

М. В. Бугайов, О. А. Нагорнюк, В. В. Клязника, В. Л. Рикун, О. С. Гаврилюк

РЕАЛІЗАЦІЯ ПІДХОДІВ ДО ІМІТУВАННЯ ДИНАМІЧНОЇ РАДІОЕЛЕКТРОННОЇ ОБСТАНОВКИ В ШИРОКІЙ СМУЗІ ЧАСТОТ

Однією з основних особливостей збройного конфлікту на сході України залишається ведення бойових дій в умовах динамічної зміни складної радіоелектронної обстановки. Постійне зростання насиченості радіоефіру, використання різномісних протоколів передавання даних зумовлюють необхідність підвищення рівня підготовленості військових фахівців у сфері технічного аналізу сигналів та розпізнавання випромінювань. Це, у свою чергу, актуалізує необхідність реалізації підходів до імітування динамічної радіоелектронної обстановки у широкій смузі частот для їх використання в навчальному процесі. Формування радіоелектронної обстановки із використанням технологій програмно-визначеного радіо дозволяє створити сценарій практично будь-якої складності. Радіоелектронна обстановка у широкій смузі частот може розроблятися в реальному масштабі часу або бути заздалегідь сформованою і записаною в необхідному форматі. Для створення програмних модулів формування радіосигналів в реальному масштабі часу доцільно використати платформу цифрової обробки сигналів GNU Radio. Максимальна ширина смуги частот в такому разі відповідає максимальній частоті дискретизації використаного приймально-передавального модуля. За такого підходу можна в реальному масштабі часу змінювати потужність і несучу частоту всіх складових. Обмеження формування сигналів у реальному масштабі часу із використанням GNU Radio пов'язані з тим, що ширина смуги та складність сигнальної обстановки, яка може бути сформована, залежать від продуктивності процесора комп'ютера. Записи радіоелектронної обстановки можна сформувати з використанням мов програмування високого рівня: Python та Matlab, – які містять широкий вибір спеціалізованих бібліотек з обробки сигналів. У подальшому такі файли можна використовувати для їх аналізу в режимі офлайн за допомогою відповідних програмних засобів або як джерела сигналів для випромінювання програмно-визначеними радіосистемами.

Ключові слова: динамічна радіоелектронна обстановка; комплексний сигнал; програмно-визначене радіо; GNU Radio; Python; Matlab.

Постановка проблеми в загальному вигляді. Бойові дії на сході України характеризуються досить специфічними способами та методами їх ведення. Однією з основних особливостей збройного конфлікту на сході України залишається ведення бойових дій в умовах динамічної зміни складної радіоелектронної обстановки (РЕО). Тому в даний час досить актуальними є заходи, пов'язані з формуванням динамічної РЕО в ході вирішення широкого кола завдань, зокрема пов'язаних із підготовкою військових фахівців.

Для забезпечення стійкого, безперервного та надійного зв'язку, виконання вимог прихованого управління військами на сучасному полі бою розгортають багаторівневу

систему управління та зв'язку, важливою частиною якої є мережі радіозв'язку ультракороткохвильового (УКХ) діапазону частот. Організація систем тактичного УКХ радіозв'язку здійснюється з використанням аналогових та цифрових радіостанцій як військового, так і цивільного призначення. Для передачі важливих повідомлень застосовують цифрові радіостанції з псевдовипадковим перестроюванням робочої частоти (ППРЧ). Радіосигнали із цифровими видами модуляції використовують також для організації командної радіолінії управління безпілотними літальними апаратами (БПЛА) та в каналах телеметрії [1]. Можливе функціонування режиму ППРЧ у межах дозволеної смуги частот. Також для управління комерційними БПЛА використовують радіосигнали стандартів Wi-Fi IEEE 802.11 b/g/n.

Постійне зростання насиченості радіоефіру, використання різнотипних протоколів передавання даних зумовлюють необхідність підвищення рівня підготовки фахівців у сфері технічного аналізу сигналів та розпізнавання випромінювань. Це, у свою чергу, актуалізує розроблення підходів до імітування динамічної РЕО в широкій смузі частот для їх використання в навчальному процесі.

Аналіз останніх досліджень і публікацій. Серед імітаторів радіосигналів закордонного виробництва можна виділити вироби типу AN/PLM-4 (RSS), “Шиповник-2”, “Астра-2” та “Аврора”. Дані пристрої є вузькоспеціалізованими, вони призначені для вирішення часткових завдань. У [2] описано розроблений програмно-апаратний комплекс імітації сигналів радіоелектронних систем і комплект спеціального програмного забезпечення імітації РЕО. Кількість пристроїв імітації може становити кілька десятків екземплярів, вони об'єднуються в єдину просторово-розподілену систему та функціонують за заданим сценарієм. У [3–5] описано алгоритми формування багатьох типів сигналів із використанням мови програмування Python. Відомі також численні приклади реалізації подібних алгоритмів у середовищі Matlab [6–8]. Проте описані рішення забезпечують створення досить простої РЕО.

На даний час переважна більшість засобів радіомоніторингу та радіоелектронної розвідки мають відносно широку миттєву смугу огляду (десятки мегагерц), тому в деяких діапазонах частот (особливо в ISM-діапазонах) завантаженість смуги частот аналізу різнотипними сигналами може бути досить високою. Усе це створює передумови до вдосконалення відомих та розроблення нових підходів до формування складної РЕО.

Формулювання завдання дослідження. Метою статті є практична реалізація підходів до імітування динамічної РЕО в широкій смузі частот.

Виклад основного матеріалу

Технологія програмновизначеного радіо

На даний час більшість функцій сучасної радіосистеми реалізується з використанням не апаратного, а програмного забезпечення. Це спричинило розроблення архітектури програмновизначеної радіосистеми (ПВР): радіопередавач і/або радіоприймач, який використовує технологію, що дозволяє за допомогою програмного забезпечення встановлювати або змінювати робочі радіочастотні параметри, включаючи діапазон частот, тип модуляції, кодування, вихідну потужність [9–10]. Найбільш розповсюджені приймально-передавальні пристрої побудовані з використанням технології ПВР

дозволяють випромінювати сформовану сигнальну суміш у діапазоні від 0 до 18 ГГц із максимальною шириною смуги частот до 245 МГц. Розрядність цифро-аналогового перетворювача таких пристроїв досягає 16 біт, що дозволяє формувати сигнали із практично всіма видами сигнально-кодових конструкцій. У разі необхідності сформувати широкосмугову динамічну РЕО в діапазоні частот, значення якого перевищує максимальну ширину смуги передавача, можна використати декілька пристроїв ПВР. Отже, сучасна радіоелектронна база дозволяє імітувати сигнально-завадову обстановку практично в необмеженому частотному діапазоні.

Пропонуємо два основні підходи до імітування динамічної РЕО в широкій смузі частот:

формування РЕО в реальному масштабі часу та випромінювання сигналу з використанням ПВР;

формування РЕО заздалегідь, запис сформованих комплексних відліків сигналу до спеціалізованого файлу та наступне його випромінювання з використанням ПВР.

На вибір підходу до імітування динамічної РЕО впливають розрахункова складність сценарію, пов'язана із шириною частотного діапазону та кількістю сигналів, що моделюються.

Структура програмно-апаратного комплексу імітування РЕО є однаковою для двох запропонованих підходів, вона повинна включати персональну електронно-обчислювальну машину (ПЕОМ), сукупність ПВР, підсилювачів потужності та широкосмугових антен.

Нижче наведено опис процесу формування широкосмугового сценарію РЕО в середовищі GNU Radio. Розроблення з використанням інших програмних засобів буде схожим. Однак GNU Radio має низку переваг: програмний продукт розповсюджується із відкритим кодом, є кросплатформним, містить широкий набір бібліотек, які дозволяють створювати довільні радіосистеми, схеми модуляції та кодування, працює із найбільш розповсюдженими ПВР.

Формування РЕО в реальному масштабі часу

Код більшості компонентів GNU Radio написаний мовою програмування Python, а частини, критичні до швидкодії та часу затримки, – мовою C++, що дозволяє використовувати пакет для вирішення завдань у режимі реального часу [11]. До складу GNU Radio входить набір фільтрів, каналних кодеків, модулів синхронізації, демодуляторів, еквалайзерів, голосових кодеків, декодерів й інших елементів, необхідних для створення радіосистем. Зазначені елементи можуть бути використані як складові для компонування готової системи. Якщо необхідно створити компонент зі специфічним функціоналом, якого немає в бібліотеці (out-of-tree module), то GNU Radio має у своєму складі утиліту для створення шаблону нового блоку – `gr_modtool` [12].

Проект GNU Radio подано у вигляді потокового графа, вузли якого відображають блоки оброблення сигналів, а ребра – потоки даних між блоками. Блоки, які реалізують складні алгоритми обробки сигналів, написані мовою C++. Кожен блок характеризується параметрами, які позначають кількість вхідних і вихідних портів, тип даних, який вони обробляють. Пакет включає набір бібліотек, кожна з яких складається з декількох блоків для обробки сигналів. Для підключення бібліотек, реалізованих на мові C++ з керуючим

скриптом, написаним на Python, використовується генератор інтерфейсу SWIG. Програми на C++ реалізують алгоритми оброблення сигналів, Python використовують для формування потокового графа, об'єднання блоків і запуску програми на виконання. У разі використання SWIG усі бібліотеки на C++ стають доступними з вихідного програмного коду на Python.

За ПВР можна вибрати будь-який трансивер, сумісний з GNU Radio [13]. Якщо використати Hack RF One [14], який може працювати з частотами дискретизації від 8 МГц до 20 МГц, то смугу для формування сигнальної обстановки можна обрати із заданого діапазону. Більша частота дискретизації вимагає використання більш продуктивного процесора ПЕОМ та більшого об'єму оперативної пам'яті для формування сигналів у реальному масштабі часу. Для побудови графів формування радіосигналів у GNU Radio оберемо смугу частот шириною 8 МГц.

У заданій смузі будемо формувати сигнальну суміш, що є сумою широкопasmового шуму, трьох вузькосмугових (гармонічних) завад, а також OFDM, QAM і ППРЧ сигналів. Також забезпечимо можливість змінювати потужність та несучу частоту усіх складових у реальному масштабі часу. Під час проведення практичних занять у спеціалізованій аудиторії через близькість розташування приймальної та передавальної антени ефект впливу каналу поширення радіохвиль (завмирання) можна моделювати шляхом внесення спотворень у сформовану сигнальну суміш у самому графі GNU Radio.

У стандартній бібліотеці GNU Radio є набір блоків для реалізації схем формування сигналів із цифровими видами модуляції: OFDM, GFSK, GMSK, QAM, PSK, CPFSK тощо. Також за допомогою блоків Constellation Rect. Object та Constellation Modulator можна створювати довільні фазові сузір'я. Бажаний функціонал даних блоків може бути досягнутий правильним вибором значень їх параметрів. З метою формування сигналу із міжсимвольною та посимвольною ППРЧ для стандартної бібліотеки було розроблено та додано власний блок FHSS.

Після запуску файлу із зібраним графом формування сигналу з розширенням *.gnc у цьому ж каталозі буде створено файл із розширенням *.grc. Для подальшого користування в ході роботи з операційною системою Linux можна запускати будь-який із цих двох файлів. Їх відмінність полягає в тому, що в разі запуску файлу із розширенням *.grc відразу ж запускатиметься робоче вікно і вмикатиметься передавач, а за розширення *.gnc спочатку відкриється середовище GNU Radio Companion, у якому можна здійснювати зміну параметрів блоків і запускати граф на виконання. У разі формування сигналів у GNU Radio основним параметром, який визначає усі часові інтервали, що використовуються у графі, є частота дискретизації.

У ході запуску графа на виконання відкривається графічний інтерфейс користувача (рис. 1) і відразу ж починається формування сигналів. У лівій верхній частині інтерфейсу розташовано 3 вкладки: "Параметри сигналів", "Характеристики каналу" та "Параметри передавача". На вкладці "Параметри сигналів" містяться інструменти для регулювання центральної частоти та амплітуди сигналів із ППРЧ, QAM та OFDM. У нижній частині інтерфейсу відображаються спектр та спектрограма сформованої сигнально-завадової обстановки.

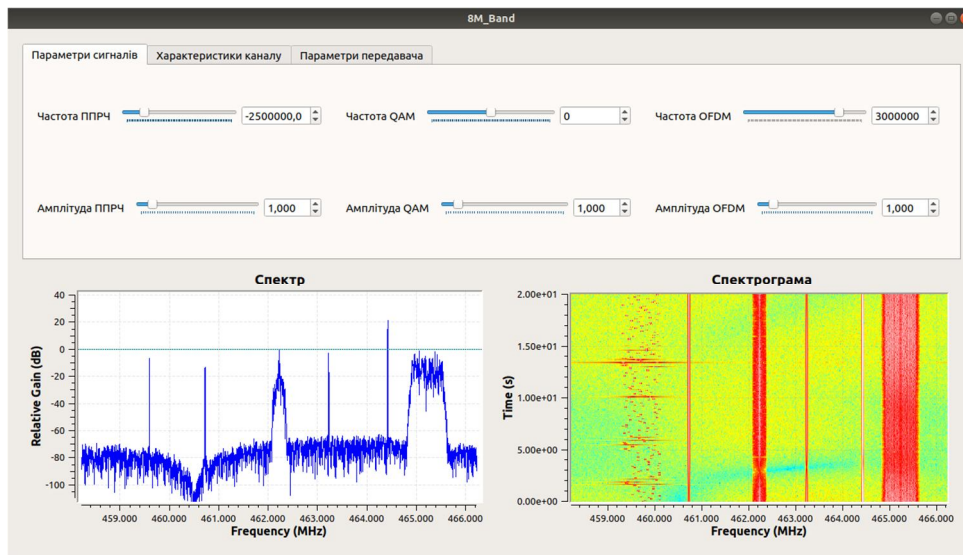


Рис. 1. Графічний інтерфейс користувача з відкритою вкладкою “Параметри сигналів”

На вкладці “Характеристики каналу” (рис. 2) розташовано інструменти для регулювання середньоквадратичного відхилення шуму, кількості синусоїд (шляхів поширення сигналу) та швидкості завмирань для моделювання стану каналу, а також амплітуд та частот вузькосмугових завад.

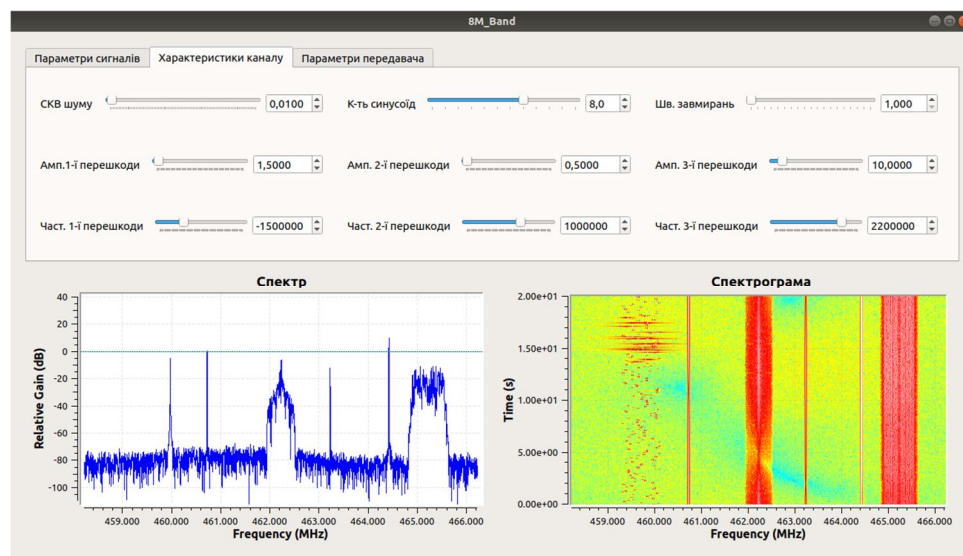


Рис. 2. Графічний інтерфейс користувача з відкритою вкладкою “Характеристики каналу”

Частотно-селективні завмирання в каналі поширення моделюються за допомогою блоку Frequency Selective Fading Model. Налаштуванню підлягають кількість синусоїд для моделювання підсилення в кожному промені (шляху поширення сигналу), нормалізований максимальний доплерівський зсув частоти, модель каналу (Релея або Райса), а також параметри фільтра для моделювання затримок у кожному промені.

На вкладці “Параметри передавача” (рис. 3) можна встановити несучу частоту передавача HackRF One (повинна бути в межах смуги пропускання антени), включати або

відключати підсилювач потужності на несучій частоті, а також змінювати в межах від 0 дБ до 47 дБ з кроком 1 дБ коефіцієнт підсилення на проміжній частоті.

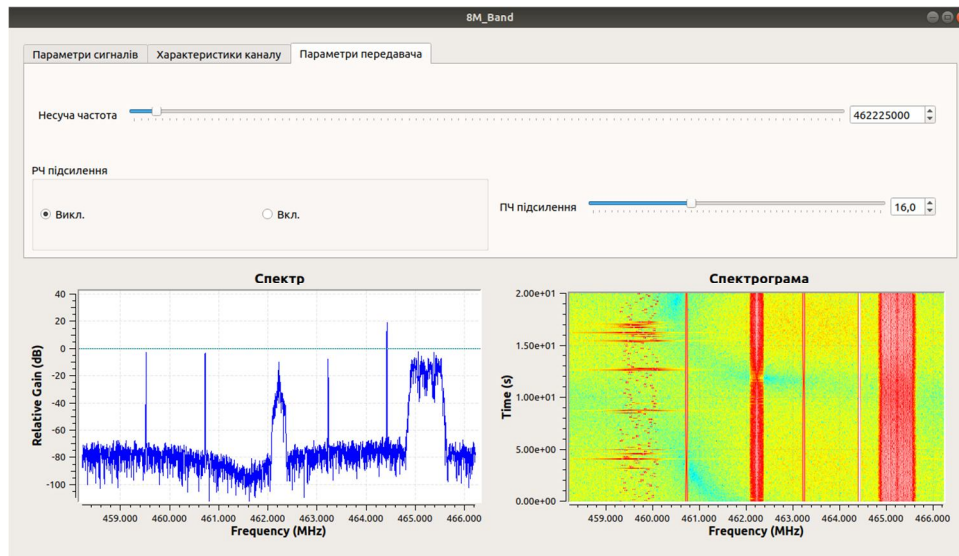


Рис. 3. Графічний інтерфейс користувача з відкритою вкладкою “Параметри передавача”

Обмеження формування РЕО в реальному масштабі часу із використанням GNU Radio пов’язані з тим, що ширина смуги і складність РЕО залежать від продуктивності ПЕОМ. Проте можна використовувати записи широкосмугових сценаріїв із використанням відповідного блоку зчитування.

Створення записів динамічної РЕО в широкій смузі частот

Записи динамічної РЕО в широкій смузі частот можна використовувати для їх аналізу в режимі офлайн за допомогою відповідних програмних засобів (наприклад, Signals Analyzer) або як файлів для випромінювання.

Кожний стандарт зв’язку встановлює перелік видів модуляцій, які можуть використовуватися для формування радіосигналу, та діапазон зміни основних параметрів (символьна швидкість, несуча частота, зсув частот). Крім того, для підвищення розвідзахисності та завадостійкості в телекомунікаційних системах використовують розширення спектра сигналу.

Загальний математичний опис радіосигналу для будь-якого виду модуляції, який розповсюджується в каналі, що описується гаусівською моделлю, можна подати як:

$$r(t, U_i) = s(t, U_i) + n(t), \tag{1}$$

де $s(t, U_i)$ – корисна складова;

$n(t)$ – адитивний гаусівський шум;

$U_i = \left[a_i, f_c, \theta, T, \varepsilon, g(t), \left\{ s_k^{(i)} \right\}_{k=1}^K \right]$ – вектор модуляційних параметрів;

a_i – амплітуда сигналу;

f_c – частота несучого коливання;

θ – початкова фаза несучого коливання;

T – символний період;

ε – параметр, що визначає фазу модулюючого коливання;

$g(t)$ – сумісна імпульсна характеристика формуючого фільтра та каналу;

$\{s_k^{(i)}\}_{k=1}^K$ – K комплексних символів, що передаються та беруться з i -го кінцевого алфавіту для відповідного виду модуляції.

Корисний сигнал з лінійною цифровою модуляцією описується таким виразом:

$$s(t, U_i) = a_i e^{j(2\pi f_c t + \theta)} \sum_{k=1}^K s_k^{(i)}(t) g(t - (k-1)T - \varepsilon T). \quad (2)$$

Амплітуда сигналу визначається за формулою

$$a_i = \sqrt{\frac{E_s}{\sigma_{s^{(i)}}^2} E_p}, \quad (3)$$

де E_s – енергія сигналу на нульовій частоті;

E_p – енергія імпульсу;

$\sigma_{s^{(i)}}^2$ – дисперсія i -го фазового сузір'я сигналу з нульовим математичним сподіванням.

Розглянемо формування сигнальної суміші з використанням мови програмування Python та її запис у форматі *.wav, хоча вибір формату файлу не є принциповим.

Файл у форматі *.wav складається із області заголовка та області даних. У заголовку зберігається інформація про розмір файлу, кількість каналів, частоту дискретизації, кількість біт у відліку сигналу. Амплітуда відліку виражається числом, що займає у файлі, як правило, 8, 16, 24 або 32 біти. Чим більше місця займає число у файлі, тим ширший можливий діапазон його значень і вища точність відтворення амплітуди сигналу. Для РСМ-файлів із розрядністю 16 біт значення амплітуди знаходяться в межах 32767 ... 32768.

Для дійсного сигналу значення амплітуд в області даних розміщуються послідовно. Для комплексного сигналу значення дійсної та уявної частини відліків сигналу записуються у два канали. Фрагмент коду для формування і запису комплексного сигналу у *.wav файл наведено нижче.

```

1. import wave
2. import struct
3. import numpy as np
4. from scipy import signal
5. samplerate = int(8e6)
6. t_reo = 120
7. section_length = int(1e5)
8. L = int(samplerate*t_reo)//section_length
9. nchannels = 2

```

```

10. sampwidth = 2
11. nframes = L*section_length
12. comptype = 'NONE'
13. compname = 'not compressed'
14.
15. wav_file = wave.open(file_name, 'w')
16. wav_file.setparams((nchannels, sampwidth, samplerate, nframes, comptype,
compname))
17.
18. A = np.iinfo(np.int16).max
19.
20. #Блок встановлення параметрів складових
21.
22. for i in range(L):
23.     t = np.arange(i*section_length, (i+1)*section_length)/samplerate
24.
25. #Блок формування складових широкопasmугового сценарію
26.
27.     s = signal.hilbert(sig)
28.     I = np.real(s)
29.     I = I.astype(np.int16)
30.     Q = np.imag(s)
31.     Q = Q.astype(np.int16)
32.     s = np.vstack((I, Q)).T
33.     data = s.flatten()
34.     buf = struct.pack(str(len(data)) + 'h', *data)
35.     wav_file.write(buf)
36. wav_file.close()

```

У рядках 1–4 виконується імпорт необхідних пакетів. У 5 встановлюється необхідна частота дискретизації *samplerate*, у рядку 6 – тривалість запису в секундах, у 7 – довжина фрагмента сигналу (у відліках), який формується і записується за один раз. Для обраної частоти дискретизації, тривалості запису і розрядності амплітуди об’єм сформованого сигналу на диску займатиме $8 \text{ (МГц)} \cdot 120 \text{ (с)} \cdot 16 \text{ (біт)} = 1,92 \text{ (ТГб)}$. Тому одночасне формування і запис такого величезного об’єму даних доцільно проводити фрагментами. Для довжини фрагмента 100000 відліків час, необхідний для формування і запису сигналу тривалістю 120 с на ПЕОМ із процесором Intel Core i7-10510U CPU 1,80 GHz \times 8 з об’ємом оперативної пам’яті 16 ГГб під управлінням операційної системи Ubuntu 20.04.2, становить близько 500 с. Кількість фрагментів запису *L* розраховується в рядку 8. Для кожного фрагмента в циклі (рядки 22–35) створюється масив моментів часу (рядок 23), для яких розраховуються відліки складових широкопasmугового сценарію *sig*. Після формування усіх складових вони додаються. У рядку 15 створюється об’єкт файлу запису, а в 16 встановлюються його параметри. Перед циклом формування сценарію необхідно задати часові, частотні та енергетичні параметри усіх складових. Значення центральних частот складових задаються в межах від 0 до *samplerate*. Особливістю

встановлення часових параметрів для сигналів із цифровими видами модуляції є те, що на тривалості фрагмента запису повинна вміщатися ціла кількість символів (частотних елементів для ППРЧ). Щодо енергетичних параметрів сигналів, то сума амплітуд складових у жоден момент часу не повинна перевищувати 2^{16} або бути меншою -2^{16} . У протилежному разі виникатимуть амплітудні спотворення.

Для формування дійсного сигналу необхідно встановити один канал запису ($nchannels = 1$), значення центральних частот складових обирати в межах від 0 до $sampleRate/2$ та в рядку 34 для упаковки буфера замість даних data використовувати відліки sig, які попередньо необхідно привести до типу int16.

Створені записи широкосмугових сценаріїв можна використовувати для проведення практичних занять як завдання для аналізу в режимі офлайн, як джерело сигналу в графах GNU Radio або інших програмних засобах. На рис. 4 наведено графічний інтерфейс користувача програми “HackRFBoard”, яка дозволяє завантажувати запис комплексного сигналу у форматі *.wav та розпочинати випромінювання за допомогою ПВР HackRF One. На вкладках Oscilogramm та Spectrum відображається часова реалізація та спектр завантаженого широкосмугового сценарію.

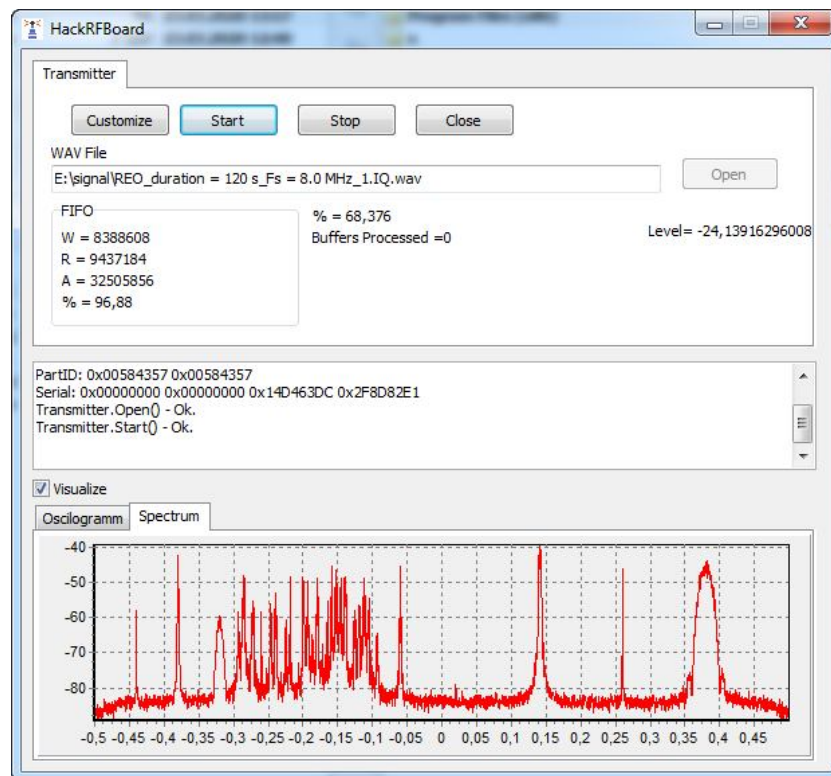


Рис. 4. Графічний інтерфейс користувача з відкритою вкладкою “HackRFBoard”

Спектрограму сформованого широкосмугового сценарію наведено на рис. 5. Поряд із сигналами із цифровими видами модуляції (QAM-16, FSK-2, PSK-4, ППРЧ) розміщено сигнали з амплітудною модуляцією АЗЕ у смузі 6 кГц та частотною модуляцією F3Е у смузі 16 кГц. Останні два види сигналів імітують роботу аналогових радіостанцій під час передачі мовленнєвих повідомлень. Гармоніки імітують вузькосмугові завади.

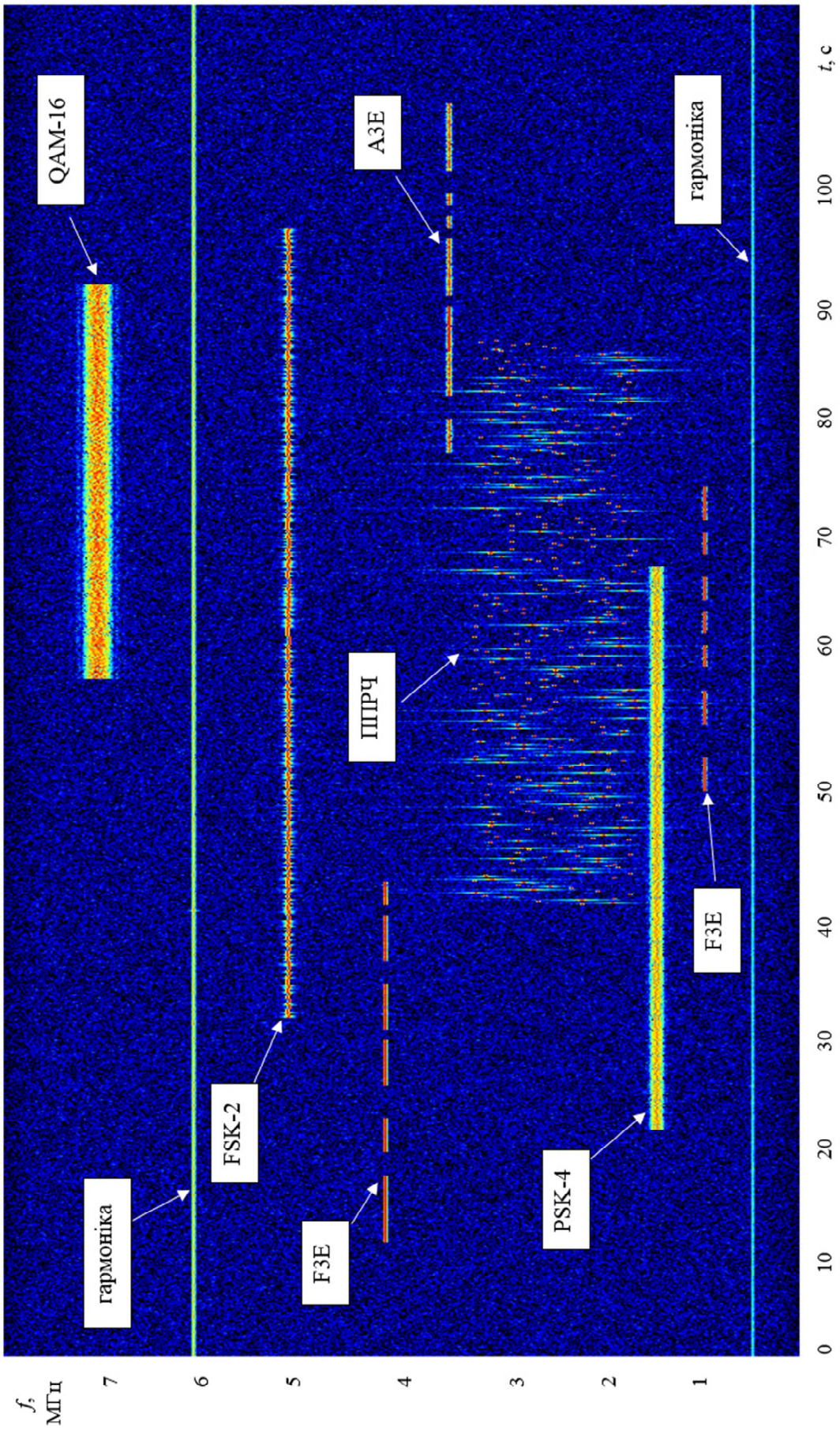


Рис. 5. Частотно-часове зображення динамічної РЕО в широкій смузі частот

Висновки. Запропоновано реалізацію підходів до формування динамічної РЕО в реальному масштабі часу та до попереднього створення їх записів із використанням мови програмування високого рівня. Окреслено ключові моменти роботи з платформою GNU Radio в ході формування динамічної РЕО в широкій смузі частот та запропоновано варіант графа генерування сигналів. Удосконалено математичний апарат описання радіосигналу зі складними видами модуляції. Наведено варіант створення запису сигнальної суміші у форматі *.wav із використанням мови програмування Python. Отримані результати в подальшому можуть бути використані під час проведення практичних занять для підвищення рівня підготовки військових фахівців.

Перспективи подальших досліджень у даному напрямку полягають у розробленні гнучких алгоритмів формування сигналів зі складними видами модуляції.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. Chamola V. et al. A Comprehensive Review of Unmanned Aerial Vehicle Attacks and Neutralization Techniques. *Ad Hoc Networks* 111. 2021. 20 p. doi: [10.1016/j.adhoc.2020.102324](https://doi.org/10.1016/j.adhoc.2020.102324)
2. Программно-аппаратные средства формирования управляемой радиоэлектронной обстановки. URL: <http://www.ntc-reb.ru/article15.html> (дата обращения: 07.10.2021).
3. Downey A. B. Think DSP: Digital Signal Processing in Python. Green Tea Press, 2014. 153 p.
4. Charbit M. Digital Signal Processing with Python Programming. Wiley, 2017. 284 p.
5. Unpingco J. Python for Signal Processing. Featuring IPython Notebooks. Springer, 2014. 133 p.
6. Gopi E. S. Algorithm Collections for Digital Signal Processing Applications Using Matlab. Springer, 2007. 199 p.
7. Quinquis A. Digital Signal Processing using MATLAB. John Wiley & Sons, Inc., 2007. 425 p.
8. Bassem R. Mahafza. MATLAB simulations for radar systems design. Springer, 2004. 686 p.
9. Grayver E. Implementing Software Defined Radio. Springer, 2013. 271 p.
10. Collins T. F. et al. Software-Defined Radio for Engineers. Artech House, 2018. 375 p.
11. GNU Radio. URL: <https://www.gnuradio.org/> (last accessed: 12.10.2021).
12. Out Of Tree Modules. URL: https://wiki.gnuradio.org/index.php/OutOfTreeModules#Other_types_of_blocks (last accessed: 10.10.2021).
13. List of software-defined radios. URL: https://en.wikipedia.org/wiki/List_of_software-defined_radios (last accessed: 12.10.2021).
14. HackRF. URL: <https://greatscottgadgets.com/hackrf/> (last accessed: 10.10.2021).

Стаття надійшла до редакції 08.11.2021.

REFERENCES

1. Chamola, V. et al. (2021). A Comprehensive Review of Unmanned Aerial Vehicle Attacks and Neutralization Techniques. *Ad Hoc Networks*, 111. <https://doi.org/10.1016/j.adhoc.2020.102324>
2. *Программно-аппаратные средства формирования управляемой радиоэлектронной обстановки [Hardware and software for the formation of a controlled electronic environment]*. (n.d.). Retrieved from <http://www.ntc-reb.ru/article15.html> [in Russian].

3. Downey, A. B. (2014). *Think DSP: Digital Signal Processing in Python*. Green Tea Press.
4. Charbit, M. (2017). *Digital Signal Processing with Python Programming*. Wiley.
5. Unpingco, J. (2014). *Python for Signal Processing. Featuring IPython Notebooks*. Springer. <https://doi.org/10.1007/978-3-319-01342-8>
6. Gopi, E. S. (2007). *Algorithm Collections for Digital Signal Processing Applications Using Matlab*. Springer.
7. Quinquis, A. (2007). *Digital Signal Processing using MATLAB*. John Wiley & Sons, Inc.
8. Bassem, R. Mahafza. (2004). *MATLAB simulations for radar systems design*. Springer.
9. Grayver, E. (2013). *Implementing Software Defined Radio*. Springer.
10. Collins, T. F. et al. (2018). *Software-Defined Radio for Engineers*. Artech House.
11. *GNU Radio*. (n.d.). Retrieved from <https://www.gnuradio.org/>
12. *Out Of Tree Modules*. (n.d.). Retrieved from https://wiki.gnuradio.org/index.php/OutOfTreeModules#Other_types_of_blocks
13. *List of software-defined radios*. (n.d.). Retrieved from https://en.wikipedia.org/wiki/List_of_software-defined_radios
14. *HackRF*. (n.d.). Retrieved from <https://greatscottgadgets.com/hackrf/>

M. V. Buhaiov, O. A. Nahorniuk, V. V. Kliazyuka, V. L. Rykun, O. S. Havryliuk
REALIZATION OF APPROACHES TO SIMULATION DYNAMIC BROADBAND SCENARIOS OF ELECTRONIC ENVIRONMENT

One of the main principles of the armed conflict in the east of Ukraine remains warfare in a dynamic change of complex electronic environment. The constant increase frequency bands occupancy, the use of various types of data transmission protocols necessitates an increase in the level of training of military specialists in the field of technical analysis of signals and radio emissions recognition. This, in turn, necessitates the development of approaches to simulating wideband scenarios for use in training process. The formation of the electronic environment using the technology of software-defined radio allows us to create a scenario of almost any complexity. Wideband scenarios can be generated in real time or be pre-generated and recorded in desired format. It is advisable to use the GNU Radio digital signal processing platform to develop software modules for generating real-time radio signals. Using this software it is possible to generate signals with analog and digital types of modulation, including frequency hopping and spread spectrum signals. The maximum bandwidth of the scenario corresponds to the maximum sampling frequency of the transceiver used. With this approach, you can change the power and carrier frequency of all components online. The limitations of real-time signal generation using GNU Radio are due to the fact that the bandwidth and complexity of the signal environment that can be generated depends on the performance of the computer's processor. Wideband scenario can be generated in any format using the high-level Python programming language. Further such files can be used for their analysis offline using appropriate software or as a signal source for radiation by software-defined radio systems.

Keywords: *electronic environment; wideband scenario; complex signal; GNU Radio; Python; Matlab.*