

Laboratorium nr 4

Zadanie:

Wykonaj w formie programistycznej implementacji poniżej przedstawione zadania.

1) Wygeneruj sygnały zmodulowane $z_A(t)$ oraz $z_P(t)$ dla następujących przypadków:

- a) $1 > k_A > 0; k_P < 2;$
- b) $12 > k_A > 2; \pi > k_P > 0;$
- c) $k_A > \hat{B}\hat{A}; k_P > \hat{A}\hat{B}$

Wykonaj wykresy, w tym sygnału informacyjnego.

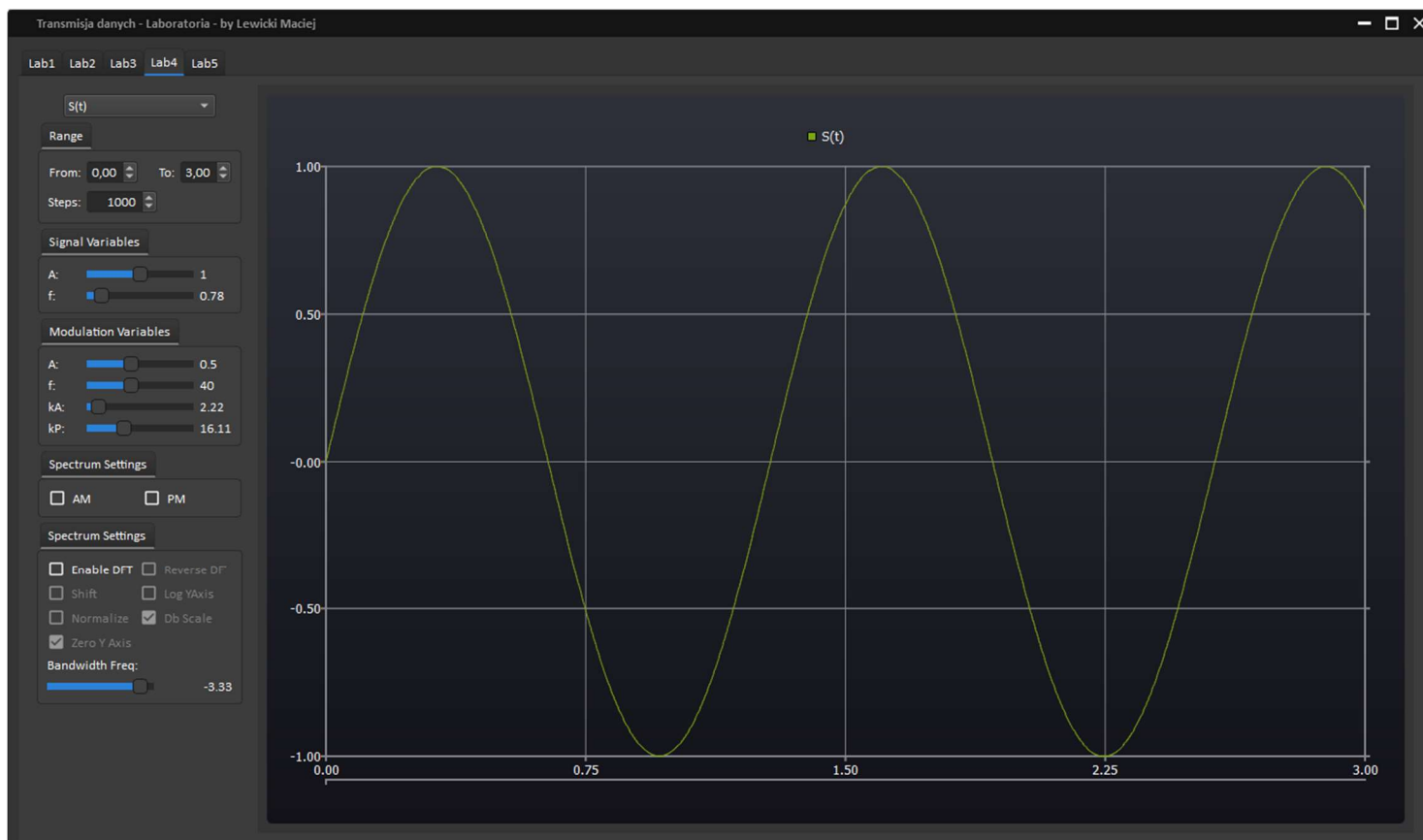
2) Wykonaj wykresy widm amplitudowych sygnałów zmodulowanych $z_A(t)$ oraz $z_P(t)$. Należy tak dobrać skalę (liniową lub logarytmiczną) osi poziomej i pionowej aby jak najwięcej prążków widma było widocznych na wykresie.

3) Zbadaj szerokości pasma sygnałów zmodulowanych (dla poziomu -3dB) wykonując wyznaczenie granicy f_{min} i f_{max} oraz obliczenie

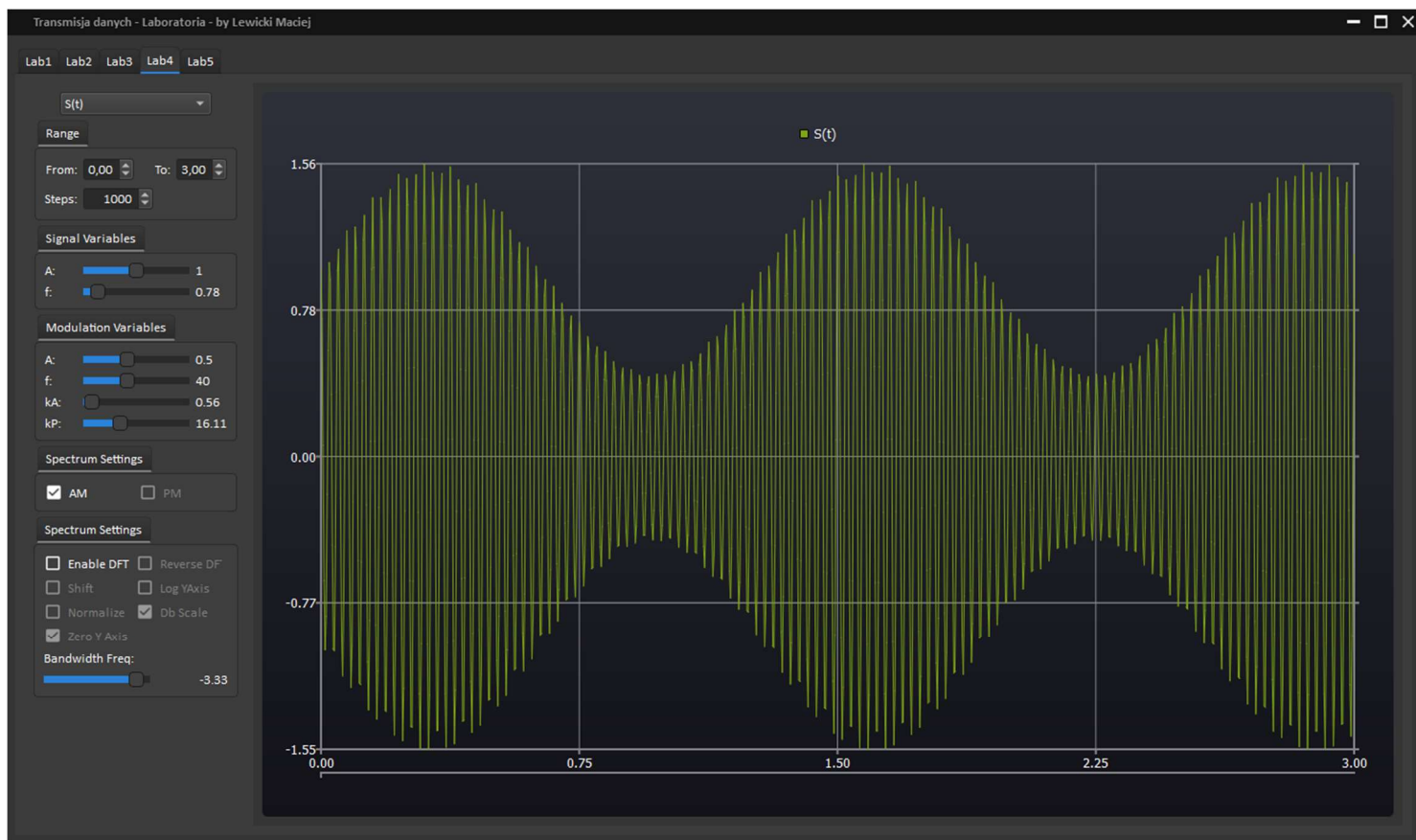
$$W = f_{max} - f_{min}$$

Szerokości wyznaczonych w zadaniu pasm dla poszczególnych aproksymacji zapisz w formie komentarza w kodzie programu.

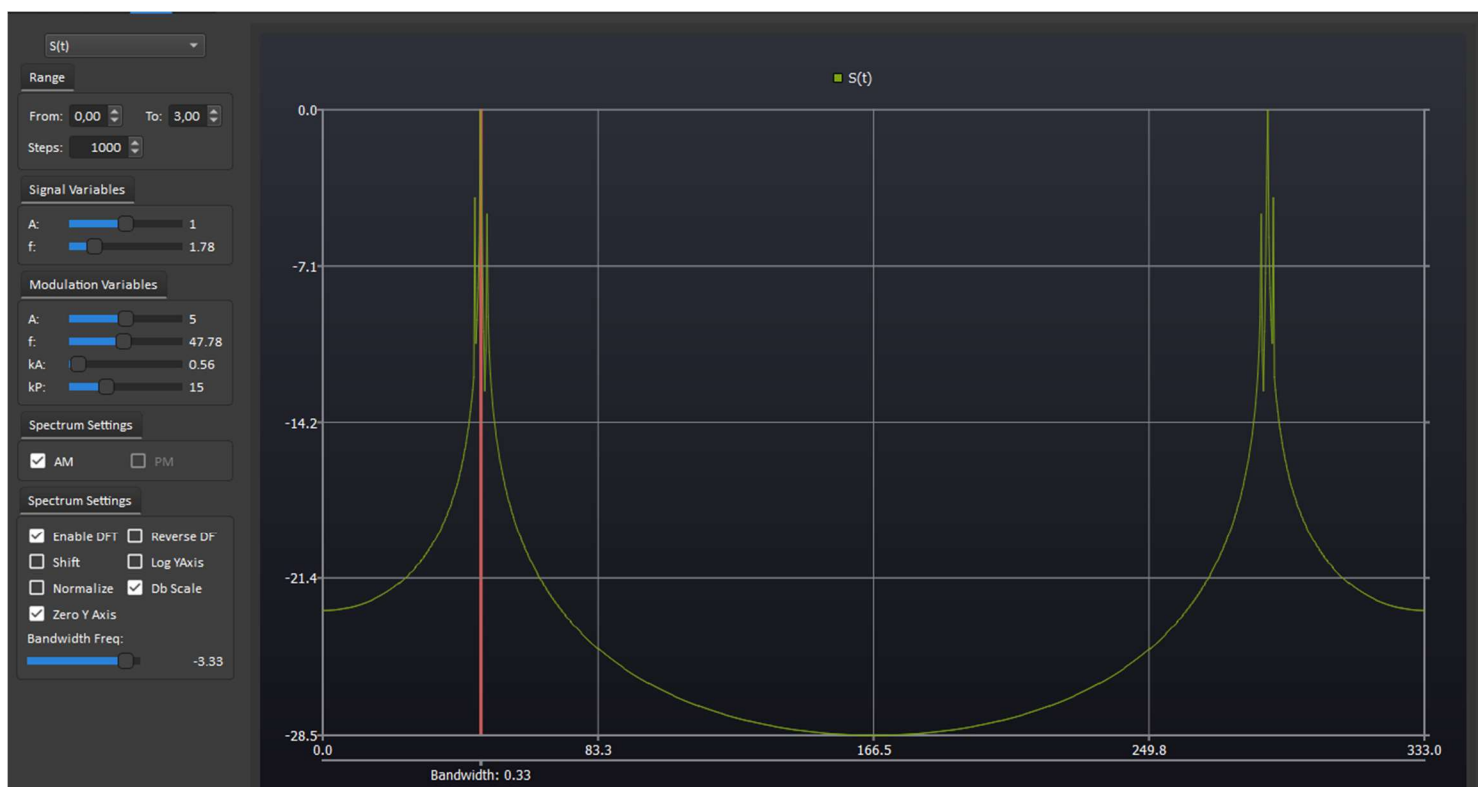
Sygnał informacyjny:



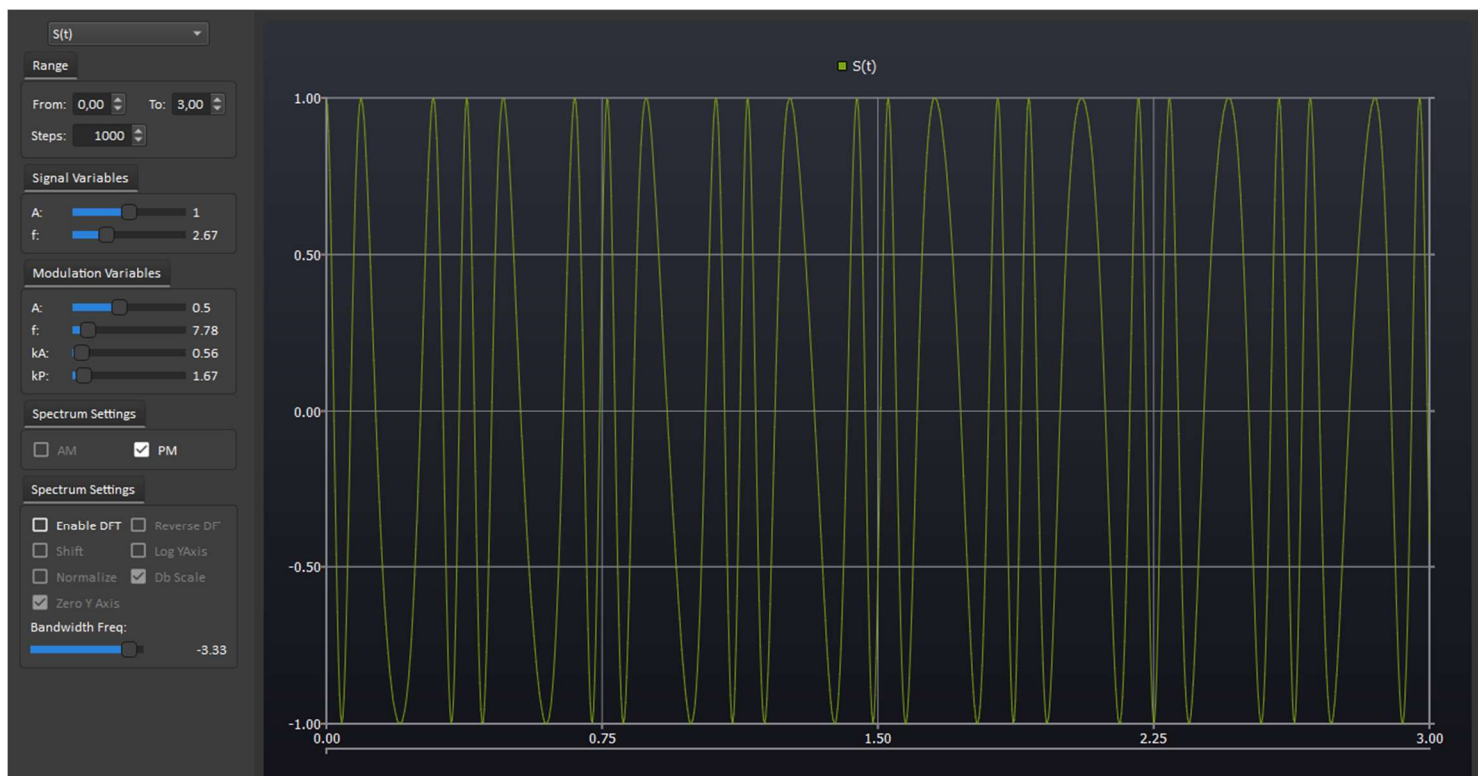
$ZA(t)$, $k_a = 0.56$:



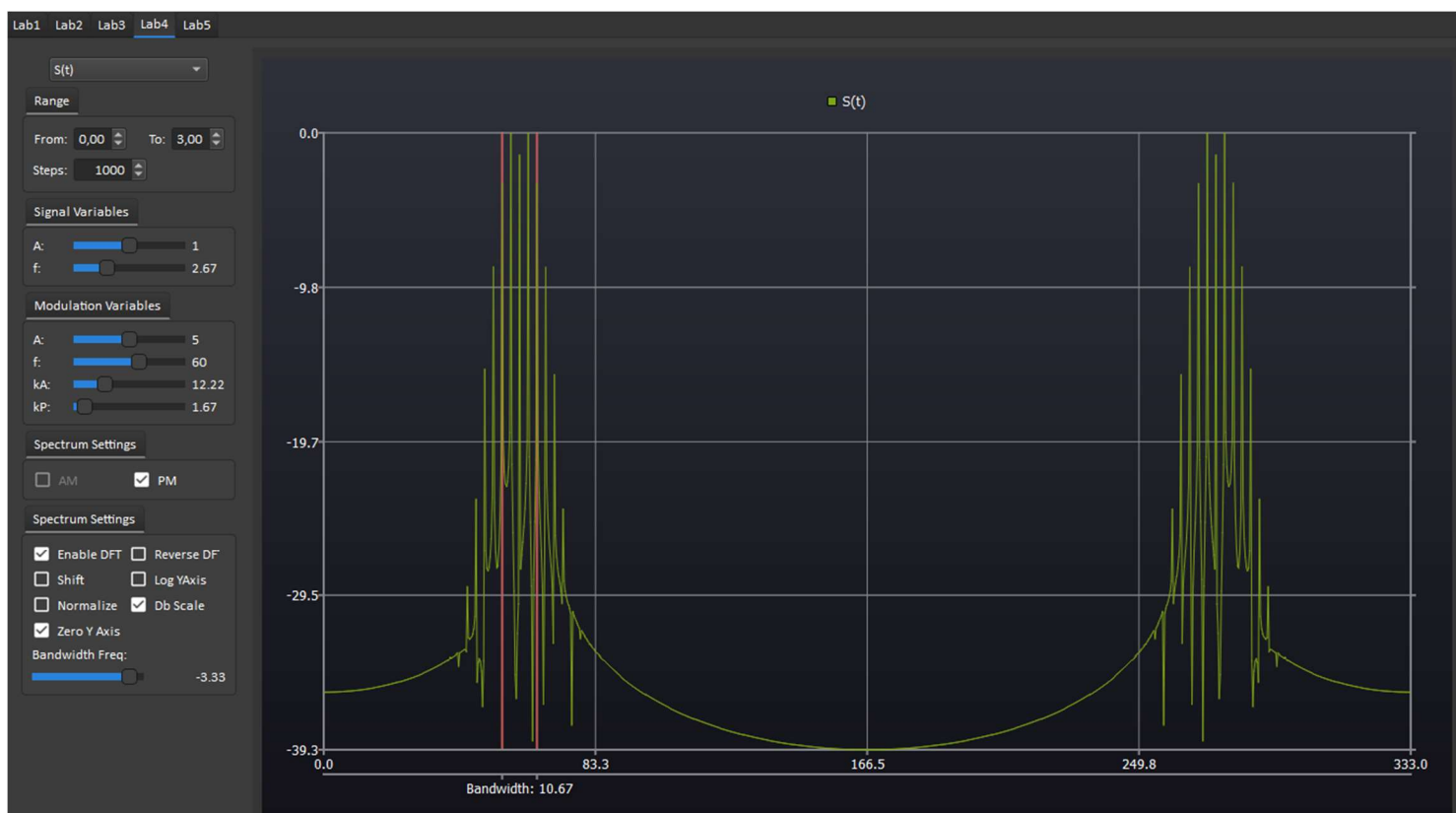
Widmo (szerokość pasma zaznaczona na wykresie):



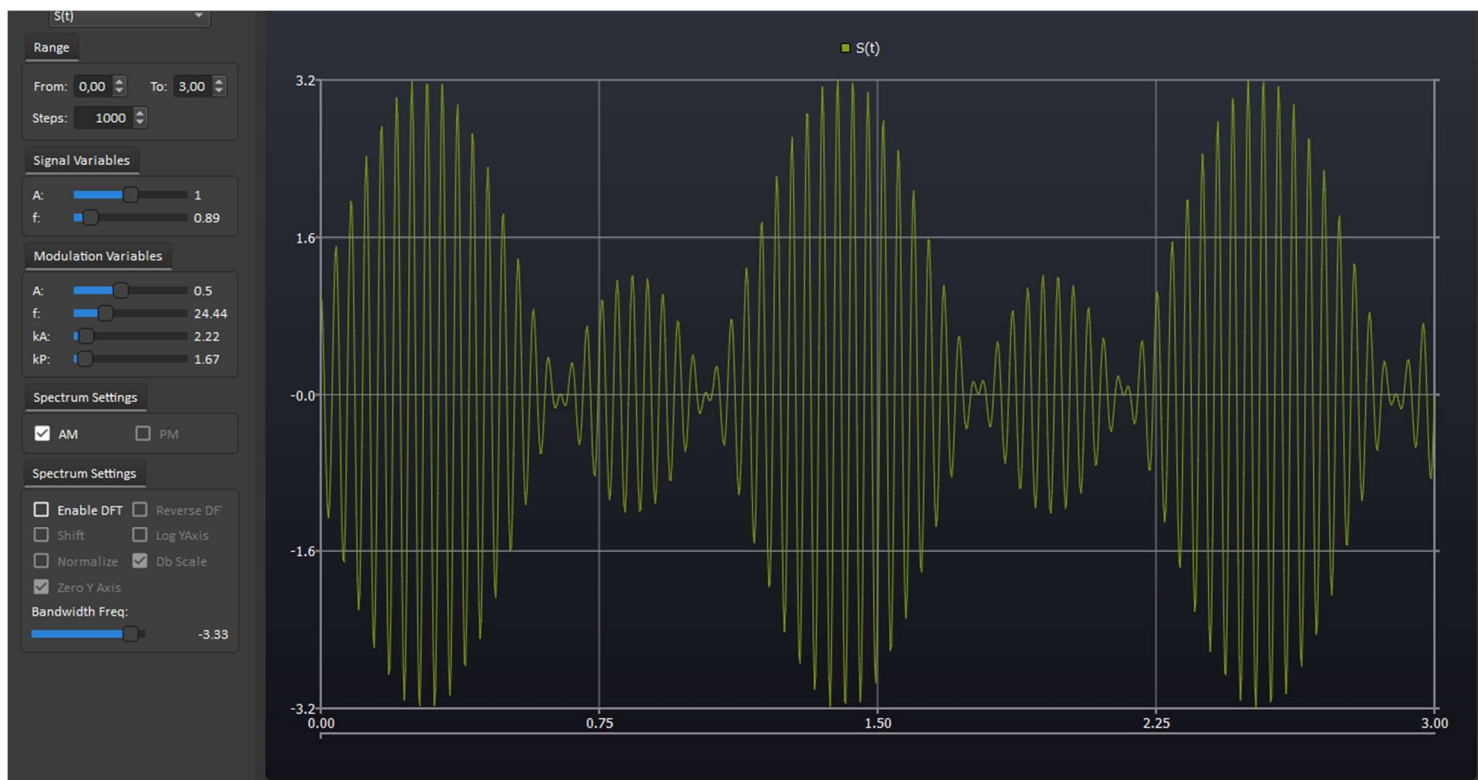
ZP(t), $k_p = 1.67$:



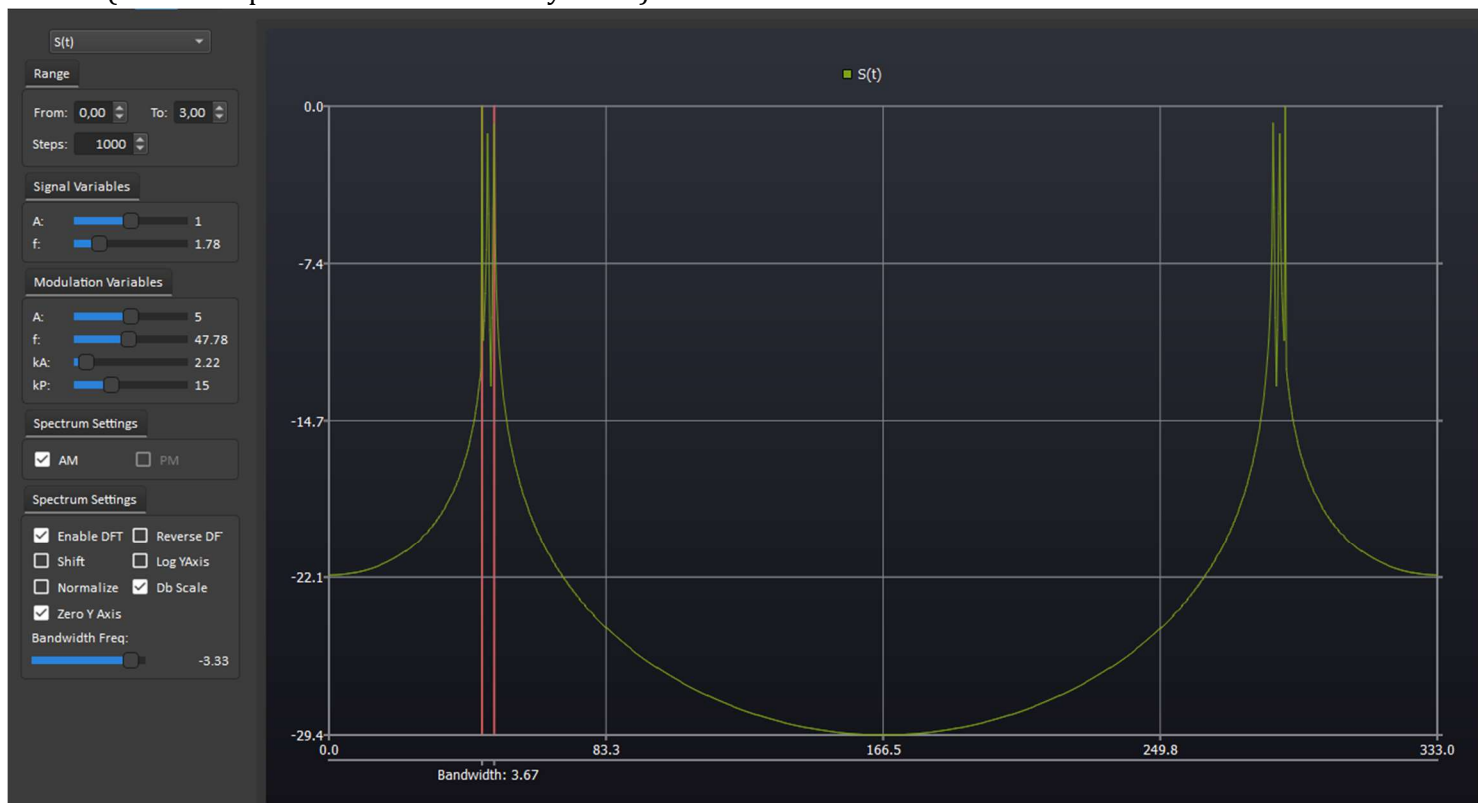
Widmo (szerokość pasma zaznaczona na wykresie):



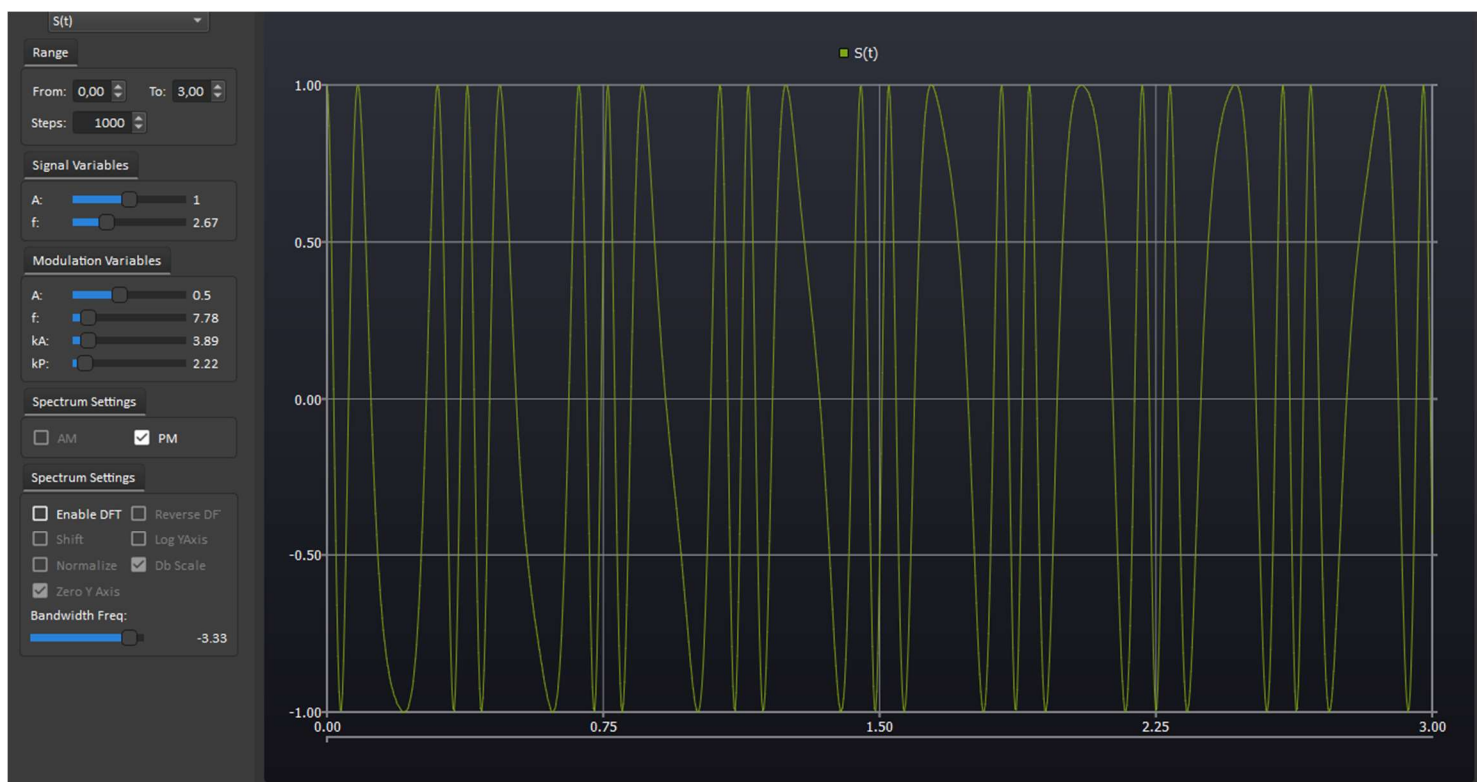
$ZA(t)$, $ka = 2.22$:



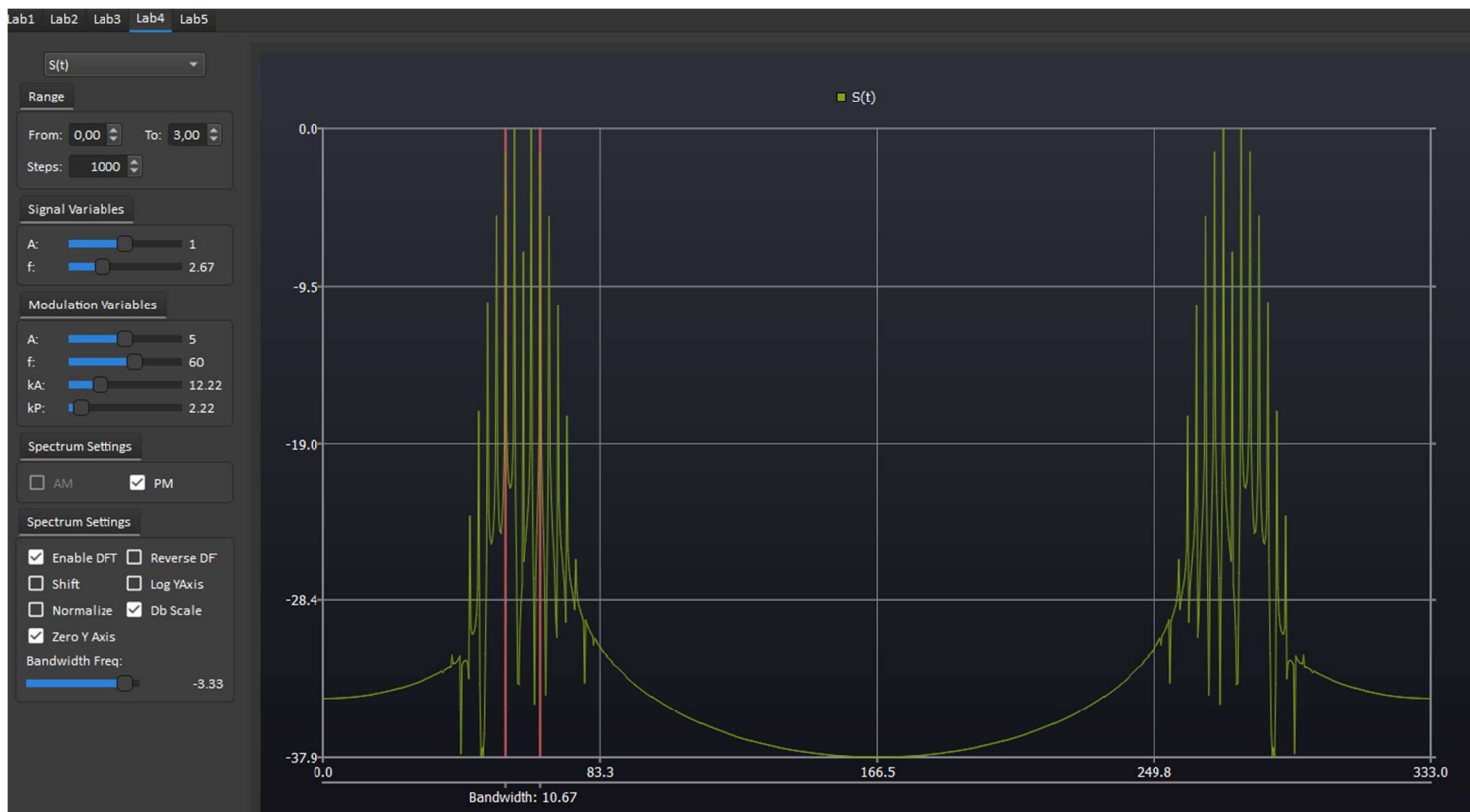
Widmo (szerokość pasma zaznaczona na wykresie):



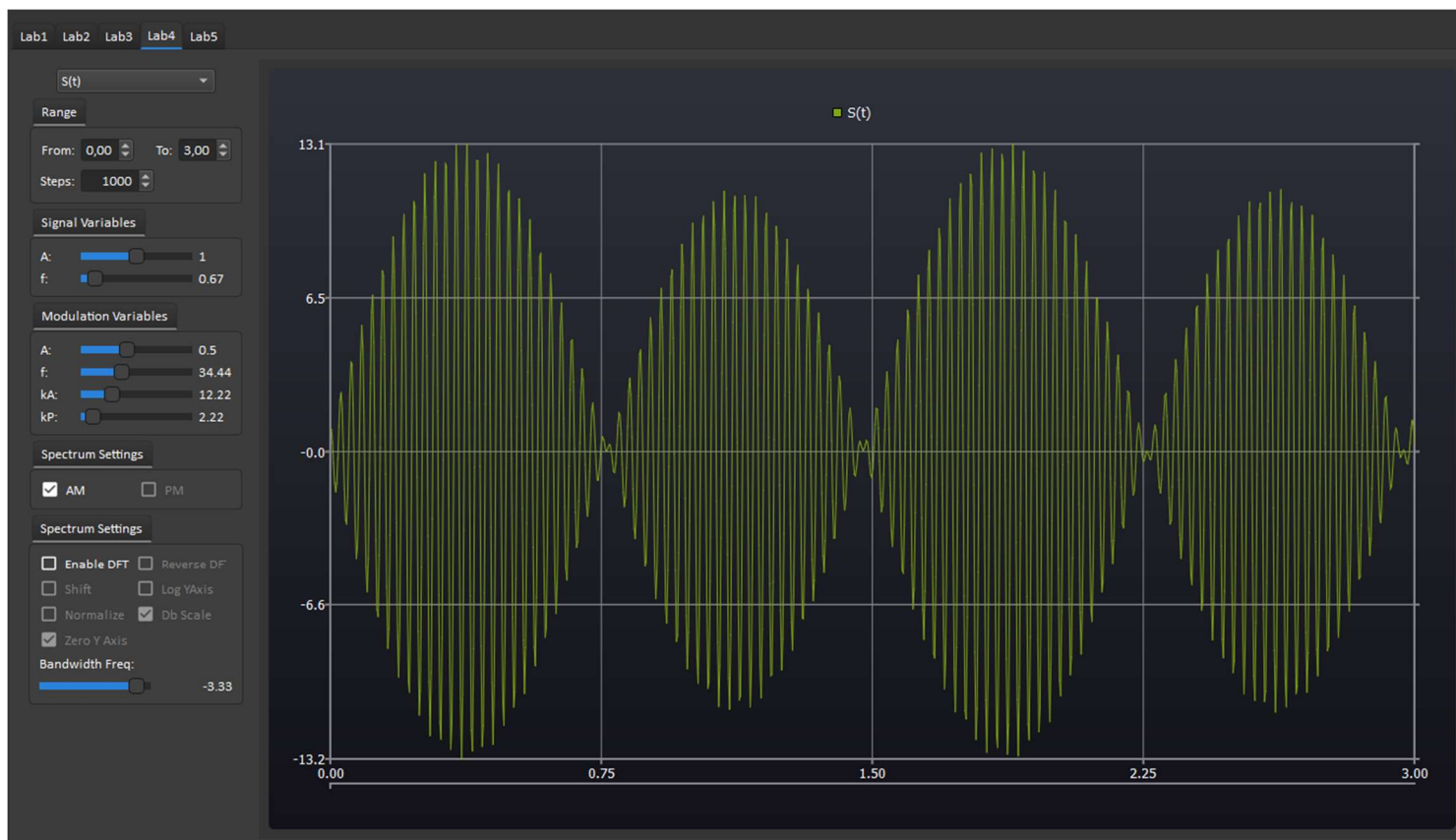
ZP(t), $k_p = 2.22$:



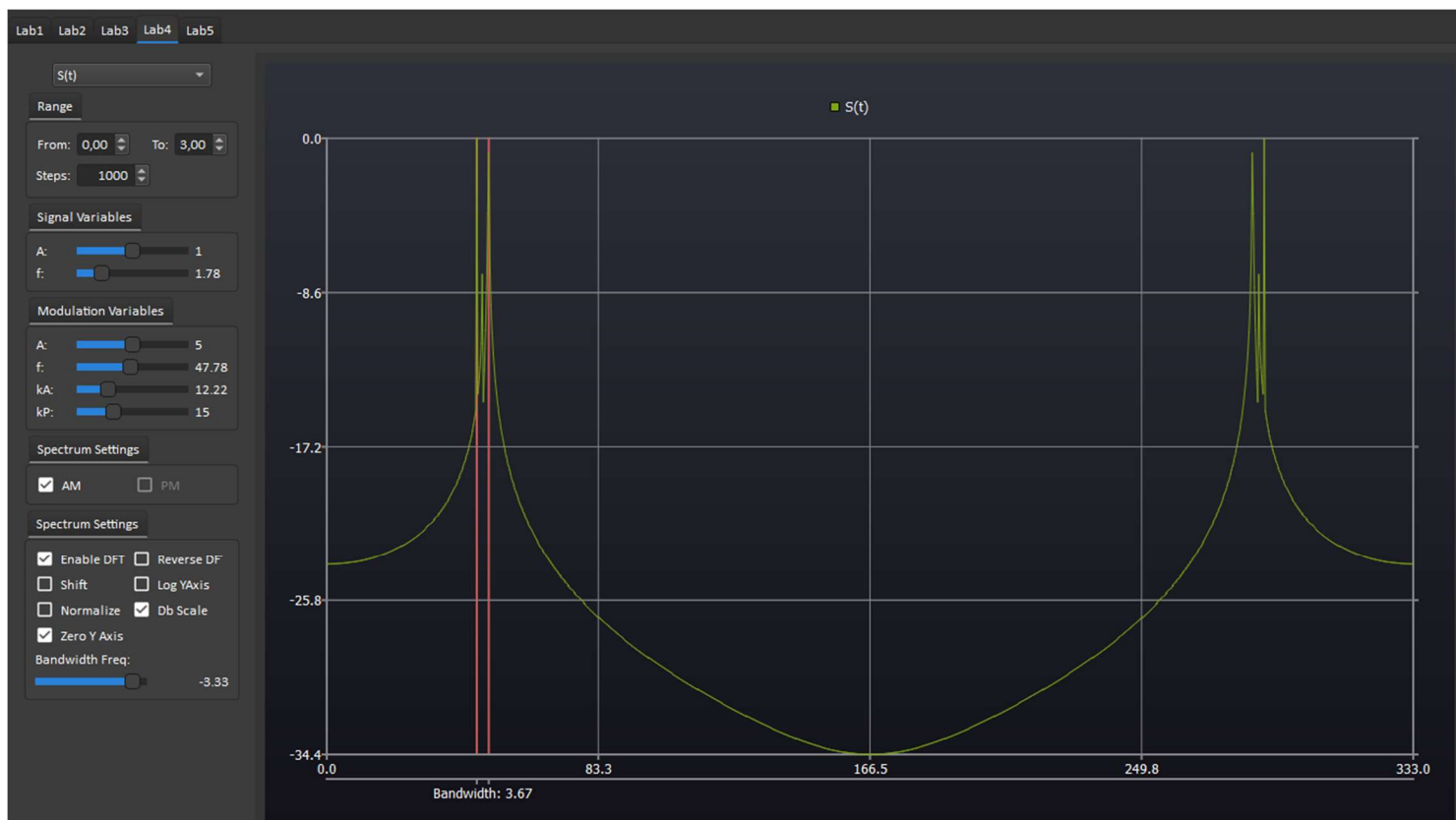
Widmo (szerokość pasma zaznaczona na wykresie):



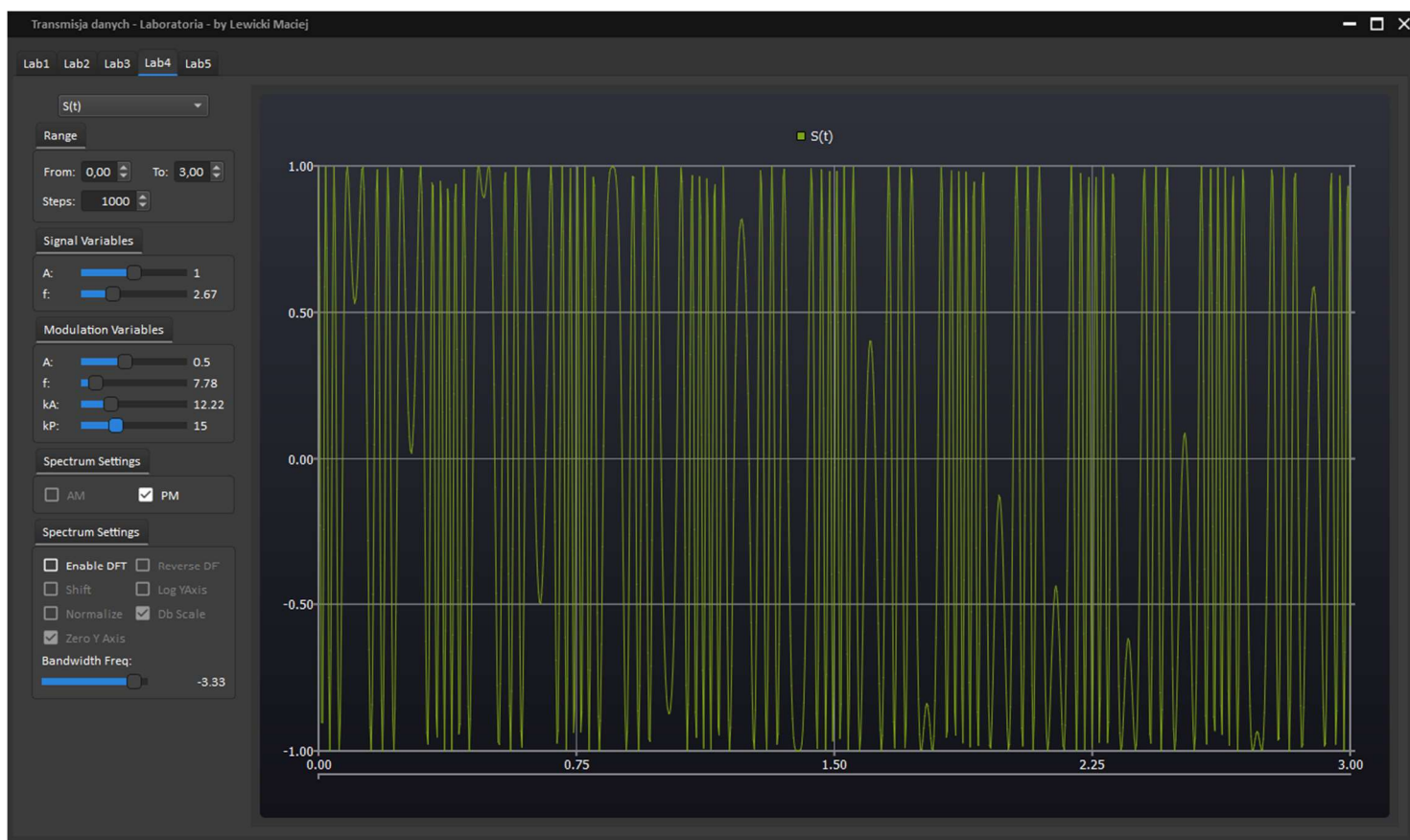
$ZA(t)$, $ka = 12.22$:



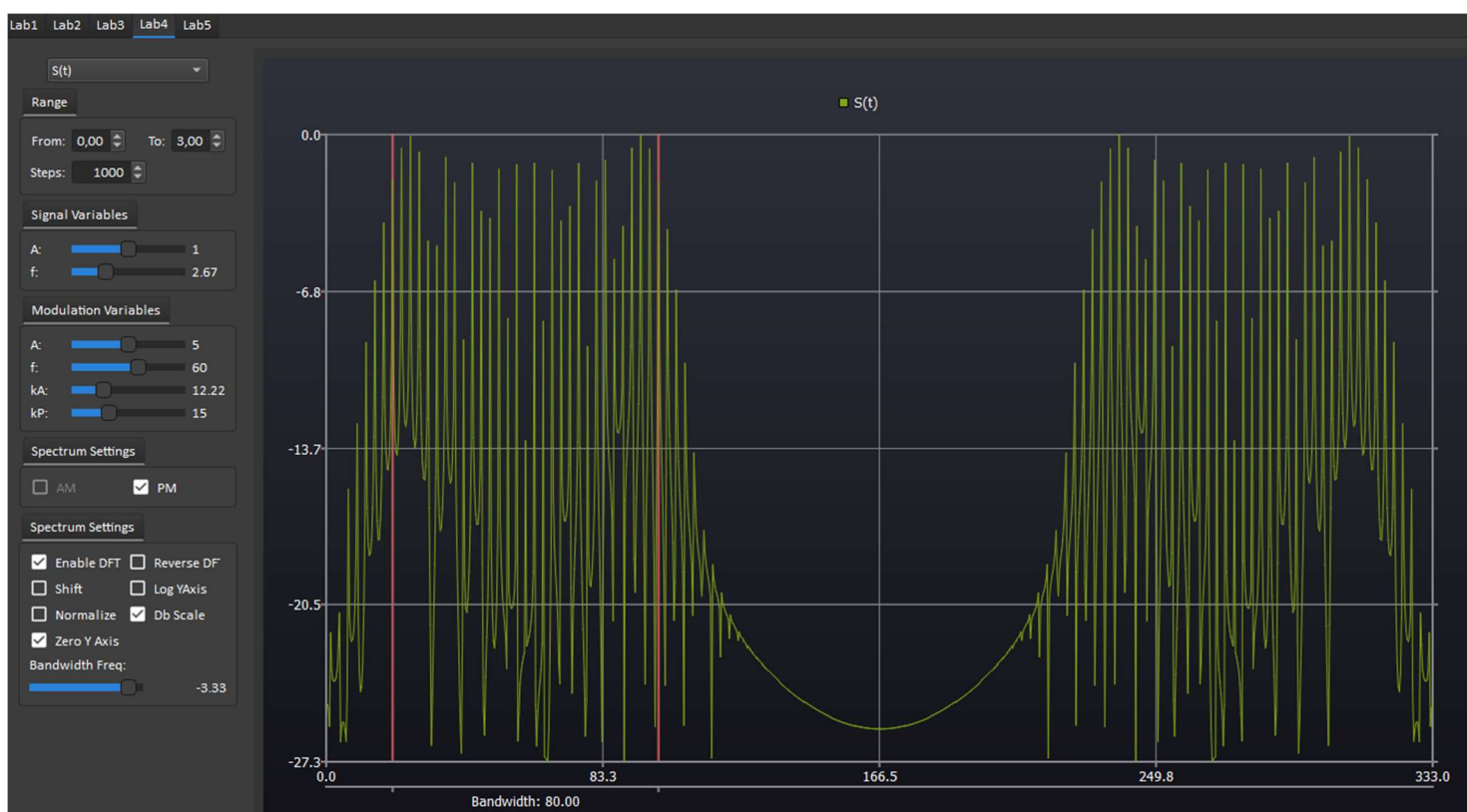
Widmo (szerokość pasma zaznaczona na wykresie):



ZP(t), $k_p = 15$:



Widmo (szerokość pasma zaznaczona na wykresie):



Fragmenty kodu.

Sygnał wejściowy:

```
double Lab4::sFunction(double x)
{
    double amp = static_cast<double>(amplitude->value())/100;
    double freq = static_cast<double>(frequency->value())/100;
    return amp*sin(2*M_PI*freq*x);
}
```

Modulacja amplitudy:

```
double Lab4::modulateAmplitude(std::function<double(double)> foo, double x, double modFreq)
{
    return (1+kA*foo(x))*cos(2*M_PI*modFreq*x);
}
```

Modulacja fazy:

```
double Lab4::modulatePhase(std::function<double(double)> foo, double x, double modFreq)
{
    return cos(2*M_PI*modFreq*x+kP*foo(x));
}
```

Obliczanie szerokości pasma:

```
QPair<double, double> Lab4::calculateBandwidth(QVector<QPointF> vec, double dec)
{
    double min = 0;
    double max = 0;
    bool gotMin = false;
    for(int i=0; i<vec.length()/2; i++)
    {
        if(vec.at(i).y()>dec)
        {
            if(gotMin)
                max = vec.at(i).x();
            else
            {
                min = vec.at(i).x();
                gotMin = true;
            }
        }
    }
    if(min==0)
        max=0;
    if(max==0)
        min=0;
    return QPair<double, double>(min, max);
}
```

Reszta funkcjonalności podobnie jak w poprzednich zadaniach.