

MOZA Projekt

Wzmacniacz Kaskodowy 4

(wariant B)

Jakub Półtorak

25 maja 2022

Spis treści

1	Sformułowanie matematyczne zadania optymalizacji	2
1.1	Optymalizacja jednokryterialna	2
1.2	Optymalizacja wielokryterialna	3
2	Wyznaczenie przybliżenia początkowego rozwiązania	3
3	Wyznaczanie parametrów roboczych, gładkość funkcji celu, opis kodu	4
3.1	Wyznaczanie parametrów roboczych	4
3.2	Gładkość funkcji celu i ograniczeń	5
3.3	Opis kodu	5
4	Propozycja rozwiązania numerycznego	6
4.1	Algorytm, skalowanie	6
4.2	Przebieg optymalizacji, otrzymane wyniki	8
5	Grafiki w wysokiej rozdzielczości.	10

Etap 1

Opis problemu

Zadanie polega na doborze wartości elementów wzmacniacza tak, aby uzyskać maksymalnie duży iloczyn GBW. Na układ nałożono dodatkowe ograniczenia w postaci minimalnego wzmocnienia dla małych częstotliwości $k_{u0} > 20dB$ ($10 \frac{V}{V}$ dla źródła AC o amplitudzie 1 V) oraz minimalnej częstotliwości granicznej $f_g > 200MHz$ (rozumianej jako częstotliwość spadku o 3 dB względem k_{u0}).

1 Sformułowanie matematyczne zadania optymalizacji

1.1 Optymalizacja jednokryterialna

Poszukiwane jest minimum funkcji celu:

$$\min_{\mathbf{x} \in \mathbf{R}^+} f(\mathbf{x})$$

p.o.

$$g_i(\mathbf{x}) \leq 0 \quad i = 1..n_g$$

gdzie:

$$f(\mathbf{x}) = -(k_{u0} \cdot f_g)$$

\mathbf{x} - wektor zmiennych optymalizowanych:

$$\mathbf{x} = [REE1 \quad REE2 \quad RE \quad RC2 \quad RC3 \quad CEE \quad CG],$$

$k_{u0}(\mathbf{x})$ - wzmocnienie dla małych częstotliwości, rozumiane jako wzmocnienie dla częstotliwości 1 kHz.

$f_g(\mathbf{x})$ - częstotliwość graniczna, rozumiana jako częstotliwość, dla której wzmocnienie spada o 3 dB względem $k_{u0}(\mathbf{x})$.

Parametry $k_{u0}(\mathbf{x})$ oraz $f_g(\mathbf{x})$ obliczane są w Matlabie na podstawie surowych danych ($U_{out}^{AC}(x, f)$) zwracanych przez symulator LTSpice.

Dodatkowo, w zadaniu pojawiają się ograniczenia nieliniowe związane z wymaganiami projektowymi:

- $g_1(\mathbf{x}) : -(\frac{k_{u0}(\mathbf{x})}{k_{u_{min}}} - 1) < 0$

Warunek minimalnego wzmocnienia, $k_{u_{min}} = 20dB$

- $g_2(\mathbf{x}) : -(\frac{f_g(\mathbf{x})}{f_{g_{min}}} - 1) < 0$

Warunek minimalnej częstotliwości granicznej, $f_{g_{min}} = 200MHz$

- $g_3(\mathbf{x}) : b(\mathbf{x}) - b_{max} < 0$
Ograniczenie podbicia charakterystyki, $b_{max} = 1dB$. Podbicie b rozumiane jest jako różnica między maksymalnym poziomem wzmocnienia a k_{u0} . Podbicie jest obliczane w Matlabie.

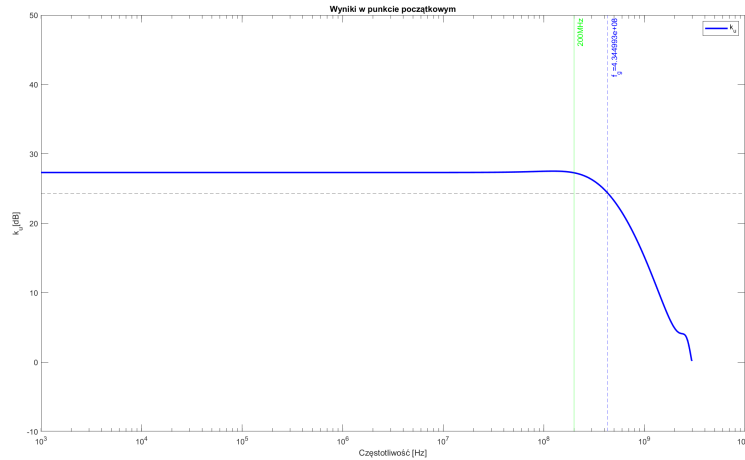
1.2 Optymalizacja wielokryterialna

2 Wyznaczenie przybliżenia początkowego rozwiązania

Zgodnie z poleceniem zmodyfikowano domyślne wartości elementów tak, aby uzyskać rozwiązanie spełniające warunek minimalnej częstotliwości granicznej i wzmocnienia. Ostatecznie, po wybraniu wartości, wektor \mathbf{x} wygląda następująco:

$$\mathbf{x} = [5\Omega \quad 15\Omega \quad 320\Omega \quad 220\Omega \quad 200\Omega \quad 45p \quad 50p],$$

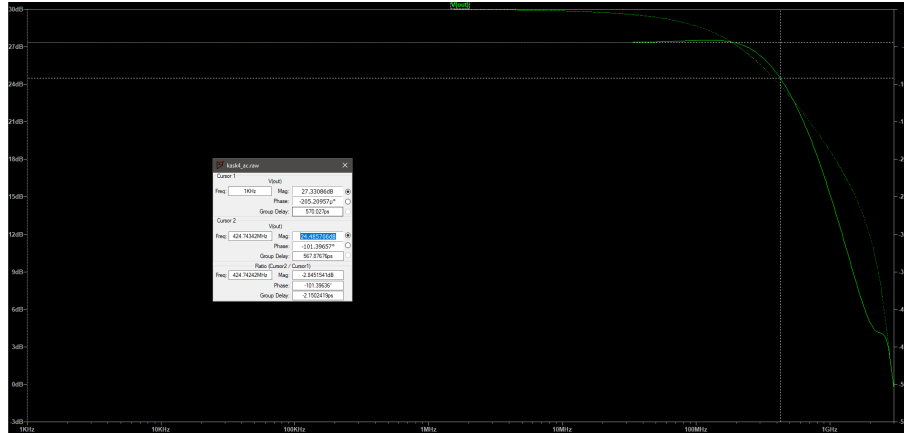
Wyniki w punkcie początkowym można zobaczyć na poniższym wykresie:



Rysunek 1: Charakterystyka układu w punkcie startowym.

Jak widać spełnione są warunki postawione w zadaniu (minimalna wartość wzmocnienia to 20 dB, przy źródle AC mającym 1 V amplitudy) oraz wzmacniacz pracuje prawidłowo (symulacja czasowa wykonana w LTSpice potwierdziła prawidłową pracę układu).

Aby potwierdzić, że Matlab i Spice zwracają te same wyniki przeprowadzono symulację w LTSpice:



Rysunek 2: Charakterystyka układu w punkcie startowym.

3 Wyznaczanie parametrów roboczych, gładkość funkcji celu, opis kodu

3.1 Wyznaczanie parametrów roboczych

W zadaniu badane są trzy parametry: częstotliwość graniczna, wzmocnienie oraz podbicie charakterystyki.

Podbicie b

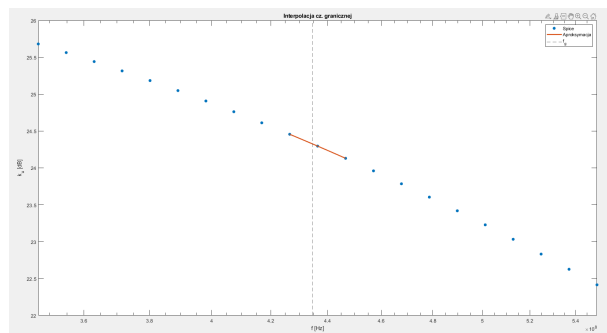
Podbicie rozumiane jest jako różnica między wzmocnieniem k_{u0} a maksymalnym wzmocnieniem jakie osiąga charakterystyka.

Wzmocnienie małowartościowościowe k_{u0}

Wzmocnienie k_u rozumiane jest jako wartość wzmocnienia pozyskana z danych $U_{out}^{AC}(x, f)$ dla częstotliwości 1 KHz.

Częstotliwość graniczna f_g

Częstotliwość graniczna wyznaczana jest jako częstotliwość, dla której wzmocnienie względem k_{u0} spada o 3 dB. Ponieważ LTSpice zwraca wyniki w postaci punktów, uznano, że wymagana jest interpolacja częstotliwości granicznej. Interpolacja pozwoliła zminimalizować "skoki" w funkcji celu. Do interpolacji wykorzystano wielomian drugiego stopnia. Wynik interpolacji przedstawia poniższy wykres:



Rysunek 3: Interpolacja częstotliwości granicznej.

3.2 Gładkość funkcji celu i ograniczeń

Przyjęto, że funkcja celu, w obrębie odpowiednich wartości elementów układu, jest ciągła. Tak długo, jak w układzie zmieniane są wartości pojemności i rezystancji i nie powodują nieprawidłowej pracy układu zawsze możliwe będzie otrzymanie charakterystyki, która, nawet jeśli nie spełnia wymagań projektowych, daje "sensowne" wartości parametrów roboczych (bez dużych skoków np. wzrost wzmocnienia do 10000 dB).

Funckje ograniczeń i celu mogą być w niektórych przypadkach niegładkie. Solver fmincon nie spełnia ograniczeń dla każdej iteracji, przez co teoretycznie może zdarzyć się np. płaska charakterystyka, dla której funkcje nie są gładkie. Przeprowadzone próby wykazały jednak, że problem ten zachodzi bardzo rzadko i ma niewielki wpływ na optymalizację.

3.3 Opis kodu

Dołączony katalog z kodem podzielony został na odpowiednie katalogi dla wyników (results, gdzie zapisany jest workspace z Matlaba i plots, wykresy) i plików dla symulacji(spice). W katalogu głównym znajdują się skrypty i m-funkcje. W celu weryfikacji poprawności działania kodu można uruchomić skrypt starting_point.m, który przedstawia wyniki w pkt. początkowym.

Aby uruchomić optymalizację należy uruchomić skrypt main.m. Po optymalizacji wyniki zostaną zapisane do odpowiednich folderów.

Aby nie czekać aż optymalizator zakończy pracę (ok. 5 minut) można wczytać gotowe wyniki za pomocą komendy load('results/latest.mat').

Opis plików

- main.m - główny skrypt realizujący zadanie optymalizacji.
- display_results.m - Skrypt wyświetlający wyniki optymalizacji. Uruchamiany automatycznie po main.m

- `starting_point.m` - Skrypt obliczający wyniki w pkt. startowym. Do weryfikacji działania funkcji.
- `get_fg.m` - Funkcja obliczająca częstotliwość graniczną.
- `boost.m` - Funkcja obliczająca podbicie charakterystyki.
- `extract_results.m` - Funkcja odczytująca dane z pliku `output_results` powstającego przez funkcję `output_fun`.
- `modify_params.m` - Funkcja modyfikująca parametry w pliku `params.inc`
- `nonlcon.m` - Funkcja nieliniowych ograniczeń nierównościowych.
- `obj_fun.m` - Implementacja funkcji celu.
- `output_fun.m` - Funkcja wyjściowa dla optymalizatora.
- `run_sim.m` - Funkcja uruchamiająca symulator LTSpice.
- `LTSpice2Matlab.m` - Funkcja do odczytu danych z LTSpice.

4 Propozycja rozwiązania numerycznego

4.1 Algorytm, skalowanie

Solver i algorytm optymalizacji

Jako solver wykorzystano `fmincon` z domyślnym algorytmem (Interior Point). Zdecydowano się na wykorzystanie metod gradientowych, ponieważ założono gładkość funkcji dla danych ograniczeń.

Po próbach przeprowadzonych w LTSpice ustalono, że niektóre elementy mają większy wpływ na układ niż inne, jednak, przy różnych kombinacjach wartości, wpływ różnych elementów jest trudny do przewidzenia. Stwierdzono np. że zmniejszanie wartości `REE1` powoduje wzrost wzmocnienia przy spadku pasma, manipulowanie pojemnością `CEE` wpływa na pasmo, `CG` na podbicia itp. Ponieważ nie znaleziono jednoznacznej zależności (działanie na kształt algorytmu Gaussa-Seidla) postanowiono nie ograniczać liczby optymalizowanych zmiennych, a jedynie zadbać o odpowiednie ograniczenia kostkowe parametrów.

Skalowanie

Zarówno wektor wartości elementów jak i funkcja celu zostały przeskalowane.

W przypadku wektora parametrów optymalizowanych zastosowano proste skalowanie do 1 względem punktu startowego $\frac{x}{x_0}$. W badanym przypadku wartości są podobnego rzędu (jednostki zostają dopisane dopiero w funkcji modyfikującej plik z parametrami), jednak dla przejrzystości postanowiono wykonać skalowanie wektora względem wektora startowego.

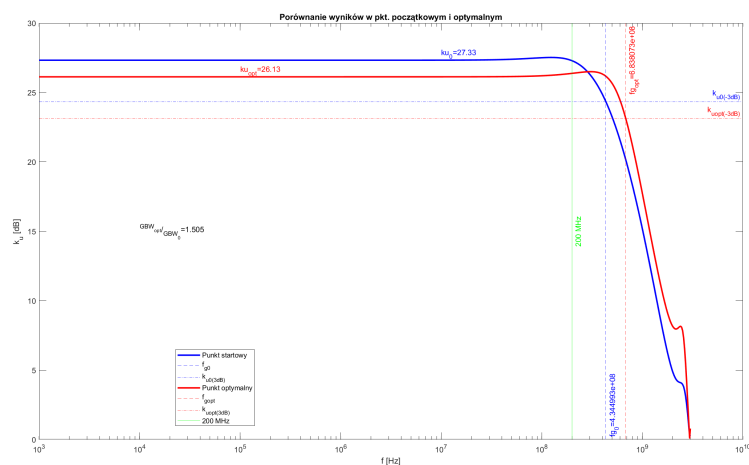
W przypadku funkcji celu iloczyn wzmocnienia i częstotliwości granicznej sięga rzędu 10^9 . Aby usprawnić pracę optymalizatora wyjście z zaimplementowanej funkcji celu jest postaci $-\log(GBW)$.

4.2 Przebieg optymalizacji, otrzymane wyniki

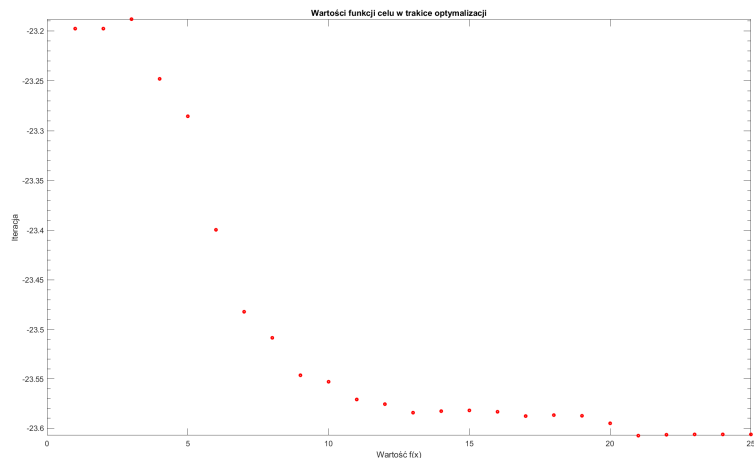
Zmieniane są wszystkie wartości parametrów dostępne w zadaniu. Ograniczenia dobrano na podstawie metody prób i błędów i zapisano w postaci wektorów lb i ub w pliku main.m. Dolne ograniczenie zapewnia, że wszystkie elementy mają dodatnie wartości.

Przebieg optymalizacji zobaczyć można poniżej:

	k_u [dB]	f_g [MHz]	GBW [$dB \cdot MHz$]	b [dB]
P. start	27,33	4,345	1,187	0,198
P. opt.	26,13	6,838	1,781	0,371



Rysunek 4: Porównanie punktu optymalnego i startowego.



Rysunek 5: Przebieg wartości funkcji celu.

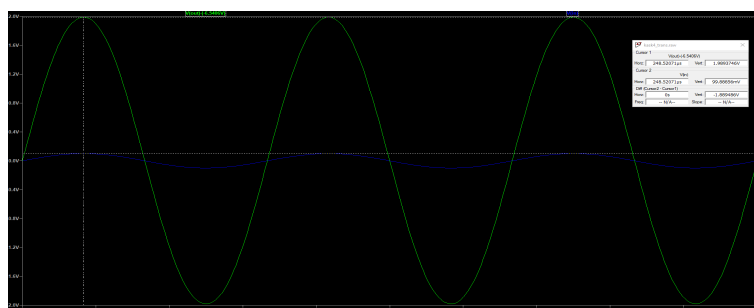
```
Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.

<stopping criteria details>

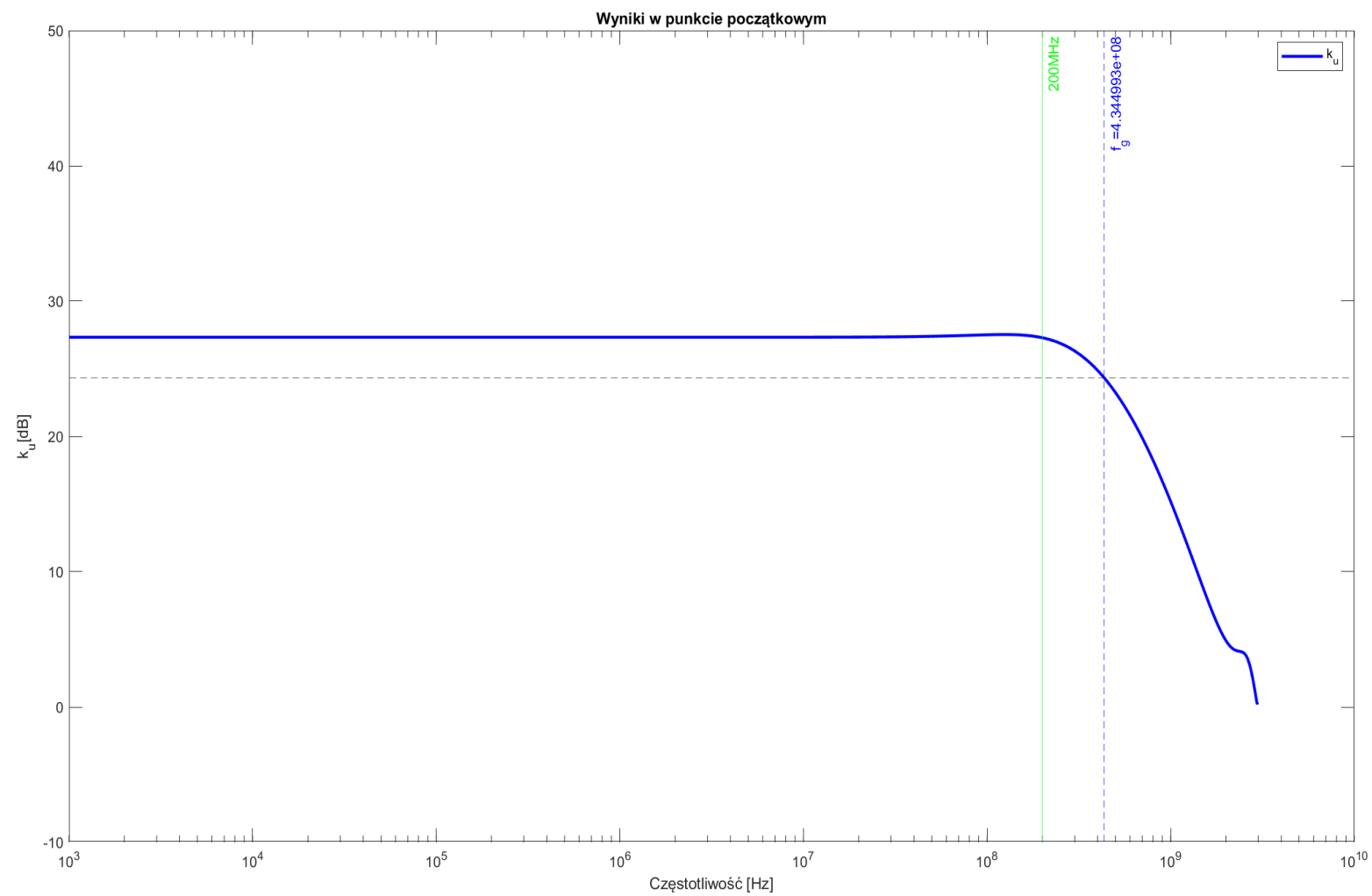
Optimization stopped because the relative changes in all elements of x are
less than options.StepTolerance = 1.000000e-10, and the relative maximum constraint
violation, 0.000000e+00, is less than options.ConstraintTolerance = 1.000000e-06.
```

Rysunek 6: Informacja o zakończeniu pracy przez optymalizator.

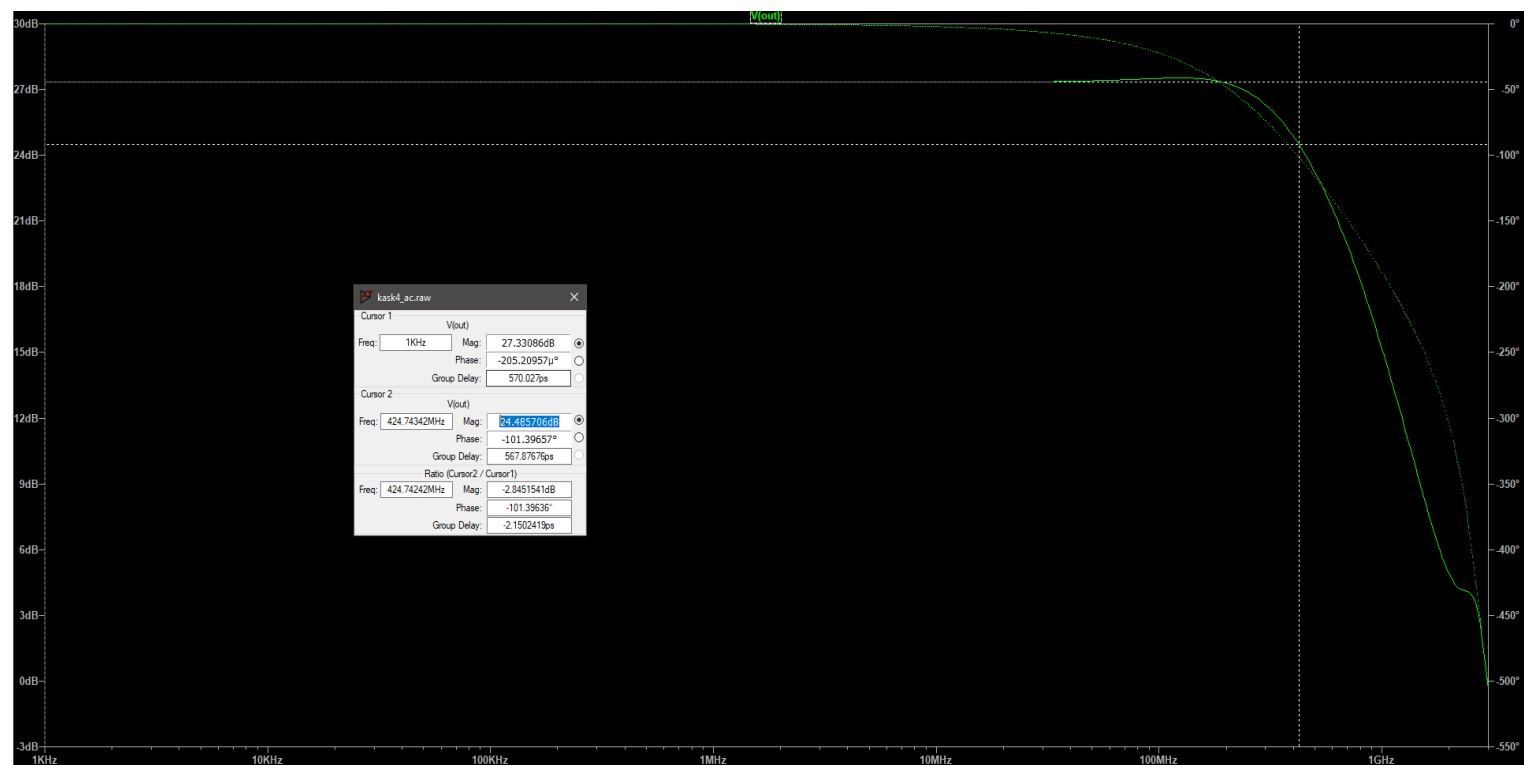


Rysunek 7: Symulacja czasowa w pkt. optymalnym. Wzmacniacz wzmacnia.

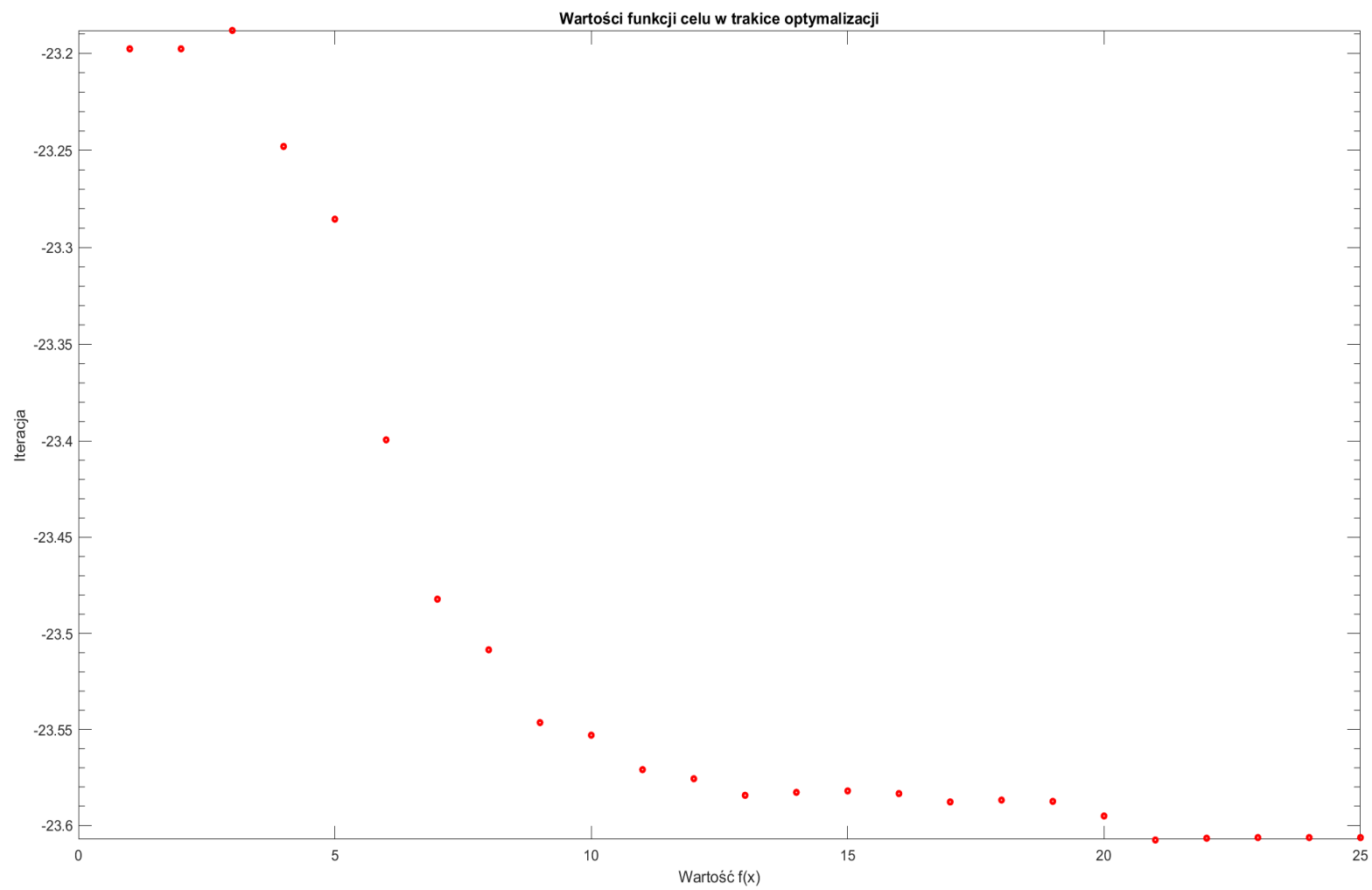
5 Grafiki w wysokiej rozdzielczości.



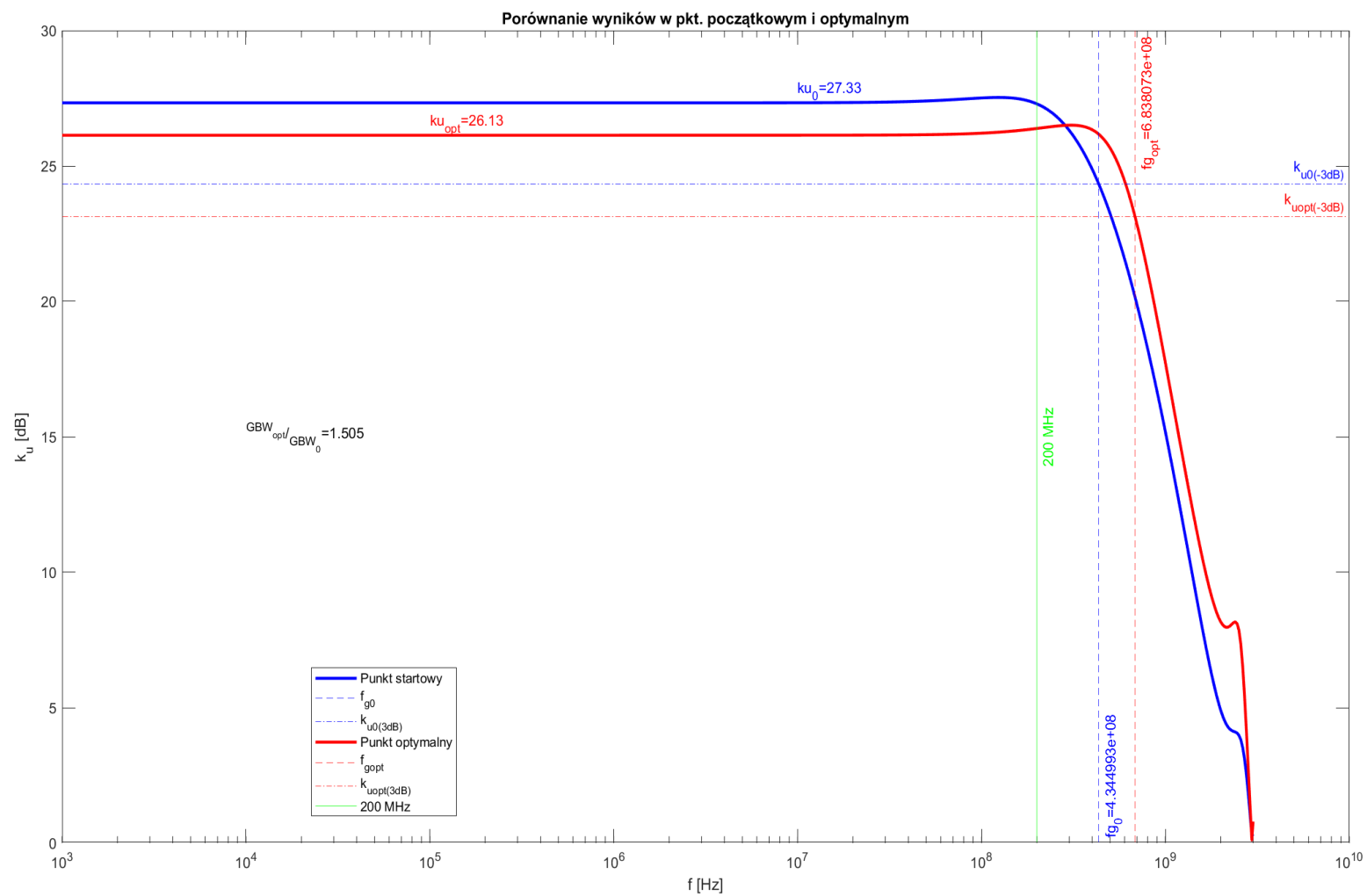
Rysunek 8: Charakterystyka układu w punkcie startowym.



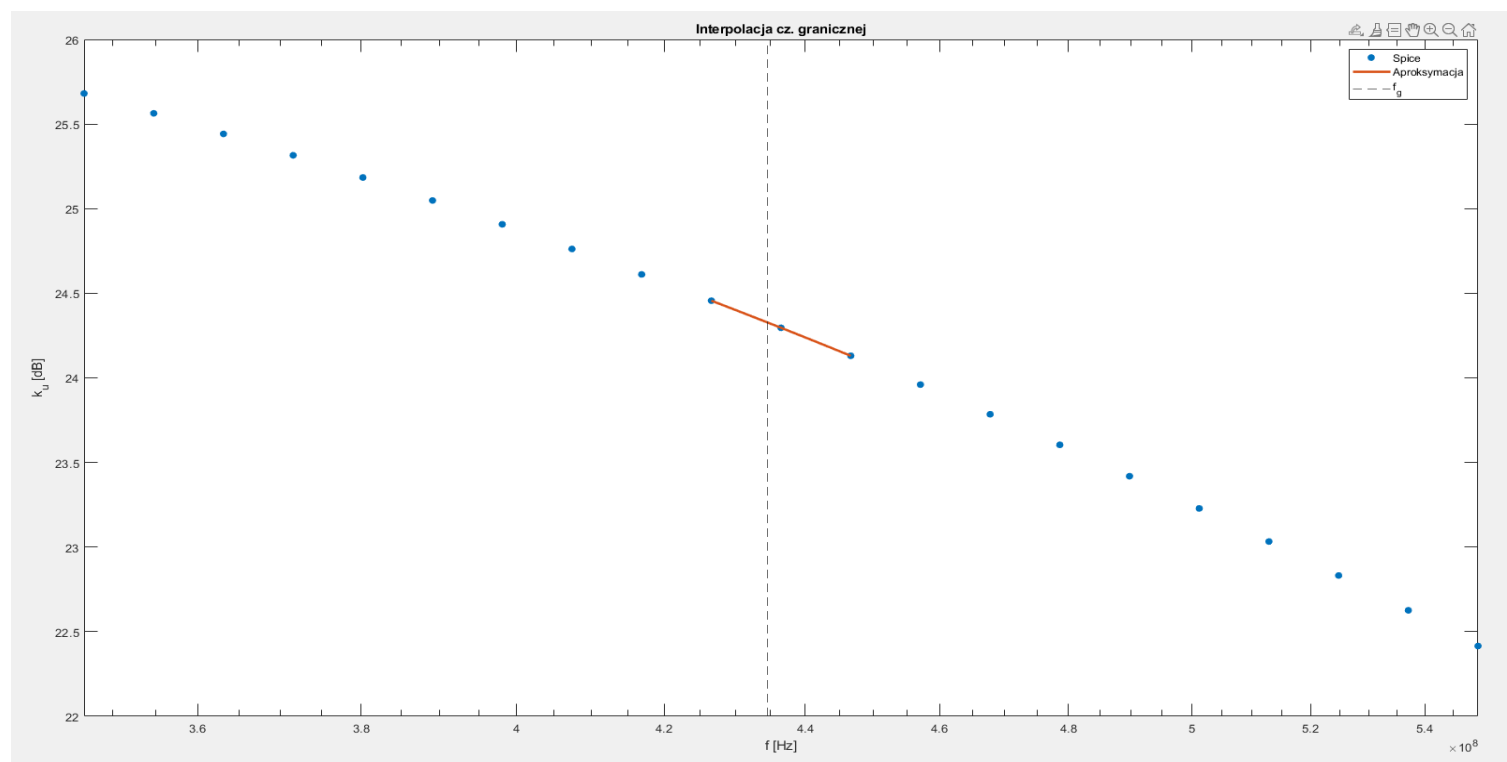
Rysunek 9: Charakterystyka układu w punkcie startowym (LTSpice).



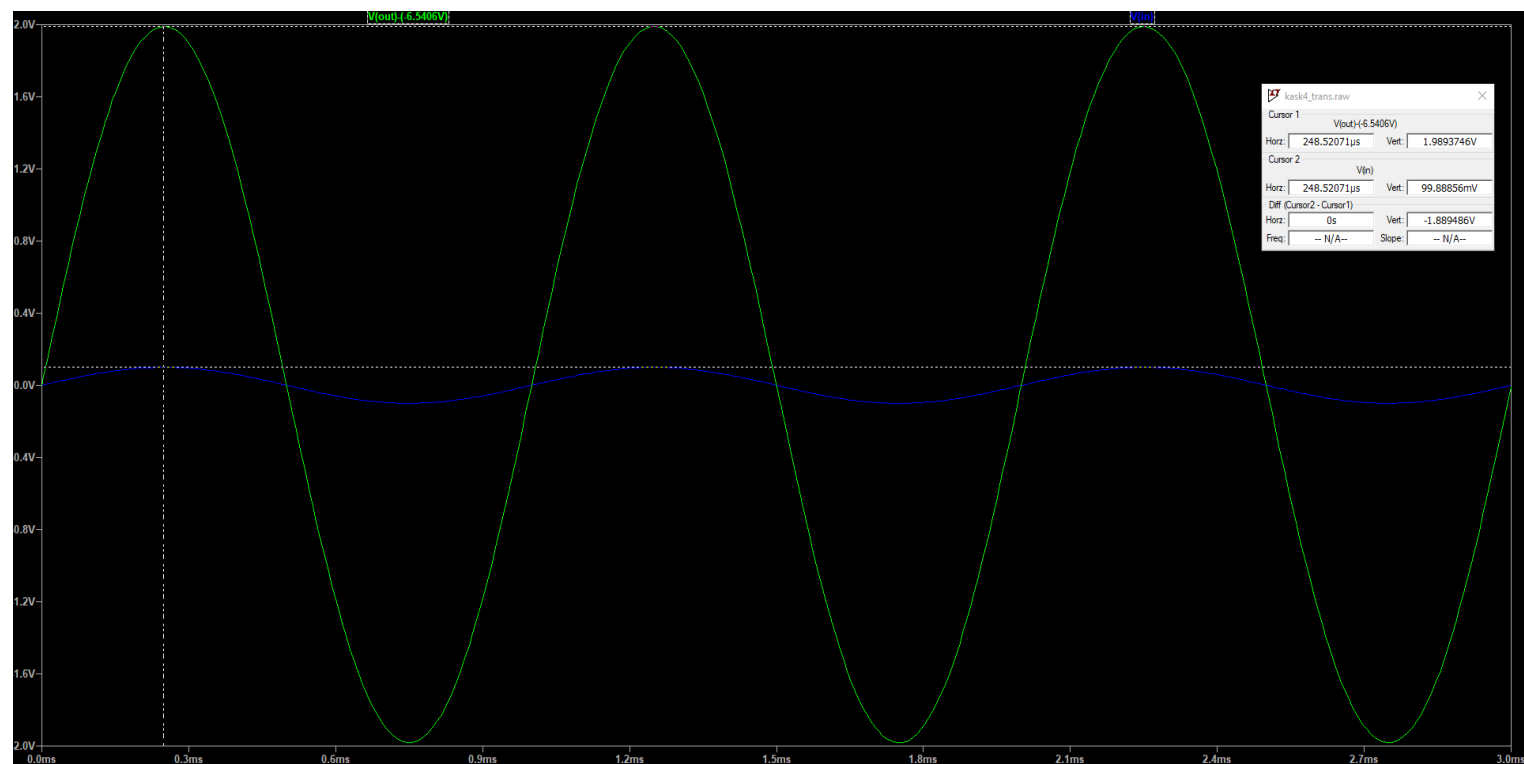
Rysunek 10: Przebieg wartości funkcji celu.



Rysunek 11: Porównanie wyników w punkcie optymalnym i startowym.



Rysunek 12: Interpolacja częstotliwości granicznej.



Rysunek 13: Symulacja czasowa w pkt. optymalnym. Wzmacniacz wzmacnia.