MOZA Projekt
Wzmacniacz Kaskodowy 4
(wariant B)

Jakub Półtorak

25 maja 2022

Spis treści

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:

$$\mathbf{x} = \begin{bmatrix} REE1 & REE2 & RE & RC2 & RC3 & CEE & CG \end{bmatrix},$$

$k_{u0}(\mathbf{x})$ - 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 $k_{u_0}(\mathbf{x})$.

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, $k_{umin} = 20dB$
- $g_2(\mathbf{x}) : -(\frac{f_g(\mathbf{x})}{f_{g_{min}}} - 1) < 0$
  Warunek minimalnej częstotliwości granicznej, $f_{gmin} = 200MHz$. Częstotliwość graniczna f_g obliczana jest jako częstotliwość, dla której wzmocnienie spada o 3 dB względem ku_0.

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

1.2 Optymalizacja wielokryterialna

Poszukiwane jest minimum funkcji celu:

$$\min_{\mathbf{x} \in \mathbf{R}^+} [f_1(\mathbf{x}), f_2(\mathbf{x})]$$

p.o.

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

gdzie:

$$f_1(\mathbf{x}) = -k_{u0}$$

$$f_2(\mathbf{x}) = -f_g$$

$\mathbf{x}$ - wektor zmiennych optymalizowanych:

$$\mathbf{x} = \begin{bmatrix} REE1 & REE2 & RE & RC2 & RC3 & CEE & CG \end{bmatrix},$$

$k_{u0}(\mathbf{x})$ - 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 $k_{u_0}(\mathbf{x})$.

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 (identyczne jak dla optymalizacji jednokryterialnej):

- $g_1(\mathbf{x}) : -(\frac{k_{u0}(\mathbf{x})}{k_{u_{min}}} - 1) < 0$
  Warunek minimalnego wzmocnienia, $k_{umin} = 20dB$

- $g_2(\mathbf{x}) : -(\frac{f_g(\mathbf{x})}{f_{g_{min}}} - 1) < 0$
  Warunek minimalnej częstotliwości granicznej, $f_{gmin} = 200MHz$. Częstotliwość graniczna f_g obliczana jest jako częstotliwość, dla której wzmocnienie spada o 3 dB względem ku_0.

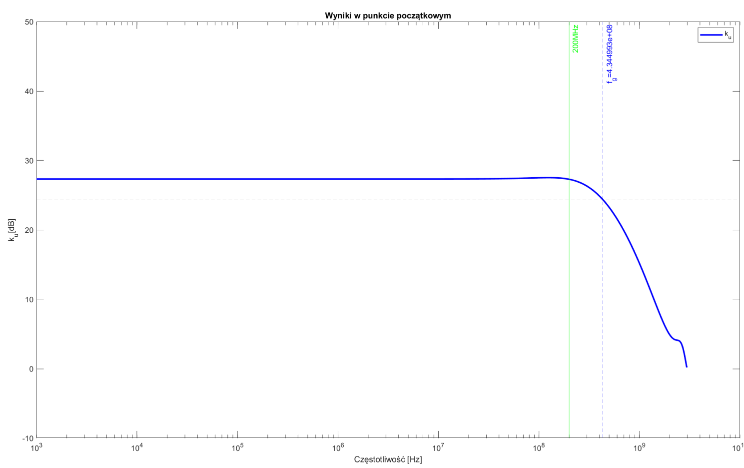
- $g_3(\mathbf{x}) : b(\mathbf{x}) - b_{max} < 0$
  Ograniczenie podbicia charakterystyki, $b_{max} = 1dB$. Podbicie $b(\mathbf{x})$ rozumiane jest jako różnica między maksymalnym poziomem wzmocnienia a k_{u0} ((czyli $b(\mathbf{x}) = max(k_u(\mathbf{x})) - b_{max}$)).Podbicie jest obliczane w Matlabie.

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 **x** wygląda następująco:

$$\mathbf{x} = \begin{bmatrix} 5\Omega & 15\Omega & 320\Omega & 220\Omega & 200\Omega & 45p & 50p \end{bmatrix},$$

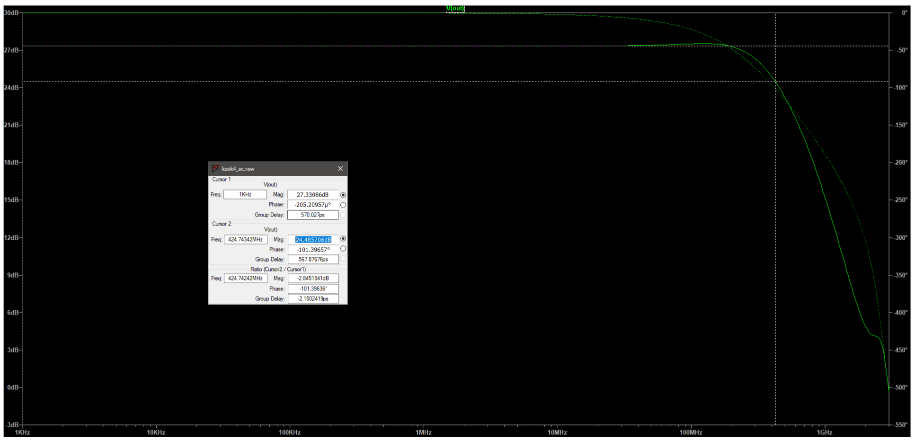
Wyniki w punkcie początkowym można zobaczyć na poniższym wykresie:



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 wykonan 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.

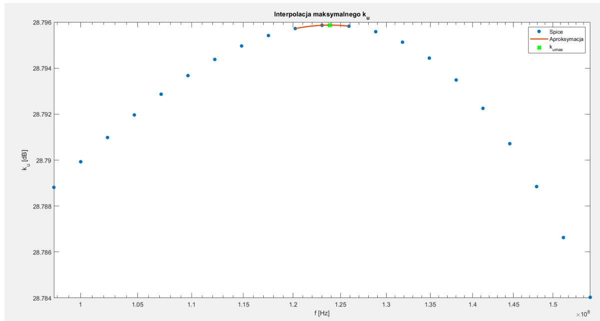
3 Wyznaczanie parametrów roboczych, gładkość funkcji celu, opis kodu

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 ($b(\mathbf{x}) = max(k_u(\mathbf{x})) - b_{max}$). Maksimum charakterystyki jest wyznaczane przez interpolację (wielomian drugiego stopnia). Pozwala to dokładniej ustalić maksymalne wzmocnienie i wygładza funkcję ograniczeń. Efekty interpolacji widać na poniższym wykresie:



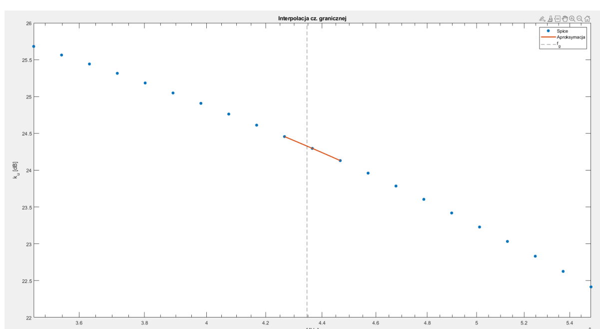
Rysunek 3: Interpolacja maksimum charakterystyki.

Wzmocnienie małoczęstotliwościowe 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_{u_0} 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 4: 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).

Funkcje ograniczeń i celu mogą być w niektórych przypadkach niegładkie. Może dojść do takiej sytuacji, gdy aktualny zestaw wartości elementów spowoduje powstanie charakterystyki, dla której niektóre z parametrów roboczych nie są możliwe do obliczenia lub nie mają fizycznego sensu (np. płaska charakterystyka na poziomie 0 dB spowoduje, że nie istnieje częstotliwość graniczna rozumiana wg. zasad zdefiniowanych na początku raportu, funkcja obliczająca zwraca wtedy wartość NaN). Punkty przejścia między dobrą a złą charakterystyką są właściwie punktami, gdzie funkcja nie jest gładka. Solver fmincon nie spełnia ograniczeń dla każdej iteracji, przez co teoretycznie może zdarzyć się np. płaska charakterystyka, dla której funkcje nie są gładkie, jednak wg. dokumentacji, fmincon jest w stanie wrócić z takich punktów do poprawnej pracy. Przeprowadzone próby wykazały jednak, że problem ten zachodzi bardzo rzadko i ma niewielki wpływ na optymalizację.

W przypadku optymalizacji wielokryterialnej zdecydowano się wykorzystać solver paretosearch (wykorzystujący wewnętrznie algorytm patternsearch), niegładkość funkcji celu i ograniczeń nie ma tutaj więc dużego znaczenia (metoda bezgradientowa).

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.
- optmization_wrapper.m - funkcja będąca nakładką na optymalizator pozwalającą na współdzielenie między optymalizatorem a funkcjami celu i ograniczeń obliczonych wartości. Funkcja ta wykorzystuje zagnieżdżone

funkcje celu i ograniczeń (oraz skalowania). Wszystkie ustawienia (dolne i górne ograniczenia, opcje itp.) znajdują się w tym pliku.

- multiobj_optimization_wrapper.m - Funkcja będąca nakładką na optymalizator wielokryterialny. Podobnie jak w wersji jednokryterialnej steruje całym procesem optymalizacji.
- 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 funckję output_fun.
- modify_params.m - Funkcja modyfikująca parametry w pliku params.inc
- 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 jednokryterialnej

Jako solver wykorzystano fmincon z domyślnym algorytmem (Interior Point). Zdecydowano się na wykorzystanie metod gradientowych, ponieważ założono gładkość funkcji dla danych ograniczeń. Po przeprowadzonych próbach optymalizacji uznano, że założenie jest słuszne (niegładkość jedynie w punktach nie dających wyników o sensie fizycznym, widoczna poprawa parametrów układu, wyraźna zbieżność).

Po próbach przeprowadzonych w LTSpice ustalono, że niektóre elementy mają większy wpływ na układ niż inne, jednak, przy różnych kombinacjach wartości, wpływ rożnych 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.

Solver i algorytm dla optymalizacji wielokryterialnej

W praktyce ciężko określić co jest dobrym rozwiązaniem zadania wielokryterialnego. Przeważnie mamy do czynienia z optymalizowaniem parametrów gdzie poprawa jednego powoduj epogorszenie pozostałych. Aby znaleźć rozwiązanie zadanego problemu zdecydowano się użyć solvera paretosearch (wykorzystującego algorytm patternsearch) do wyznaczenia zbioru Pareto, który powinien dać wybór w kwestii czy bardziej interesuje użytkownika wzmocnienie czy pasmo.

Ponieważ patternsearch jest algorytmem bezgradientowym ewentualny brak gładkości f. celu i ograniczeń nie ma tutaj dużego znaczenia.

Skalowanie

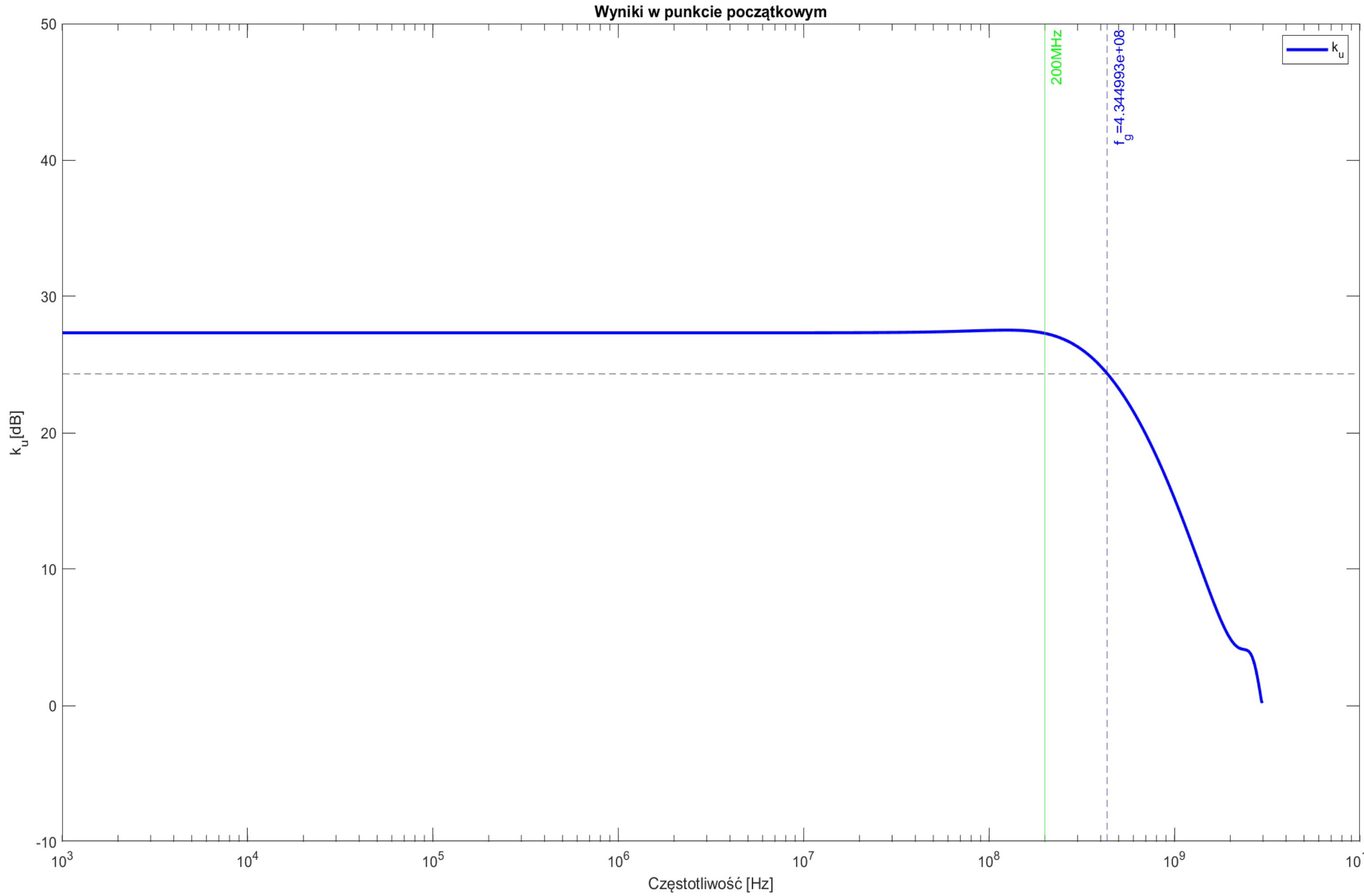
Zarówno wektor wartości elementów jak i funkcja celu zostały przeskalowane.

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.

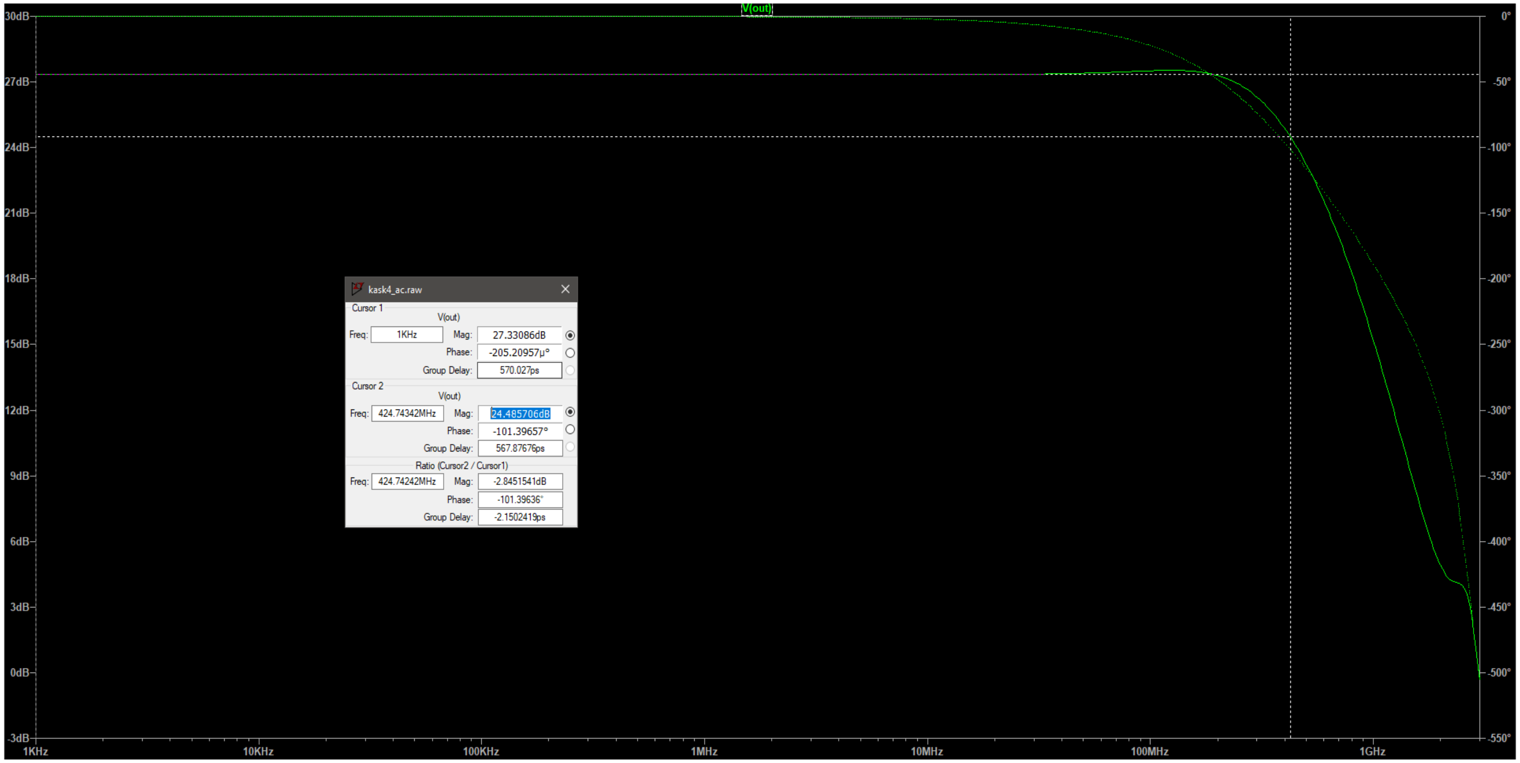
W przypadku funkcji celu iloczyn wzmocnienia i częstotliwości granicznej sięga rzędu 10^9. Aby usprawnić pracę optymalizatora wyjście z zaimplementowanej funkcji celu jest postaci $-log_{10}(GBW)$.

W przypadku optymalizacji wielokryterialnej częstotliwość graniczna zwracana jest w postaci $-log_{10}(f_g)$.

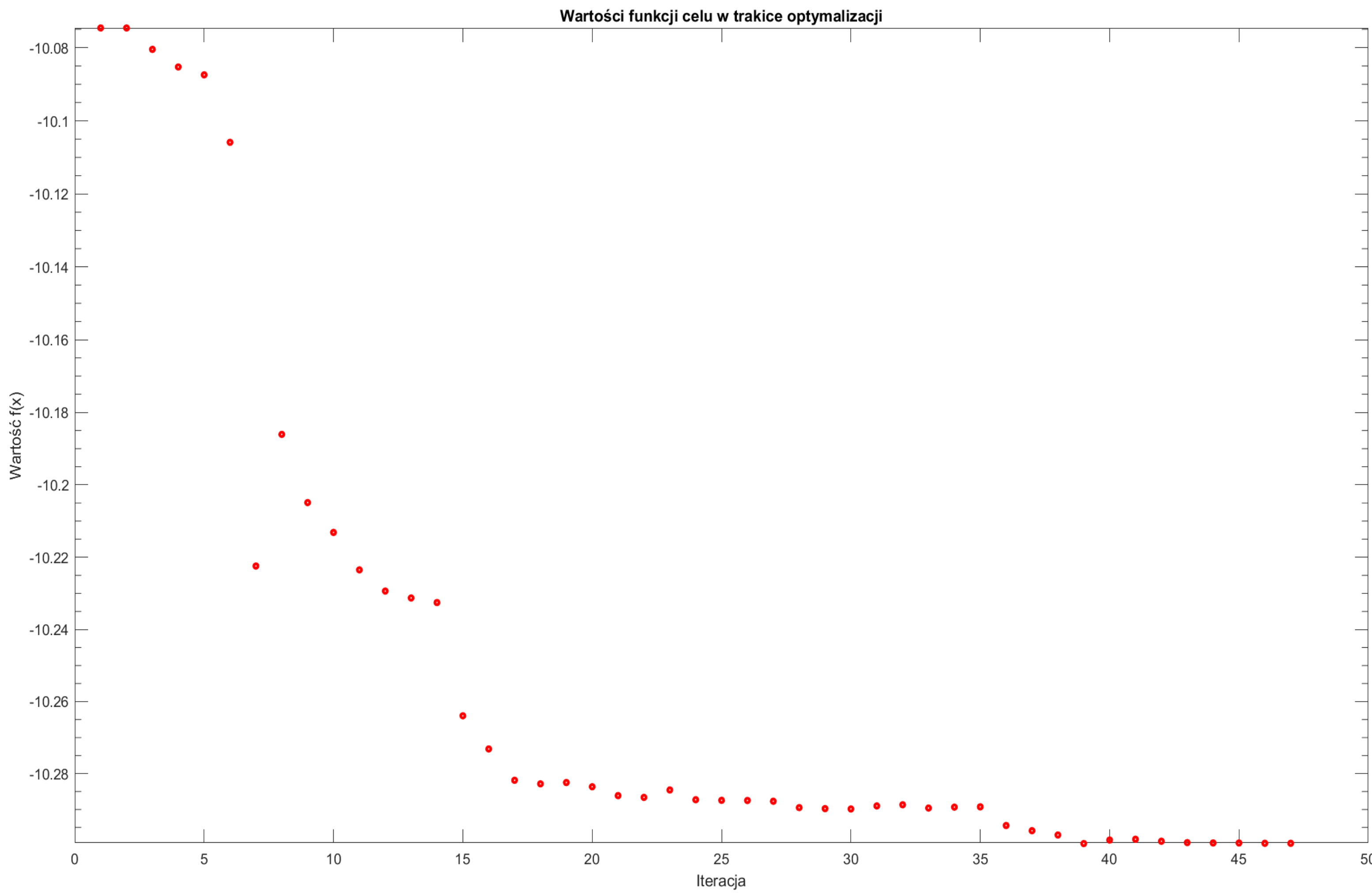
5 Grafiki w wysokiej rozdzielczości.



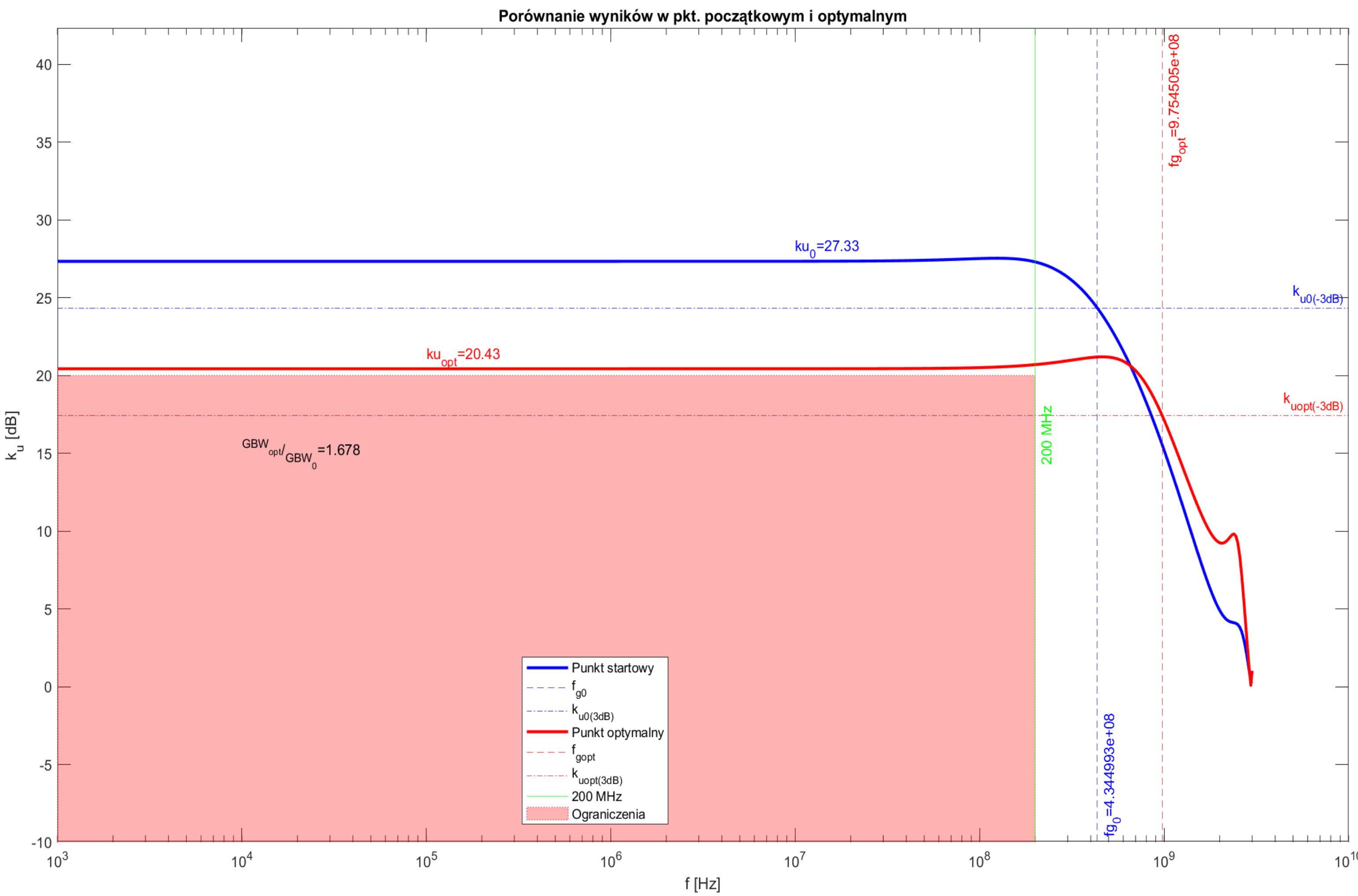
Rysunek 5: Charakterystyka układu w punkcie startowym.



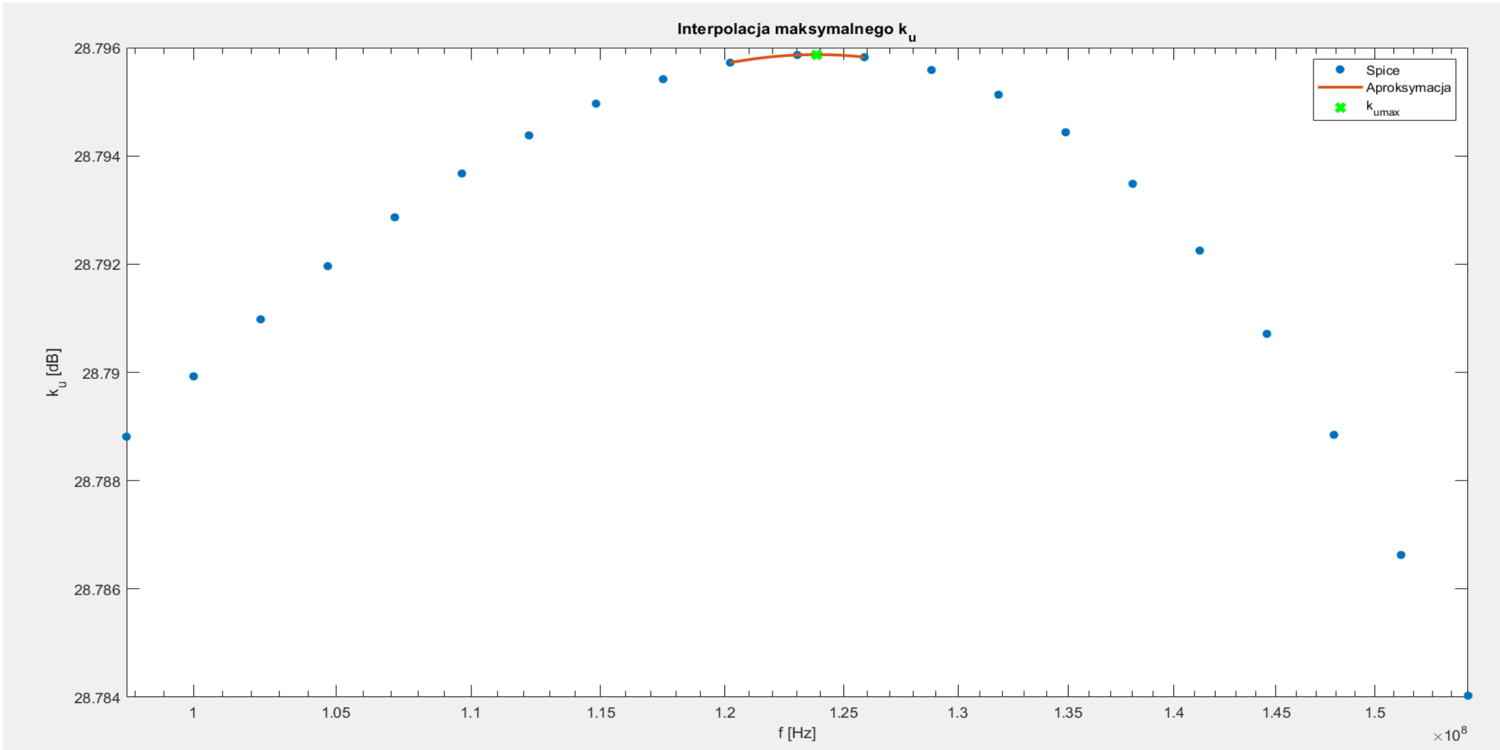
Rysunek 6: Charakterystyka układu w punkcie startowym (LTSpice).



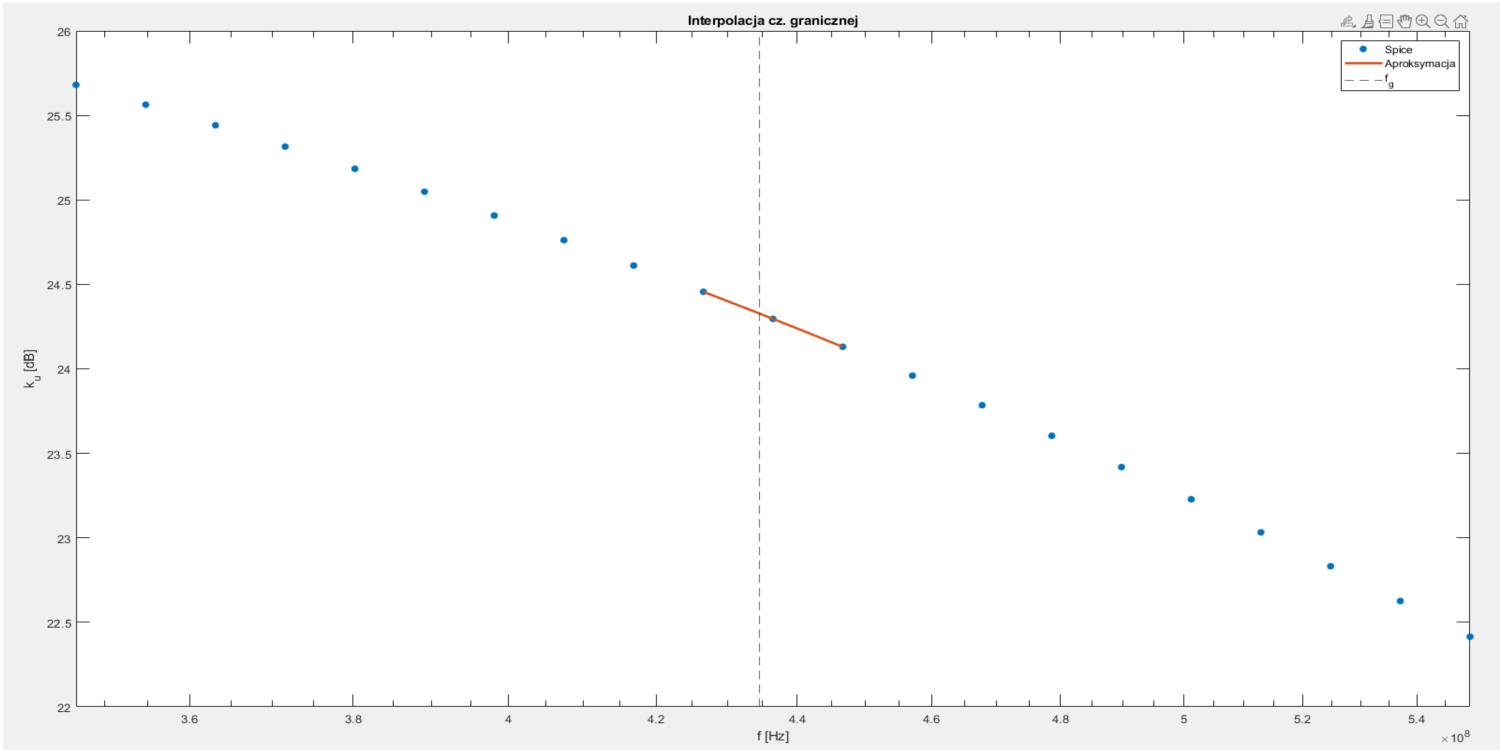
Rysunek 7: Przebieg wartości funkcji celu.



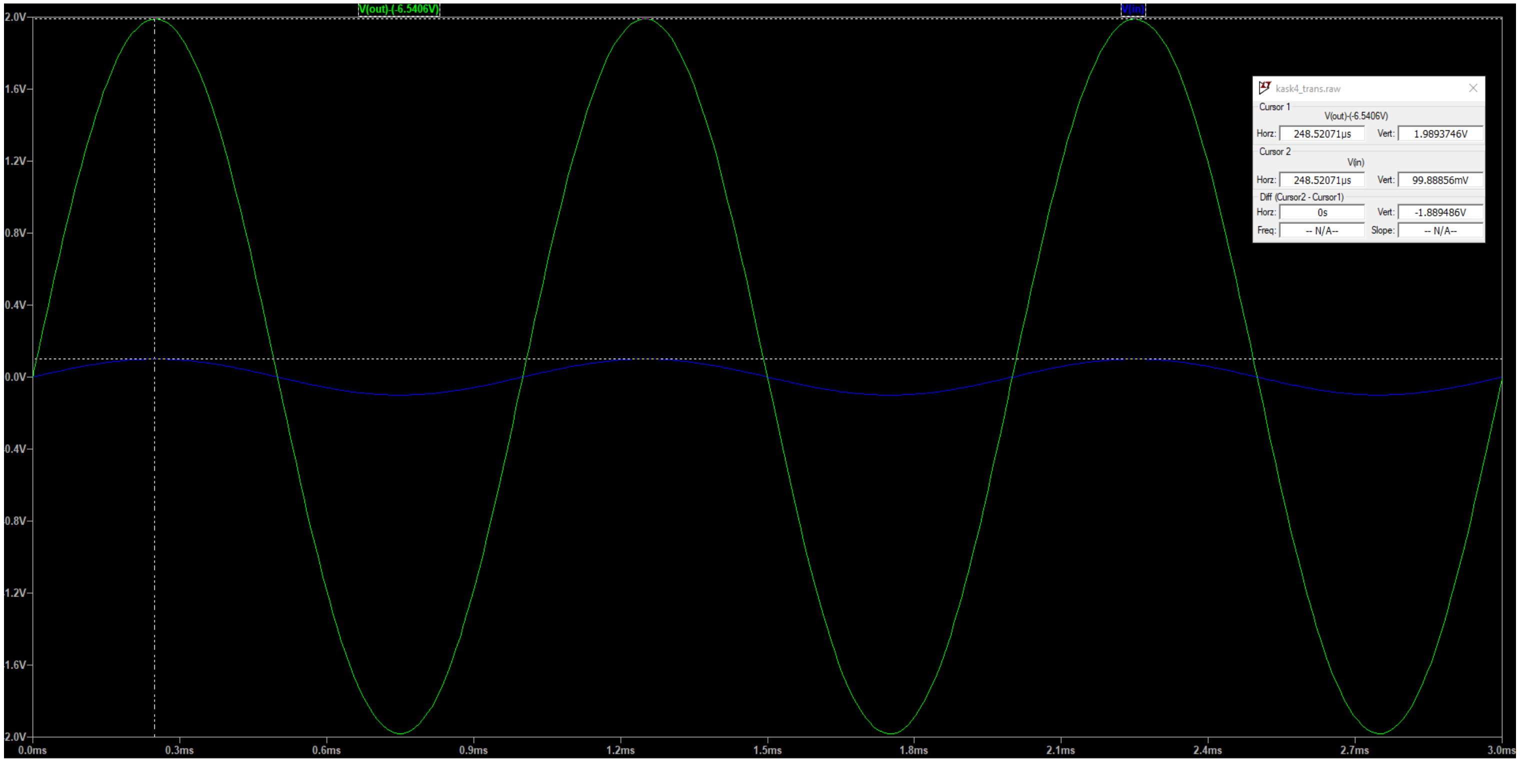
Rysunek 8: Porównanie wyników w punkcie optymalnym i startowym.



Rysunek 9: Interpolacja maksymalnego wzmocnienia.



Rysunek 10: Interpolacja częstotliwości granicznej.



Rysunek 11: Symulacja czasowa w pkt. optymalnym. Wzmacniacz wzmacnia.