



Projekt GUI – modulacja, demodulacja AM i FM

Identyfikacja układów mechatronicznych

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

12.05.2022



Jakub Nowak
Maciej Mazurkiewicz



Plan prezentacji

- 1 Teoria – jak to działa? (Modulacja)
- 2 Transformata Hilberta – na co pozwala?
- 3 Teoria – jak to działa? (Demodulacja)
- 4 Implementacja w MATLAB
- 5 Graphical User Interface (GUI) – jak to zrobić?
- 6 Prezentacja działania naszego rozwiązania
- 7 Podsumowanie

Teoria – modulacja

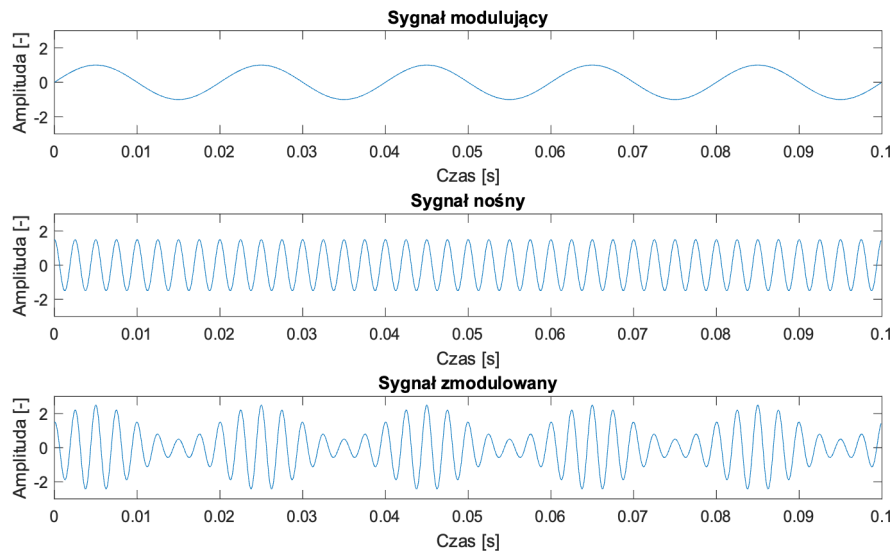
AM – Amplitude Modulation

Sygnal nośny: $c(t) = A_c \cos(2\pi f_c t)$

Sygnal modulujący: $m(t) = A_m * x(t)$

Sygnal zmodulowany:

$$s(t) = (A_c + A_m * x(t)) * \cos(2\pi f_c t)$$



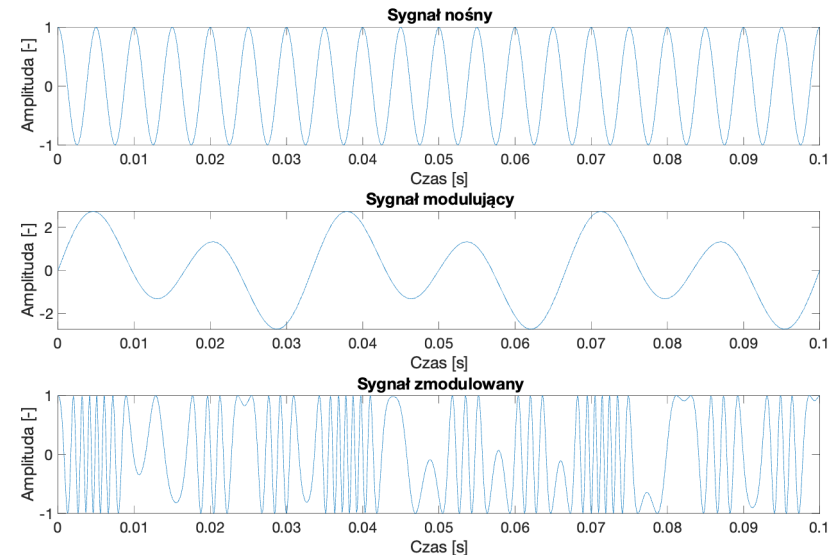
FM – Frequency Modulation

Sygnal nośny: $c(t) = A_c \cos \phi(t)$

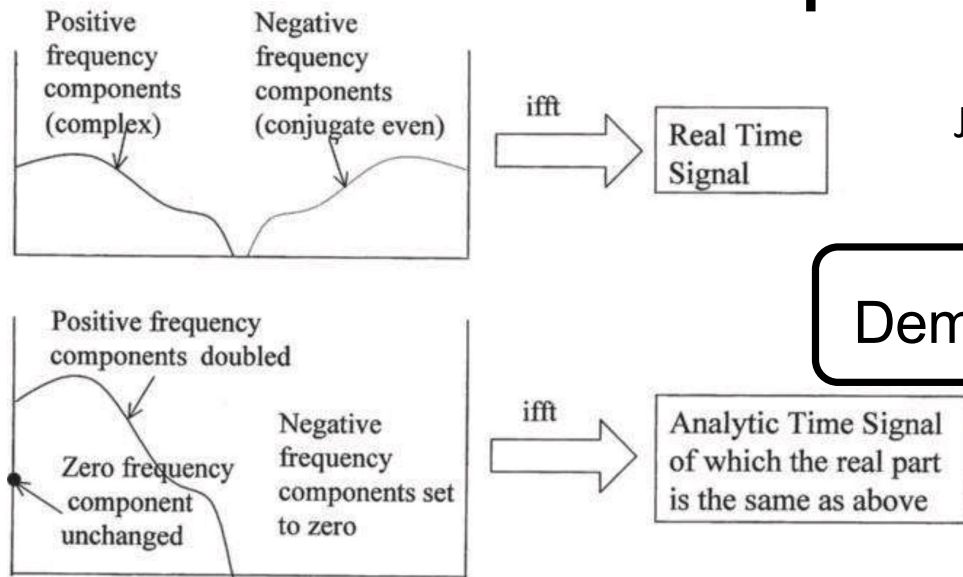
Sygnal modulujący: $f_m(t) = A_m * x(t)$

Sygnal zmodulowany:

$$f_{FM}(t) = A \times \cos \left[f_c t + k_f \int_0^t f_m(\tau) d\tau \right]$$



Transformata Hilberta – na co pozwala?



Jest ona definiowana w następujący sposób:

$$\hat{s}(\tau) = \frac{1}{\pi} \int_{-\infty}^{\infty} s(t) \frac{1}{\tau - t} dt$$

Demodulacja FM

Demodulacja AM

W efekcie zgodnie ze **wzorem Eulera**: $e^{i\phi} = \cos\phi + i\sin\phi$, otrzymujemy część rzeczywistą analogiczną do oryginalnego sygnału i część ujemną. Korzystając teraz z nowo stworzonego **sygnału analitycznego** można wyznaczyć **fazę** i **obwiednie** sygnału.

Teoria – demodulacja

AM – Amplitude Modulation

Demodulacja synchroniczna

$$s(t) = [A_c + x(t)] * \cos(\omega t) * \cos(\omega t)$$

Demodulacja Hilbert – prościej!

$$a(t) = \sqrt{x^2(t) + H[x(t)]^2}$$

Moduł sygnału analitycznego – obwiednia,
pozwala uzyskać sygnał zdemodulowany

FM – Frequency Modulation

Demodulacja Hilbert

Faza sygnału analitycznego: $\Phi(t) = \arctg\left(\frac{H[x(t)]}{x(t)}\right)$

Częstotliwość chwilowa: $\omega(t) = \frac{d\Phi(t)}{dt}$

Więc zgodnie z:

$$f_{FM}(t) = A \times \cos \left[f_c t + k_f \int_0^t f_m(\tau) d\tau \right]$$

$$\omega(t) = f_c + k_f f_m$$

Ostatecznie:

$$f_m = \frac{1}{2\pi k_f} \times \omega(t) - f_c$$

Ponieważ jest to stałe przesunięcie można się
pozybć bez znajomości dokładnej wartości

Implementacja – jak to wygląda?

AM – Amplitude Modulation

Modulacja

```
if(app.CosinewaveButton.Value == 1)
    for i=1:length(app.t)
        app.fmod(1,i) = (app.x(i)+max(app.x)/app.u)*cos(2*pi*app.fc*app.t(i));
    end
    app.fmodf=abs(fft(app.fmod));
elseif(app.SinewaveButton.Value == 1)
    for i=1:length(app.t)
        app.fmod(1,i) = (app.x(i)+max(app.x)/app.u)*sin(2*pi*app.fc*app.t(i));
    end
    app.fmodf=abs(fft(app.fmod));
end
```

Demodulacja

```
function demodulate_signalAM(app)
    xh_own = fft(app.fmod);
    Nfft=length(xh_own);
    xh_own(Nfft/2:end) = 0;
    xh_own(1:Nfft/2) = xh_own(1:Nfft/2).*2;
    xh_own = ifft(xh_own);
    app.demod=abs(xh_own);
    app.demod=app.demod-mean(app.demod);
    app.demodf=fft(app.demod);
    app.demodulation_flag=1;
end
```

FM – Frequency Modulation

Modulacja

```
if(app.CosinewaveButton.Value == 1)
    for i=1:length(app.t)
        app.fmod(1,i) =cos(2*pi*app.fc*app.t(i)+ app.kf*trapz(app.x(1:i)));
    end
    app.fmodf=abs(fft(app.fmod));
elseif(app.SinewaveButton.Value == 1)
    for i=1:length(app.t)
        app.fmod(1,i) =sin(2*pi*app.fc*app.t(i)+ app.kf*trapz(app.x(1:i)));
    end
    app.fmodf=abs(fft(app.fmod));
end
```

Demodulacja

```
function demodulate_signal(app)
    xh_own = fft(app.fmod);
    Nfft=length(xh_own);
    xh_own(Nfft/2:end) = 0;
    xh_own(1:Nfft/2) = xh_own(1:Nfft/2).*2;
    xh_own = ifft(xh_own);
    a = unwrap(angle(xh_own));
    app.demod=diff(a)/(2*pi/app.fs);
    app.demod = app.demod - mean(app.demod);
    app.demodf=fft(app.demod);
    app.demodulation_flag=1;
end
```

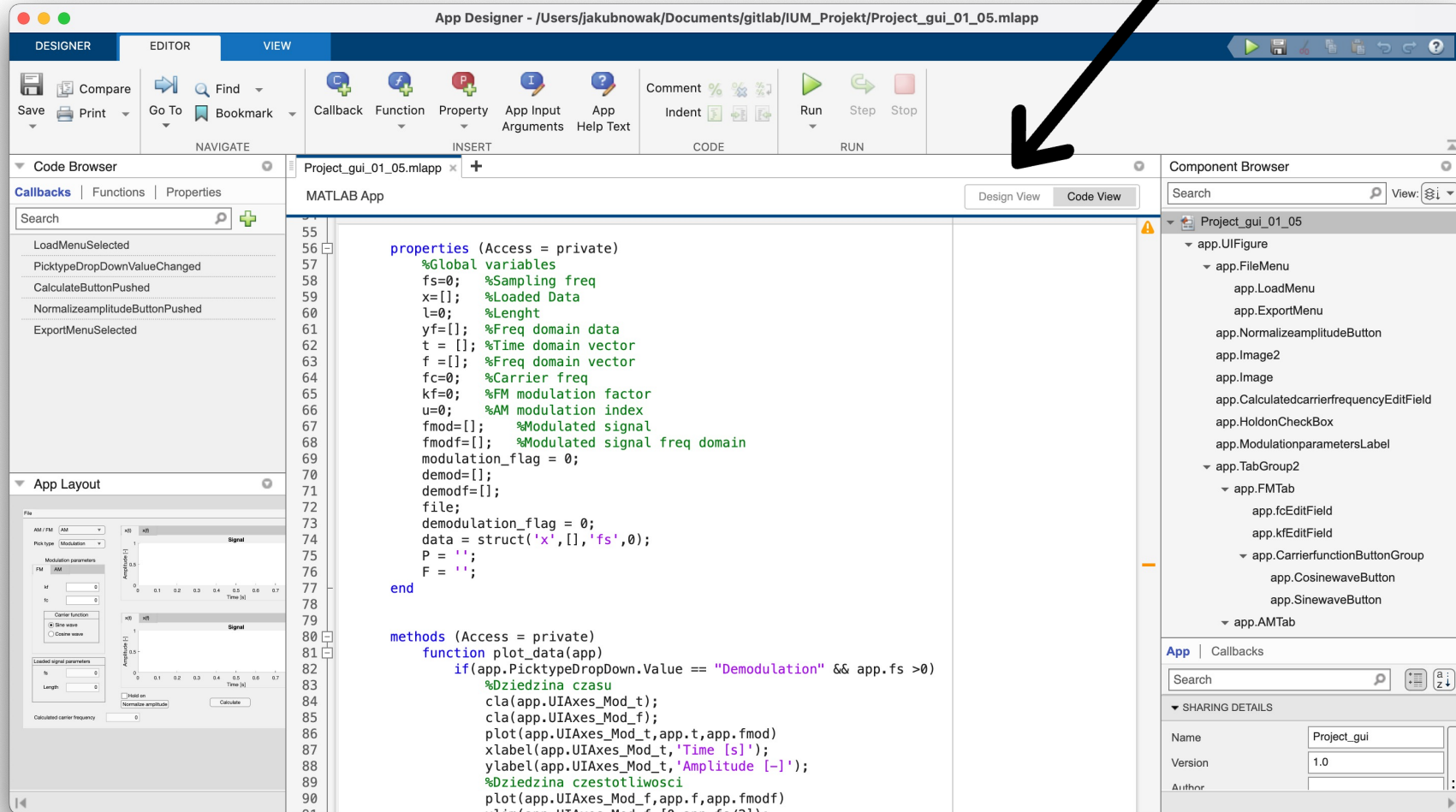
Całka
numeryczna

Hilbert!

Odjęcie średniej

GUI – jak zrobić?

Kod



Prezentacja rozwiązania

Import i eksport

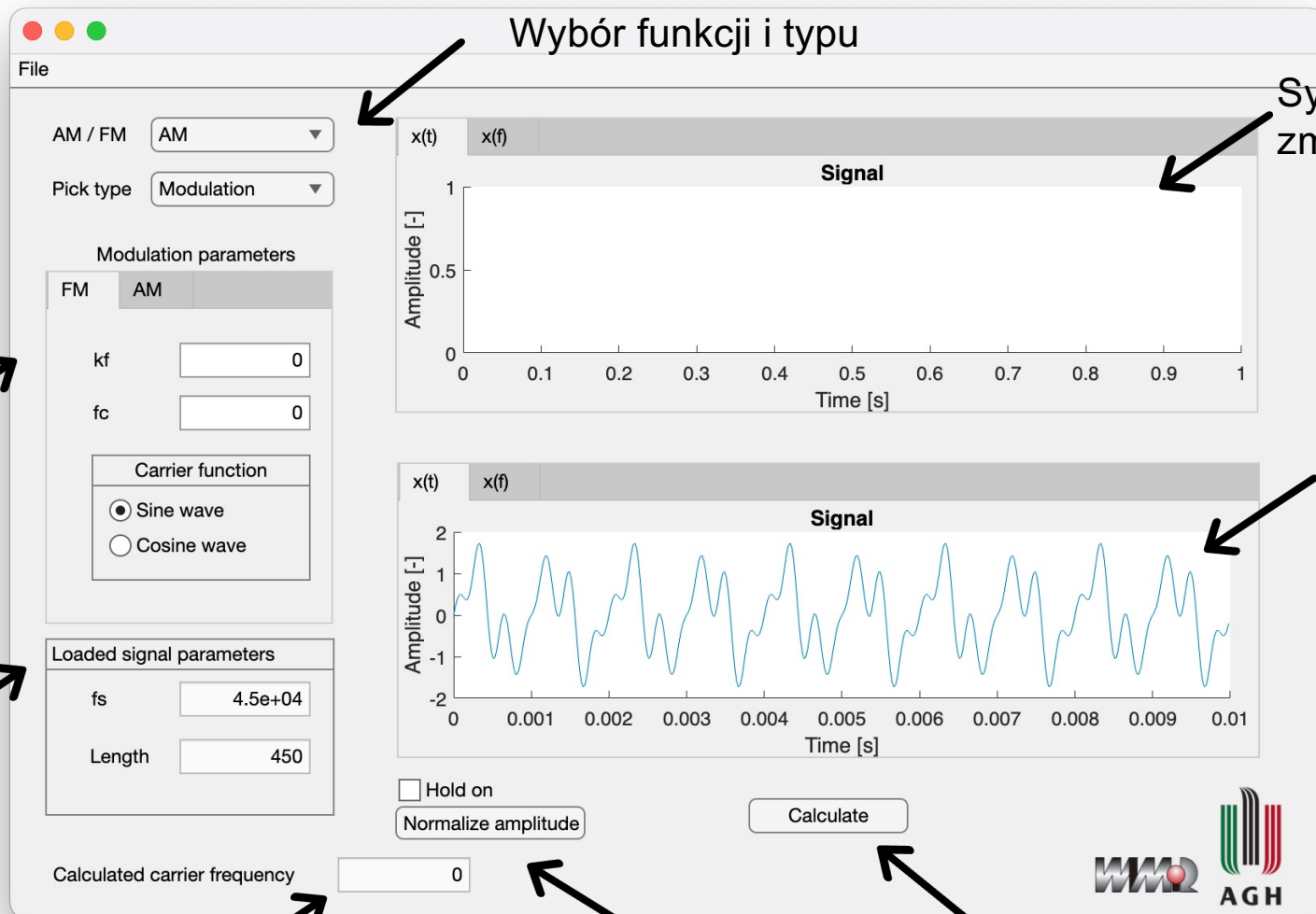
Parametry modulacji

Parametry wczytanego sygnału

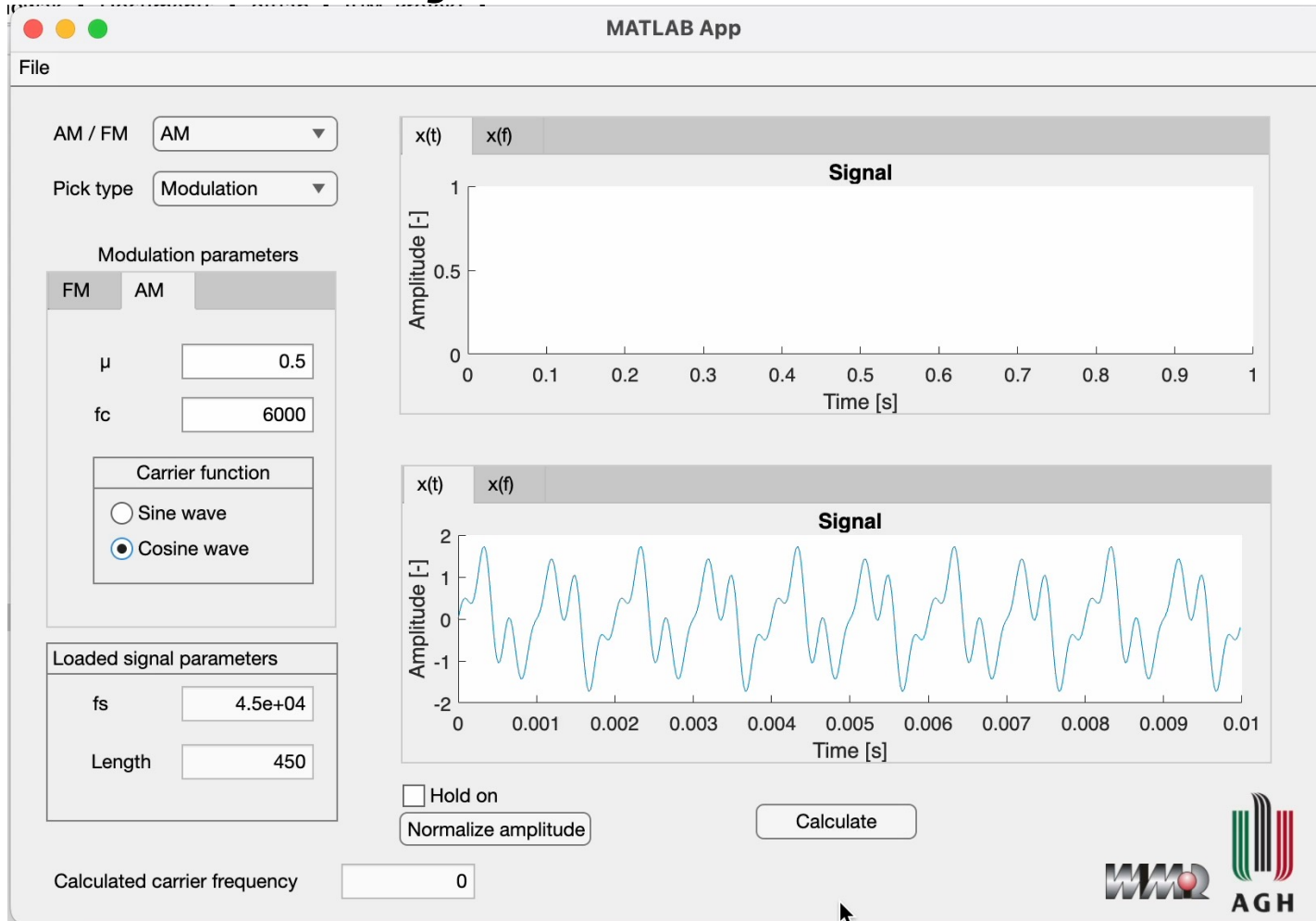
Obliczana z fft
częstotliwość
przenoszenia

Normalizacja amplitudy i opcja
nakładania się sygnałów

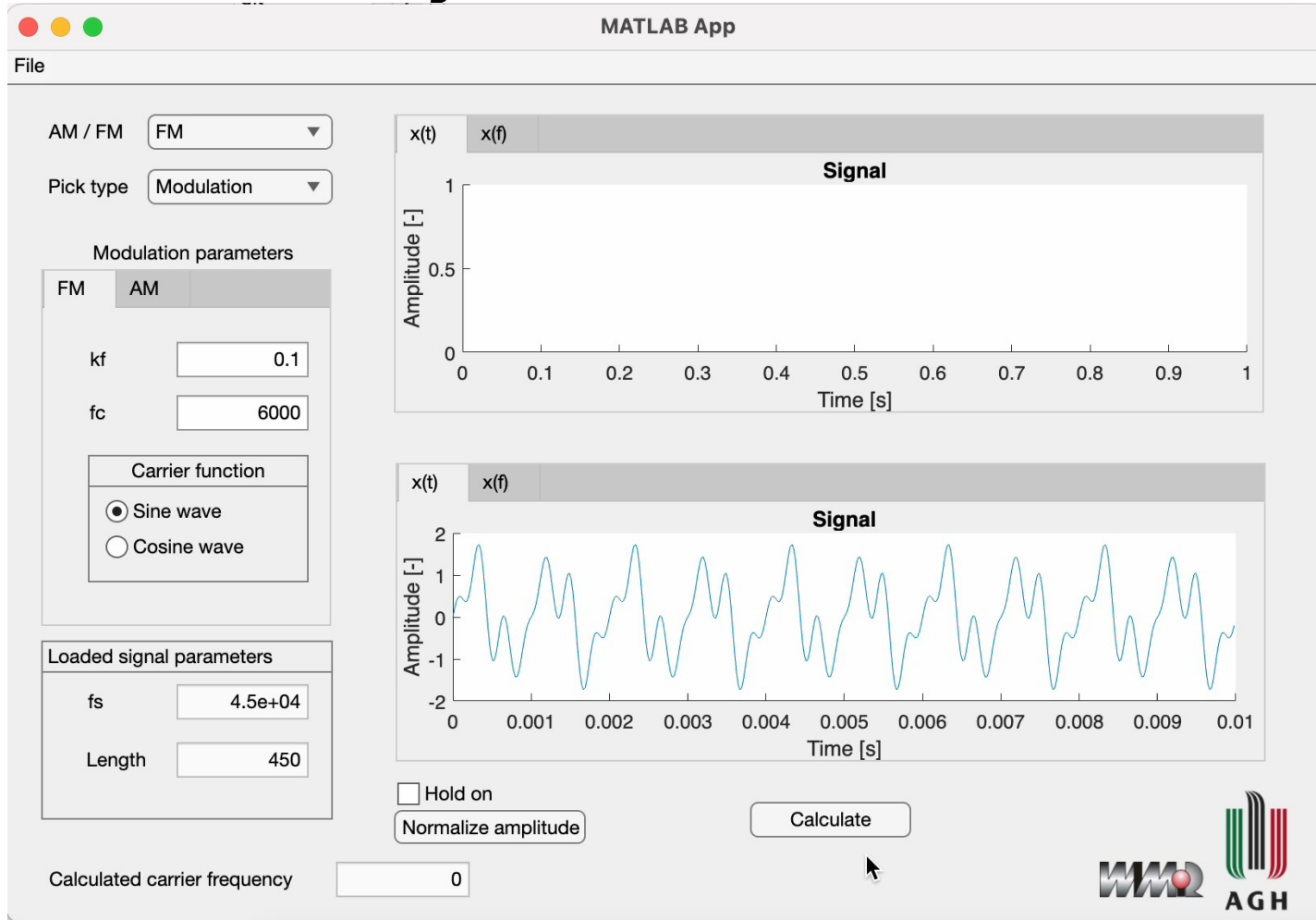
Uruchamianie
obliczeń



Prezentacja działania GUI - AM



Prezentacja działania GUI - FM



Podsumowanie

- Demodulacja AM i FM z wykorzystaniem Hilberta - nie potrzebna znajomość częstotliwości przenoszenia
- Możliwość modulacji dowolnego sygnału
- GUI umożliwia eksport wyników do pliku .mat
- App Designer – proste i szybkie narzędzie do tworzenia GUI w MATLAB

Dziękujemy za uwagę!