytmu, skalowanie

### 2.4.1 Metoda

Przy wyborze metody optymalizacji rozważano kilka opcji. Pierwszymi były metody  $\epsilon$ -ograniczeń oraz skalaryzacja przy pomocy funkcji celu (najprostszy przypadek sumy ważonej).

Metoda  $\epsilon$ -ograniczeń została odrzucona ze względu na mało intuicyjne podejście do problemu.

Metoda skalaryzacji także nie była dość "intuicyjna".

Ostatecznie postanowiono wybrać metodę programowania celowego, głównie ze względu na sposób sformułowania zadania. Metoda programowania celowego pozwala ustawić cele do jakich powinien dążyć algorytm optymalizacji. Cele te można wyznaczyć jako wartości, których oczekujemy i przypisać im odpowiednie wagi, na podstawie których, algorytm wybierze kierunek optymalizacji. Z tych powodów łatwo było utworzyć funkcję celu i określić kierunek w którym powinna dążyć optymalizacja. Dodatkowo ułatwia to ewentualną zmianę parametrów.

#### 2.4.2 Skalowanie

Ogólnie optymalizowane funkcje można skalować tak, żeby ich zakres zmieniał się np. w zakresie od 0 do 1, lub w celu zmniejszenia nakładu obliczeniowego poprzez działania na odpowiednio większych lub mniejszych liczbach.

W przypadku tego zadania skalowanie ograniczono do przeliczenia jednostek na takie, które od razu dają wynik w przeskalowanych jednostkach (np.  $pm^2$  przeliczone na  $nm^2$ , czy W na mW).

### 2.5 Dokładność algorytmu i gładkość funkcji

W badanym przypadku modyfikujemy wymiary tranzystorów, co wpływa głównie na prąd DS. Funkcja opisująca ten prąd jest, przy założeniu, że pozostałe parametry układu, takie jak napięcia zasilania i wymuszenia się nie zmieniają, gładka. Nie jest zatem wymagana żadna modyfikacja funkcji (ani metody) w celu poprawnej optymalizacji.

Gdyby jednak funkcja okazała się niegładka należałoby znaleźć "niebezpieczne" punkty w okolicy rozwiązania, którego oczekujemy i wykorzystać np. metodę aproksymacji.

Przeprowadzono próbna optymalizację i zauważono, że domyślne ustawienia optymalizatora nie dają pożądanego wyniku (optymalizator praktycznie stoi w miejscu). Aby temu zaradzić zmieniono niektóre opcje optymalizatora takie jak tolerancja algorytmu, krok czy ilość iteracji (dokładne opcje w pliku `sim.m`).

### 2.6 Uruchamianie kodu Matlaba

Funkcję celu zrealizowano jako pojedynczą funkcję `run_sim(x)`, która wywołuje symulator i zwraca wyniki w postaci wektora. Aby funkcja działała proszę zmienić w pliku `run_sim.m` wartość zmiennej `spice_string` na ścieżkę do symulatora.