

WIZUALIZACJA PARAMETRÓW ANTEN SYMULOWANYCH PROGRAMEM NEC-2

Bartosz Chaber, Jacek Starzyński

Politechnika Warszawska, Wydział Elektryczny

Streszczenie. Artykuł prezentuje informacje dotyczące wizualizacji danych wyjściowych symulacji programem NEC-2. Autorzy opisują własne narzędzia służące do generowania wizualizacji parametrów modelowanych anten takich jak charakterystyka promieniowania, impedancja falowa anteny, parametr S_{11} .

Słowa kluczowe: anteny nec2 charakterystyka promieniowania impedancja falowa S_{11}

VISUALIZATION OF ANTENNA'S PARAMETERS SIMULATED WITH NEC-2

Abstract. Article presents information about visualization of data output from NEC-2. Authors describes their own tools used for visualization of antenna's parameters like radiation pattern, wave impedance, S_{11} parameter.

Keywords: antennas nec2 radiation pattern wave impedance S_{11}

Wstęp

Zagadnienie wizualizacji parametrów anten jest bardzo istotne z perspektywy osoby projektującej anteny, dlatego oprogramowanie służące do ich modelowania powinno udostępniać dane z symulacji w przejrzystej, ułatwiającej analizę formie. Jednym z najbardziej rozpowszechnionych programów do modelowania anten jest NEC-2, który udostępnia jedynie interfejs tekstowy. Niniejszy artykuł opisuje prace nad zestawem narzędzi służącym do wizualizacji wyników symulacji wygenerowanych przez program NEC-2.

1. Program NEC-2

NEC-2 to program napisany w latach 70 w języku Fortran, którego format plików wejściowych przypomina format kart perforowanych. W pliku wejściowym zapisane są komendy sterujące programem, definiujące geometrię modelu oraz konfiguruje parametry, które mają zostać zapisane w pliku wynikowym symulacji [1]. Plik wynikowy jest plikiem tekstowym, który zawiera wszystkie dane potrzebne do wizualizacji obliczonych parametrów anteny.

Dzięki temu, że plik wynikowy jest czytelny przez człowieka możliwe jest łatwe wyodrębnienie z niego danych służących do wyznaczenia charakterystycznych wartości modelowanej anteny. Otwarte źródło programu przyczyniło się do powstania wielu aplikacji bazujących na kodzie NEC-2. Wśród nich znajdują się implementacje NEC-2 w językach C czy C++. Są to programy działające w trybie graficznym, rozbudowane o komponenty wizualizacyjne pozwalające na generowanie wykresów oraz wyświetlanie trójwymiarowych powierzchni. Mimo to, prezentowane przez te programy ilustracje parametrów anteny często nie są wystarczająco przejrzyste bądź dokładne. Dodatkowo, część z tych programów generuje wyniki symulacji w locie, nie pozwalając na ich zapisanie i późniejszą analizę. Te przesłanki spowodowały podjęcie przez autorów prac nad narzędziami do wizualizacji wyników programu NEC-2.

2. Charakterystyka promieniowania

Podstawowym parametrem anteny, który pozwala szybko ocenić jej przydatność jest charakterystyka promieniowania. Często przedstawia się ją w postaci dwóch dwuwymiarowych wykresów w układzie biegunowym lub w postaci trójwymiarowej bryły, której kształt informuje w którym kierunku antena promieniuje najbardziej, a w którym najmniej.

Charakterystyka promieniowania pokazuje, jaka moc została wypromieniowana w określonym kierunku, w porównaniu do mocy wypromieniowanej, przy takim samym zasilaniu, przez antenę izotropową (idealną antenę, która promieniuje jednakowo

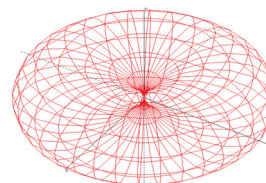
we wszystkich kierunkach). Stosunek tych dwóch mocy przez jednostkę powierzchni określany jest jako *wzmocnienie* i charakteryzuje się wzorem:

$$G = \frac{(P/S)_{ant}}{(P/S)_{iso}} \quad (1)$$

gdzie: $(P/S)_{ant}$ – moc wypromieniowana przez jednostkę powierzchni przez badaną antenę, $(P/S)_{iso}$ – moc wypromieniowana przez jednostkę powierzchni przez antenę izotropową.

Najłatwiejszym do analizy przykładem jest charakterystyka promieniowania dipola. Jest to na tyle prosta antena, że możliwe jest analityczne wyznaczenie jej charakterystyki promieniowania. Dipol promieniuje najmocniej w kierunku prostym do swojej osi, natomiast nie promieniuje wzdłuż niej. Jego charakterystyka promieniowania ma kształt toroidu.

Na rysunku 1 przedstawiono wizualizację charakterystyki z programu Xnecview. Rysunek 2 pokazuje natomiast wizualizację wykonaną za pomocą autorskiego skryptu w języku Python.



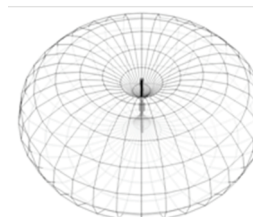
Rys. 1. Charakterystyka promieniowania dipola półfalowego (Xnecview)

Skrypt polega na wyodrębnieniu z pliku wynikowego NEC-2 charakterystyki promieniowania i zapisania go jako plik z trójwymiarową siatką powierzchniową (w formacie Wavefront OBJ) który może zostać potem zwizualizowany. Program NEC-2 zapisuje charakterystykę promieniowania za pomocą wartości wzmocnienia względem kątów θ, ϕ .

Do celów prezentacji wzmocnienie jest dodatkowo przeliczone za pomocą jednej z czterech formuł skalujących [5]. W prezentowanym skrypcie wykorzystywany jest styl *Linear Power Scaling* definiujący skalowanie wzmocnienia wzorem:

$$G_{scaled} = e^{\log_{10} \frac{G}{10}} \quad (2)$$

gdzie: G – współczynnik wzmocnienia przed skalowaniem.



Rys. 2. Charakterystyka promieniowania dipola półfalowego wyodrębniona z pliku wynikowego NEC-2

Podczas modelowania anteny operator powinien móc szybko ocenić, w którym kierunku najsilniej promieniuje zaprojektowana antena. W narzędziu przedstawionym przez autorów możliwe jest to dzięki wskazówkom wizualnym, polegających na zmianie koloru lub odcienia wraz z odległością elementu charakterystyki od obserwatora. Przesłonięte elementy wzorca promieniowania są prezentowane jako półprzezroczyste, co ułatwia trójwymiarowe postrzeganie tej charakterystyki. Dodatkowo naniesienie na obraz struktury anteny pomaga odnieść jej charakterystykę promieniowania do układu modelu.

3. Natężenie pola elektrycznego i magnetycznego

Podczas projektowania anten służących do testów narażeniowych ważnym parametrem świadczącym o przydatności anteny jest charakter pola elektromagnetycznego, które jest generowane w strefie bliskiej anteny [4]. Program NEC-2 pozwala obliczenie wartości bliskiego pola elektrycznego bądź magnetycznego anteny w pewnej siatce punktów, zapisując wartości pola w postaci wektora o trzech składowych zespolonych. Aby móc narysować wykres impedancji falowej konieczne jest obliczenie długości wspomnianego wektora za pomocą wzoru:

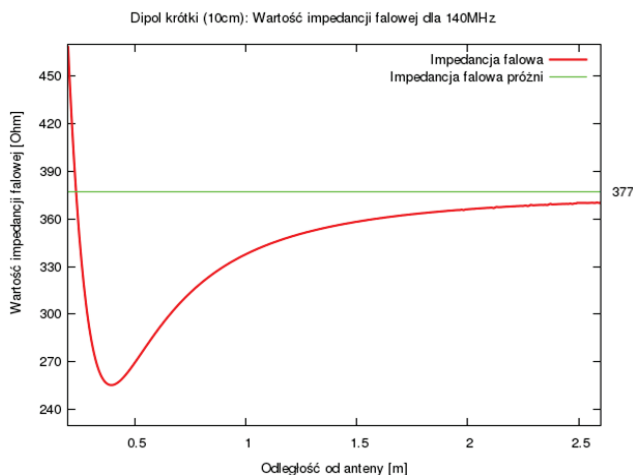
$$|v| = \sqrt{v_x \cdot v_x^* + v_y \cdot v_y^* + v_z \cdot v_z^*} \quad (3)$$

gdzie: v^* – wektor zespolony, sprzężony z v .

Do celów prezentacyjnych został przygotowany skrypt napisany w języku Python, który konfiguruje symulację NEC w ten sposób, że oblicza on wartości pola E i H w punktach leżących na półprostej biegnącej od osi dipola krótkiego wzdłuż osi Y. Następnie, skrypt oblicza za pomocą wzoru (3) długość każdego z wektorów pola i zapisuje je wraz z wartością impedancji falowej pola elektromagnetycznego, danej wzorem:

$$Z = \frac{E}{H} \quad (4)$$

gdzie: Z – impedancja falowa, E – natężenie pola elektrycznego, H – natężenie pola magnetycznego.



Rys. 3. Impedancja falowa dipola krótkiego, obliczona w strefie bliskiej przy częstotliwości 140MHz

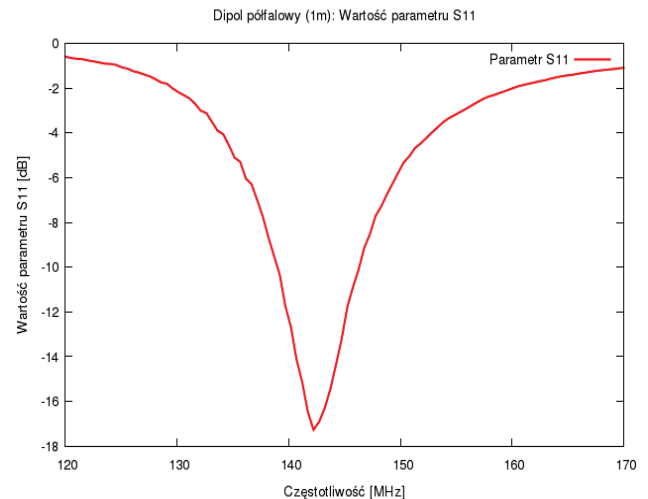
Analizując wykres impedancji falowej w funkcji odległości od anteny operator programu może łatwo określić charakter pola w strefie bliskiej anteny. Przykładowy wykres wygenerowany dla symulacji dipola krótkiego został zamieszczony na rysunku 3. Jak widać z jego przebiegu około 3m od anteny można spodziewać się tego, że impedancja falowa osiągnie wartość impedancji falowej próżni, czyli 377Ω, co pozwala traktować pole w odległości powyżej 3m jako pole dalekie.

4. Parametr S_{11}

Parametr S_{11} określa miarę dopasowania anteny do źródła zasilania. Znając informacje o impedancji wejściowej anteny oraz impedancji źródła zasilania można go obliczyć stosując wzór 5 (za [2]).

$$S_{11} = |\Gamma| = \left| \frac{Z - Z_{ref}}{Z + Z_{ref}} \right| \quad (5)$$

gdzie: Γ – współczynnik odbicia od wejścia anteny, Z – impedancja anteny na zaciskach zasilania, Z_{ref} – impedancja odniesienia źródła zasilania.



Rys. 4. Wykres parametru S_{11} w funkcji częstotliwości dla dipola półfalowego

Dane o impedancji wejściowej zapisane są w pliku wynikowym symulacji, natomiast impedancję odniesienia przyjmuje się zwykle jako 50Ω. Im niższa wartość parametru S_{11} tym lepsze dopasowanie, ponieważ parametr ten określa jak dużo energii zostało odbite od wejścia anteny. Z wykresu przedstawionego na rysunku 4 można łatwo odczytać, że najlepsze dopasowanie antena posiada około 142MHz.

5. Skrypty do wizualizacji

Proces wizualizacji parametrów anteny składa się z czterech etapów. Pierwszym etapem jest wywołanie programu NEC-2 w celu wygenerowania pliku wynikowego. Następnie w drugim etapie plik ten jest parsowany przez skrypt *necparse.py* do postaci łatwiejszej do późniejszej ekstrakcji danych do wizualizacji. Dane są przetwarzane do postaci słownikowej (zorganizowanej w postaci kluczy i wartości) a następnie zapisywane w binarnym pliku, z którego Python jest w stanie od razu wczytać słownik (za pomocą modułu *pickle*).

Następnie skrypty wyodrębniające informacje potrzebne do przedstawienia poszczególnych parametrów wykorzystują dane wyodrębnione w poprzednim kroku i przygotowują ostateczny plik z danymi.

Ostatni krok, to właściwa wizualizacja. Wykresy dwuwymiarowe są generowane przez skrypty programu *gnuplot*. Dodatkowo, do zaawansowanej wizualizacji 3D wykorzystywany jest skrypt eksportujący wygenerowane siatki powierzchniowe charakterystyk promieniowania do formatu Wavefront OBJ.

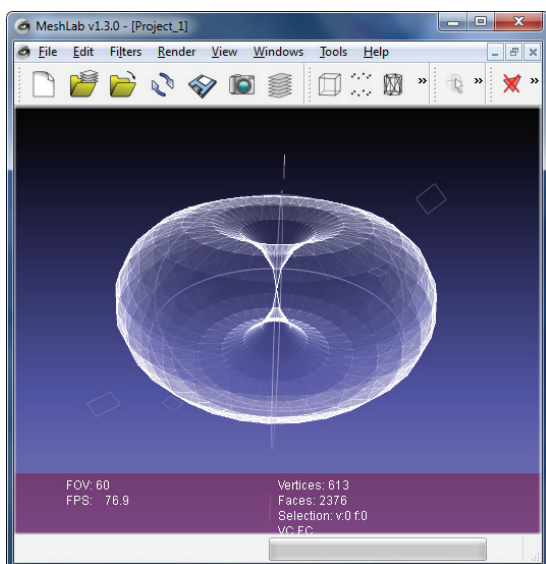
Przygotowanie obrazu z wizualizacji 3D przeprowadzone jest w programie *Blender* oraz *MeshLab* (rys. 5).

Poniżej, na Listingu 1, przedstawiony jest fragment kodu służącego do przetworzenia wyniku programu NEC-2 do wygodniejszej postaci.

Listing 1. Konwersja fragmentu pliku wynikowego do postaci słownikowej

```
# do wyodrębniania danych korzystamy
# z wyrażeń regularnych
import re

def parse(fi):
    ne = []
    print ":: [Parsing Near Field...]"
    assert fi.popleft() is '@'
    line = fi.popleft() # (header)
    line = fi.popleft() # (header)
    line = fi.popleft() # (header)
    indentation = len(line) - len(line.lstrip())
    line = fi.popleft() # (data)
    while line is not '@':
        r = r'(.{1,7})(.{1,11})(.{1,11})\
            (.{1,15})(.{1,9})(.{1,14})\
            (.{1,9})(.{1,14})(.{1,9})'
        p = re.findall(r, line[indentation:])
        (x,y,z, xm,xp, ym,yp, zm,hzp) = p[0]
        ne.append({
            'x': float(x), 'y': float(y), 'z': float(z),
            'ex': (float(xm), float(xp)),
            'ey': (float(ym), float(yp)),
            'ez': (float(zm), float(zp))
        })
    line = fi.popleft() # (data)
    print ":: [Parsed %d NF probes]" % len(ne)
    return ne
```



Rys. 5. Podgląd charakterystyki promieniowania dipola półfalowego zapisanej w formacie OBJ wyświetlonej w programie MeshLab

Powyższy listing odpowiada za parsowanie danych próbek bliskiego pola elektrycznego. W wyniku swojego działania tworzy on słownik `ne`, który zawiera listę wartości próbek pola E:

Listing 2. Reprezentacja fragmentu pliku wynikowego w postaci słownika

```
ne = [ {
    x: 0, y: 0.01, z: 0.5,
    ex: (1.60e-2, 90.11),
    ey: (6.64e-2, 90.0),
    ez: (3.77e-2, -89.88)
}, {
    x: 0, y: 0.21, z: 0.5,
    ex: (3.85e-2, 90.06),
    ey: (6.06e-2, 90.0),
    ez: (3.75e-2, -89.96)
}, {
    x: 0, y: 0.41, z: 0.5,
    ex: (4.82e-2, 90.04),
    ey: (4.69e-2, 90.0),
    ez: (2.87e-2, -89.96)
} ... ]
```

Warto zwrócić uwagę, że wektory zespolone E_x, E_y, E_z zostały zapisane w postaci pary (argument, moduł). W tej formie są wygodniejsze do późniejszej konwersji. Ponieważ NEC-2 generuje wartości pola E dla każdej z badanych przez użytkownika częstotliwości tworzona jest następująca hierarchiczna struktura:

```
out = {
    frequencies: [{
        freq: 1e6, # 1MHz
        wavelength: 299.8, # w metrach
        nearelectric: ne
    } ... ],
    ...
}
```

Drugi (przedstawiony na listingu 3) fragment skryptu odpowiada za wyodrębnienie danych z postaci słownikowej do ostatecznej formy gotowej do wizualizacji.

Listing 3. Wyluskanie danych potrzebnych do wizualizacji charakterystyki pola bliskiego

```
...
pattern = [] # Charakterystyka pola bliskiego

freqs = [] # Częstotliwości z symulacji

for i in range(len(out['frequencies'])):
    freqs.append(str(out['frequencies'][i]['freq']))

# Nagłówek pliku z wynikami
print '%\tTheta\t%s' % '\t'.join(freqs)

# Przechodzimy po wszystkich kątach phi/theta
for i in range(nangle):
    freq = [] # List częstotliwości dla jednego kierunku
    for j in range(nfreqs):
        p = out['frequencies'][j]['nearelectric'][i]
        x, y, z = p['x'], p['y'], p['z']

        # Przeliczenie współrzędnych kartezjańskich
        # na biegunowe
        r = math.sqrt(x**2 + y**2 + z**2)
        phi = degrees(math.atan2(y, x))
        theta = degrees(math.acos(z/r))
        (m, pe) = p['ex']
        ex = cmath.rect(m, pe)
        (m, pe) = p['ey']
        ey = cmath.rect(m, pe)
        (m, pe) = p['ez']
        ez = cmath.rect(m, pe)
        # Wyliczenie modułu wektora natężenia pola
        # elektrycznego E w punkcie p
        value = cmath.sqrt(ex*ex+ey*ey+ez*ez)
        freq.append({
            'theta':theta, 'phi':phi,
            'x':x, 'y':y, 'z':z,
            'value':value.real
        })

    t, p = freq[0]['theta'], freq[0]['phi']
    x, y, z = freq[0]['x'], freq[0]['y'], freq[0]['z']
    # t, p, x, y, z powinny być takie same
    # dla każdego freq
    pattern.append({
        'theta':t, 'phi':p,
        'x':x, 'y':y, 'z':z,
        'frequencies': freq
    })

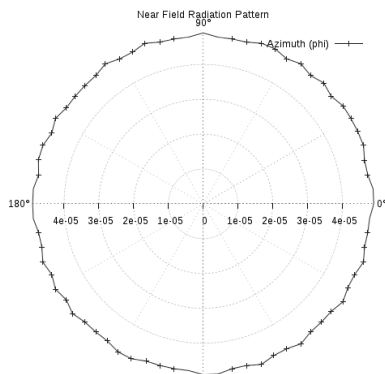
# wypisanie danych do wizualizacji
# theta, phi, |E| dla każdej z częstotliwości
for p in pattern:
    theta, phi = p['theta'], p['phi']
    x, y, z = p['x'], p['y'], p['z']
    print '%f\t%f' % (phi, theta),
    for f in p['frequencies']:
        print '\t%f' % f['value']
    print
```

Po uruchomieniu skryptu przedstawionego na listingu 3 na standardowe wyjście wypisywane są dane w formacie, który może zostać wykorzystany przez program *gnuplot* do wyświetlenia wykresu w układzie współrzędnych biegunowych ilustrującego wartości bliskiego pola elektrycznego w formie podobnej do charakterystyki promieniowania. Skrócona wersja stosowanego skryptu została przedstawiona na listingu 4.

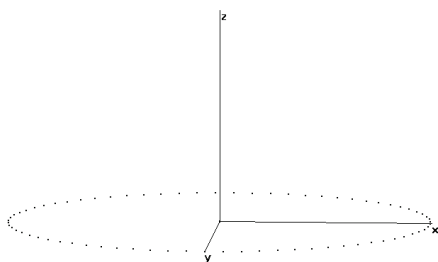
Listing 4. Skrypt *gnuplot* rysujący wykres charakterystyki pola bliskiego anteny w współrzędnych biegunowych

```
set polar
set clip
set angles degrees
set grid polar 30 lc rgbcolor "#999999"
set size square
unset border
unset param

set zeroaxis
set xtics axis in scale 1,0.5 nomirror norotate off-
set character 0, 0, 0
set noyticks
set label 1 "0°" at graph 1.01,0.5 front
set label 2 "180°" at graph -0.01,0.5 right front
set label 3 "-90°" at graph 0.5,-0.03 center front
set label 4 "90°" at graph 0.5,1.03 center front
plot 'dip/ne.dat' using 1:3 w lp title "Azimuth (phi)"
```



Rys. 6. Wykres wartości pola elektrycznego $|E|$ w zależności od kierunku dla dipola krótkiego w odległości 1.5m od niego (odległość punktu od środka wykresu jest równa wartości $|E|$ w V/m)



Rys. 7. Punkty 1.5m w odległości od anteny z wyliczonymi wektorami pola E (słabo widoczne, z powodu małych wartości pola w tych punktach)

Wynikiem wyżej przedstawionego skryptu jest wykres przedstawiony na rysunku 6. Plik wejściowy dla NEC-2 zawierał żądanie obliczenia wartości natężenia pola elektrycznego w punktach 1.5m od środka dipola. Rysunek 7 przedstawia zaznaczone punkty wyświetlone w programie Xnec2.

6. Wnioski

Oryginalny format pliku wyjściowego NEC-2 nie pozwala na bezpośrednią wizualizację parametrów anteny ponieważ wiele z wartości w nim zawartych wymaga przetworzenia. Przy użyciu skryptu w Pythonie, możliwa jest konwersja pliku wynikowego NEC do postaci dużo wygodniejszej do późniejszej wizualizacji. Przetworzone wyniki zapisane w hierarchicznej, opartej o słownik strukturze bardzo dobrze nadają się do analizy służącej wizualizacji parametrów anten. Autorzy zaprezentowali możliwości wizualizacji charakterystyki promieniowania anteny, impedancji falowej oraz parametru S_{11} wykorzystując do tego zestaw skryptów w języku Python, program *gnuplot* oraz aplikację do renderowania obiektów zapisanych w formacie OBJ (MeshLab, Blender). Wygenerowane wykresy oraz obiekty są przejrzyste i łatwe do interpretacji, przez co stanowią pomoc w procesie modelowania anten.

Literatura

- [1] Cebik L. B.: *A Beginner's Guide to Modeling with NEC*, 2005.
- [2] Kishk, A. A., *Fundamentals of Antennas*, Chapter 1 on Antennas for Base Stations in Wireless Communications, Edited by Zhi Ning Chen and Kwai-Man Luk, by The Mc Graw Hill Companies, 2009.
- [3] Richeson P.: *NEC-2 Manual, Part III: User's Guide*, 1996.
- [4] Starzyński J., Sroka J.: *Optymalne projektowanie anteny generującej pole elektryczne do testów narażeniowych*. Przegląd Elektrotechniczny 4a/2012, s. 114.
- [5] <http://www.qsl.net/5b4az/pkg/nec2/xnec2c/doc/xnec2c.html#RadiationWindow.MeshLab> <http://meshlab.sourceforge.net/>
- [6] Blender <http://www.blender.org/>

Mgr inż. Bartosz Chaber

e-mail: chaberb@iem.pw.edu.pl

Asystent w Instytucie Elektrotechniki i Systemów Informatycznych Wydziału Elektrycznego Politechniki Warszawskiej. W swojej pracy naukowej zajmuje się budowaniem narzędzi do wizualizacji i obliczeń naukowych oraz modelowaniem anten. Ukończył studia magisterskie na Politechnice Warszawskiej w 2011 roku.



Dr hab. inż. Jacek Starzyński

e-mail: jstar@ee.pw.edu.pl

Jacek Starzyński urodził się w Warszawie w 1962. Otrzymał tytuł magistra, oraz stopnie doktora i doktora habilitowanego na Politechnice Warszawskiej odpowiednio w 1986, 1995 i 2009 roku. Aktualnie jest profesorem w dziedzinie elektrotechniki w Instytucie Elektrotechniki Teoretycznej i Systemów Informatycznych PW. W swojej pracy naukowej zajmuje się polem elektromagnetycznym, EMC, metodami optymalizacji oraz metodami numerycznymi.



Artykuł recenzowany