

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Комп’ютерна система тестування алгоритмів обробки сигналів в
електронних комунікаційних системах

Виконав: студент 6 курсу, групи РАм-62
спеціальності 172 Електронні комунікації та
радіотехніка
(шифр і назва спеціальності)

Стащук Д.А.
(підпис) (прізвище та ініціали)

Керівник Дунець В.Л.
(підпис) (прізвище та ініціали)

Нормоконтроль Хвостівська Л.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри Дунець В.Л.
(підпис) (прізвище та ініціали)

Рецензент
(підпис) (прізвище та ініціали)

Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)
Кафедра радіотехнічних систем
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Дунець В.Л.
(підпис) (прізвище та ініціали)
«29» листопада 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)
за спеціальністю 172 Електронні комунікації та радіотехніка
(шифр і назва спеціальності)
Студенту Стащуку Дмитру Анатолійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система тестування алгоритмів обробки сигналів в електронних комунікаційних системах.

Керівник роботи Дунець Василь Любомирович, к.т.н., доцент, завідувач кафедри РТ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» листопада 2024 року № 4/7-1146
2. Термін подання студентом завершеної роботи 27 грудня 2024р.
3. Вихідні дані до роботи Об'єкт дослідження: процес тестування алгоритмів РТС.
Предмет дослідження: імітаційна модель РС.

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ.
Розділ 1. Аналітична частина
Розділ 2. Основна частина
Розділ 3. Науково-дослідна частина
Розділ 4. Охорона праці та безпека в надзвичайних ситуаціях.
Загальні висновки
Перелік посилань
Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	к.ф.-м.н., доцент кафедри ПВ Зелінський І.М.		
Безпека в надзвичайних ситуаціях	к.т.н., завідувач кафедри МТ Окіпний І.Б.		

КАЛЕНДАРНИЙ ПЛАН

Студент		Стащук Д.А.
	(підпис)	(прізвище та ініціали)
Керівник роботи		Дунець В.Л.
	(підпис)	(прізвище та ініціали)

АНОТАЦІЯ

Комп'ютерна система тестування алгоритмів обробки сигналів в електронних комунікаційних системах // Кваліфікаційна робота освітнього рівня «Магістр» // Стащук Дмитро Анатолійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет прикладних інформаційних технологій та електроінженерії, кафедра радіотехнічних систем, група РАМ-62 // Тернопіль, 2024 // С.76, рис. – 50, табл. – 0 , кресл. – 17, додат. – 3, бібліогр. – 15.

Ключові слова: АМ-МОДУЛЯЦІЯ, РАДІОСИГНАЛ, ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ, РАДІОТЕХНІЧНІ СИСТЕМИ, СТОХАСТИЧНІСТЬ, ПЕРІОДИЧНІСТЬ, АДТИВНИЙ ШУМ, АЛГОРИТМИ ОБРОБКИ СИГНАЛІВ.

Кваліфікаційна робота присвячена розробці комп'ютерної імітаційної моделі РС. Уперше створено модель РС протягом одного циклу у формі АМ суми продовжених у часі хвиль РС. З урахуванням періодичності модель реалізовано як АМ-суму хвиль, продовжених на k -і періоди по всій часовій осі.

Ця модель дозволяє, базуючись на заданих параметрах, точно імітувати РС з урахуванням їх повторності та стохастичних факторів, забезпечуючи високу точність відтворення.

ANNOTATION

Computer system for testing signal processing algorithms in electronic communication systems // Qualification work of the educational level “Master” // Stashchuk Dmytro // Ternopil Ivan Puluj National Technical University, Faculty of Applied Information Technologies and Electrical Engineering, Telecommunications and Radio Engineering, group RAm-62 // Ternopil, 2024 // P.76, fig. – 50, tabl. – 0, chair. – 17, annexes. – 3, references – 15.

Keywords: AM MODULATION, RADIO SIGNAL, SIMULATION MODELING, RADIO ENGINEERING SYSTEMS, STOCHASTICITY, PERIODICITY, ADDITIVE NOISE, SIGNAL PROCESSING ALGORITHMS.

The qualification work is devoted to the development of a computer simulation model of radio signals. For the first time, a model of a radio signal within a single period was created in the form of an AM sum of waves of radio signals extended in time. Taking into account the periodicity, the model is realized as an AM-sum of waves extended for k periods along the entire time axis.

This model allows you to accurately simulate radio signals based on the specified parameters, taking into account their repetition and stochastic factors, ensuring high reproduction accuracy.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ

GUI (англ. Graphical User Interface) – Графічний інтерфейс користувача.

PSD (англ. Power Spectral Density) – Спектральна щільність потужності.

АМ – Амплітудна модуляція.

ВЧ – Високі частоти.

ГК – Гармонічне коливання.

ДМХ – Дециметрові хвилі.

ЗС – Земна станція.

КЗ – Канал зв'язку.

МС – Модульований сигнал.

НЧ – Низькі частоти.

ПЗ – Програмне забезпечення.

ПФ – Початкова фаза.

Прд – Передавач.

Прм – Приймач.

ПрП – Приймач-передавач.

РРЗ – Радіорелейний зв'язок.

РРЛ – Радіорелейні лінії.

РС – Радіосигнал.

РТС – Радіотехнічні системи.

РХ – Радіохвиля.

СГ – Сигнал.

СЗ – Супутниковий зв'язок.

СМХ – Сантиметрові хвилі.

СП – Система передачі.

СПІ – Системи передачі інформації.

СР – Супутник-ретранслятор.

СС – Синусоїдальний сигнал.

Тлф – Телефонний.

ТРЗ – Тропосферний радіозв'язок.

ФМ – Фазова модуляція.

ЧМ — Частотна модуляція.

ШСЗ – Штучний супутник Землі.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА.....	11
1.1. Радіозв'язок та радіотехнічні системи зв'язку	11
1.1.1. РРЗ	11
1.1.2. ТРЗ.....	13
1.1.3. СЗ.....	13
1.2. Математичні моделі РС в радіотехнічних системах зв'язку	14
1.2.1. Детерміновані моделі РС	14
1.2.1.1. АМ-РС	15
1.2.1.2. ЧМ та ФМ-РС	18
1.2.2. Статистичні моделі РС	24
1.3. Концепція наукової роботи.....	29
1.4. Висновок до першого розділу.....	29
РОЗДІЛ 2. ОСНОВНА ЧАСТИНА.....	31
2.1. Модель РС в каналах зв'язку із АМ.....	31
2.2. Модель РС в КЗ із АМ під впливом різного роду завад РС в системах зв'язку	36
2.3. Структура РС	37
2.4. Алгоритм імітаційного моделювання РС	38
2.5. Імітаційна модель РС.....	43
2.6. Моделювання РС.....	46
2.7. Висновок до другого розділу	52
РОЗДІЛ 3. НАУКОВО-ДОСЛІДНА ЧАСТИНА	53
3.1. Блок-схема програми імітаційного моделювання РС	53
3.2. Розробка ПЗ	55
3.2.1. Розробка функції генератора РС	56
3.2.2. Розробка GUI для імітаційного моделювання РС	58
3.3. Результати імітаційного моделювання РС	63

3.4.	Висновок до третього розділу	68
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ		69
4.1.	Охорона праці.....	69
4.2.	Безпека в надзвичайних ситуаціях	71
ЗАГАЛЬНІ ВИСНОВКИ		74
СПИСОК ПОСИЛАНЬ.....		76
ДОДАТКИ		

ВСТУП

Актуальність теми. У сучасних умовах ринку спостерігається динамічний розвиток РТС та їх активне впровадження у різні сфери. Процес проєктування РТС охоплює етапи експериментального налаштування та тестування, які реалізуються шляхом перевірки алгоритмів обробки РС [1].

Натурні дослідження РТС ускладнені інтерпретацією одержаних результатів, потребують значних витрат часу та ресурсів.

Метод імітаційного моделювання дозволяє якісно дослідити роботу РТС з мінімальними затратами.

Науковий аналіз свідчить, що моделювання базується на імітаторах, які використовують матмоделі РС у КЗ (емпіричні, детерміновані, статистичні):

- Емпіричні моделі: Реально описують СГ, апроксимують експериментальні дані, але не враховують широкий спектр умов.

- Детерміновані моделі: Високоточні, враховують фундаментальні аспекти поширення СГ (Соколова А.В., Борзова А.Б. тощо), але ігнорують стохастичність.

- Статистичні моделі: Моделюють СГ як стохастичні процеси (Введенський Б.А., Clarke R.H. тощо), однак не відображають властивості періодичності.

Наукова задача. Розробити імітаційну модель РС, що враховує стохастичність та періодичність.

Мета і задачі дослідження: створення імітаційної моделі для тестування алгоритмів роботи РТС. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Проаналізувати існуючі моделі.
- Розробити моделі з урахуванням стохастичності та періодичності.
- Імітаційне моделювання СГ.
- Створення ПЗ для моделювання.

Об’єкт дослідження. Процес тестування алгоритмів РТС.

Предмет дослідження. Імітаційна модель РС.

Методи дослідження. Використання теорії стохастичних процесів і MATLAB 2023a.

Практичне значення одержаних результатів. Розроблена імітаційна модель РС забезпечує можливість:

- Ефективного тестування алгоритмів обробки РС без необхідності проведення натурних експериментів, що значно знижує фінансові та часові витрати.

- Високоточного моделювання СГ різної структури для прогнозування роботи РТС у реальних умовах.

- Використання у навчальному процесі для підготовки фахівців у галузі РТС та цифрової обробки СГ.

- Розширення можливостей ПЗ для аналізу імітованих СГ з врахуванням реальних умов передачі, включаючи стохастичні і періодичні фактори.

Застосування розробки є доцільним у дослідницьких центрах, що займаються розробкою та тестуванням РТС, а також у промислових організаціях, які спеціалізуються на створенні обладнання для радіозв'язку та радіолокації.

Наукова новизна отриманих результатів. Розроблено модель у вигляді АМ-періодичних сум синусоїд із експоненційним затуханням, що враховує стохастичність амплітуд і тривалостей. Модель забезпечує високу точність відтворення експериментальних СГ для тестування алгоритмів РТС.

РОЗДІЛ 1.

АНАЛІТИЧНА ЧАСТИНА

1.1. Радіозв'язок та радіотехнічні системи зв'язку

Радіозв'язок – це тип безпроводного зв'язку, в якому носієм СГ є РХ, що вільно поширюються у просторі. Передача СГ здійснюється наступним чином:

- На Прд формується СГ із заданими параметрами (частота, амплітуда).
- Цей СГ модулює ВЧ-коливання (несучу), яке виконує функцію носія.
- МС випромінюється антеною в простір.

На Прм РХ збуджують МС у приймальній антені. Після цього СГ проходить через демодуляцію та фільтрування за допомогою НЧ-фільтра, який видаляє ВЧ-складову (несучу), залишаючи корисний СГ. Прийнятий СГ може дещо відрізнитися від переданого через спотворення, спричинені перешкодами та наведеннями.

1.1.1. РРЗ

РРЗ здійснюється по РРЛ, що складається з ПрП-станцій (ретрансляторів). Зазвичай РРЗ працює у діапазонах ДМХ і СМХ (сотні МГц – десятки ГГц).

Класифікація РРЛ за частотами:

- Місцеві лінії: 0,39-40,5 ГГц.
- Внутрішньозонові лінії: 1,85-15,35 ГГц.
- Магістральні лінії: 3,4-11,7 ГГц.

Цей поділ базується на впливі середовища поширення на надійність зв'язку.

- До 12 ГГц: атмосферні явища майже не впливають.
- 15-40 ГГц: вплив стає значним.

- >40 ГГц: визначальними є загасання в газах атмосфери (кисень, водяна пара).

Атмосферні втрати:

- Атмосфера майже повністю втрачає прозорість на частоті 118,74 ГГц через резонансне поглинання молекулами кисню.

- На частотах понад 60 ГГц затухання становить більше 15 дБ/км..

- У водяній парі рівень ослаблення визначається вологістю і стає значущим на частотах до 45 ГГц.

Вплив гідрометеорів:

- Опади, такі як дощ, сніг, туман або град, суттєво впливають на передачу СГ на частотах понад 6 ГГц.

- За несприятливих умов, зокрема за наявності смогу, кислотних чи металевих домішок, вплив може відчуватися навіть на низьких частотах.

Розташування антен:

- Антени сусідніх станцій повинні знаходитися в зоні прямого зорового контакту.

- Для збільшення відстані між станціями антени монтують на вежах заввишки 10-100 м (радіус видимості становить 40-50 км) або на спорудах значної висоти.

- Станції можуть бути як стаціонарними, так і пересувними.

Особливості РРЗ:

- Двосторонній зв'язок: передача й прийом даних здійснюються одночасно, але на різних частотах.

- Довжина лінії: до 10 000 км.

- Ємність:

- Аналогові лінії – до кількох тисяч каналів.

- Цифрові лінії – до 622 Мбіт/с.

- Протяжність проти швидкості передачі: вони перебувають у взаємозв'язку, де зростання одного параметра призводить до зменшення іншого – більша відстань знижує швидкість даних під час передачі і навпаки.

1.1.2. ТРЗ

ТРЗ базується на явищі повторного випромінювання електромагнітної енергії в електричній неоднорідній тропосфері під час поширення в ній РХ. Він функціонує у діапазонах ДМХ і СМХ.

Нерівномірність тропосфери викликана стохастичними локальними змінами діелектричної проникності ϵ , що виникають через коливання температури, атмосферного тиску та вологості повітря, а також через систематичне зменшення цих параметрів із підняттям на висоту. Повторне випромінювання енергії відбувається в області, де перетинаються діаграми спрямованості Прд та Прм антен.

Відстань між точками Прд та Прм може сягати до 1000 км. Втім, у реальних умовах частіше використовують РРЛ, де ТРЗ задіяний повністю або лише в окремих ділянках. Загальна довжина таких ліній може досягати кількох тисяч км.

Сучасні ПрП системи забезпечують:

- 120-240 Тлф-каналів в одному ВЧ-стовбурі на відстані $R=150-250$ км.
- 12 Тлф-каналів при $R=800-1000$ км.

Передача телефізійного СГ можлива лише за умови $R < 150-200$ км. Однак через надходження до Прм множини хвиль із різними затримками якість передачі виявляється недостатньо високою.

1.1.3. СЗ

СЗ – вид радіозв'язку, в якому ШСЗ використовуються як ретранслятори.

Зв'язок здійснюється між ЗС, які можуть бути як стаціонарними, так і мобільними.

СЗ є еволюцією традиційного РРЗ, де ретранслятор розташовується на великій висоті. Завдяки цьому зона покриття охоплює майже половину поверхні Землі, що усуває потребу в численних ретрансляторах. У більшості випадків достатньо одного супутника.

На початкових етапах використовувалися пасивні СР, які були простими відбивачами РС (зазвичай сферичними об'єктами з металевим покриттям). Такі супутники не містили ПрП-апаратури і не стали популярними.

Сучасні СЗ є активними і обладнані електронними системами для прийому, обробки, посилення та ретрансляції РС.

Типи СР:

1. Нерегенеративні супутники:

- Приймають РС, змінюють його частоту, підсилюють і передають на іншу ЗС.

- Можуть використовувати кілька незалежних каналів для обробки РС. Такі канали, кожен з яких працює з певною частиною спектру, називаються транспондерами.

2. Регенеративні супутники:

- Демодулюють прийнятий РС, обробляють його і модулюють заново.
- Завдяки цьому виправлення помилок відбувається двічі: на супутнику та на Прм-станції.

- Недоліки:

- Висока складність конструкції, що значно підвищує вартість супутника.

- Збільшена затримка передачі РС.

СЗ широко використовується для передачі даних, мовлення, інтернет-з'єднання, а також у військових та наукових цілях.

1.2. Математичні моделі РС в радіотехнічних системах зв'язку

1.2.1. Детерміновані моделі РС

У сучасних ідеальних КЗ використовуються моделі РС із застосуванням АМ і ФМ.

1.2.1.1. АМ-РС

Модель АМ-СГ будується шляхом зміни амплітуди НК пропорційно до МС:

$$A_m(t) = A_{m0} + k \cdot u(t), \quad (1.1)$$

де:

A_{m0} – амплітуда НС (немодульованого);

k – коефіцієнт пропорційності;

$u(t)$ – МС.

ПФ АМ-СГ змінюється лінійно в часі, як і фаза НК:

$$\psi = \omega_0 t + \varphi_0, \quad (1.2)$$

де:

– ω_0 – кутова частота НС;

– φ_0 – ПФ.

Розгорнута модель РС:

$$S_{AM}(t) = A_m(t) \cdot \cos(\omega_0 t + \varphi_0), \quad (1.3)$$

З цього виразу видно, що АМ-СГ є добутком огибаючої $A_m(t)$ та ВЧ-коливання $\cos(\omega_0 t + \varphi_0)$.

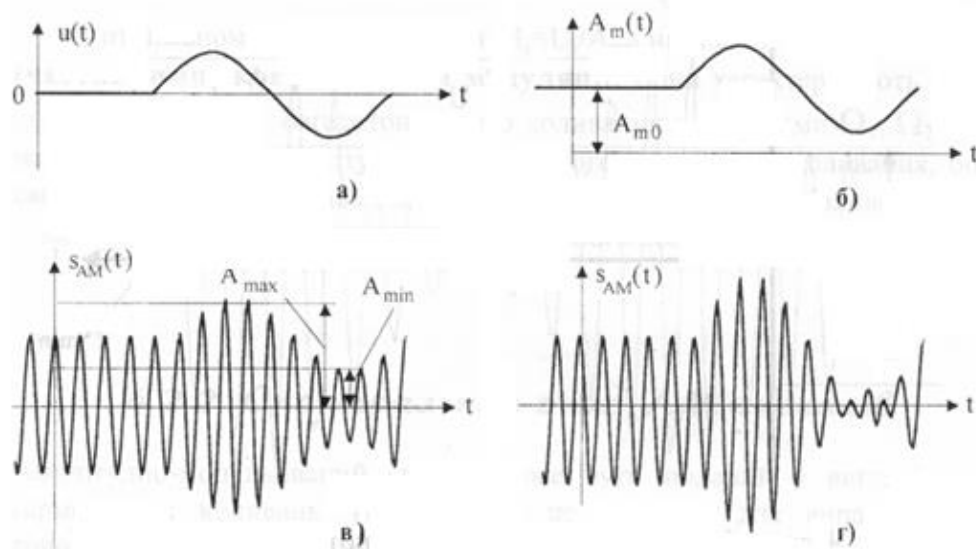


Рис. 1.1. – (а) – МС $u(t)$, (б) – Обгинаюча амплітуди $A_m(t)$, (в) – АМ-СГ за умови $|ku(t)|_{\max} < A_{m0}$, (г) – АМ-СГ за умови $|ku(t)|_{\max} > A_{m0}$

На рисунку 1.1 показано, що при $|ku(t)|_{\max} < A_{m0}$ між миттєвими значеннями МС $u(t)$ і огибаючої $A_m(t)$ існує однозначний пропорційний зв'язок. Якщо ця умова не виконується (див. рисунок 1.1, г), виникає небажане явище – спотворення огибаючої АМ-СГ, через що форми $u(t)$ і $A_m(t)$ більше не збігаються.

У найпростішому варіанті одотональної модуляції МС має вигляд ГК з частотою $\Omega \ll \omega_0$ та ПФ Φ_0 :

$$k \cdot u(t) = U_{\Omega} \cos(\Omega t + \Phi_0), \quad (1.4)$$

Матмодель АМ-СГ для цього випадку записується як:

$$S_{AM}(t) = A_{m0} \left[1 + \frac{U_{\Omega}}{A_{m0}} \cos(\Omega t + \Phi_0) \right] \cos(\omega_0 t + \varphi_0) \quad (1.5)$$

Графік цього ГК наведено на рисунку 1.2.

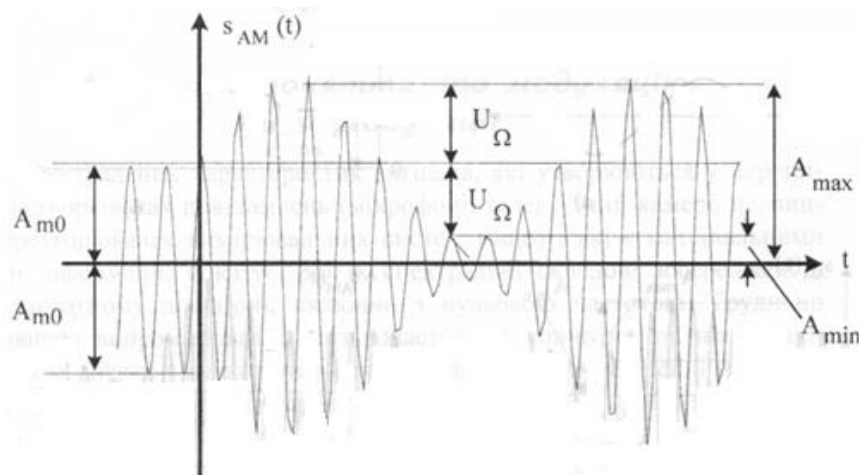


Рис. 1.2. – Графік зміни АМ-СГ в часовій області для однотонольної модуляції

Відношення $M = U_{\Omega} / A_{m0}$ називається КМ. Для уникнення спотворень модуляції необхідно, щоб ця величина задовольняла умову: $0 \leq M \leq 1$.

Амплітуда коливання змінюється в межах:

- Мінімальна: $A_{\min} = A_{m0}(1 - M)$.
- Максимальна: $A_{\max} = A_{m0}(1 + M)$.

КМ також можна визначити через $A_{\min}$ та $A_{\max}$:

$$M = (A_{\max} - A_{\min}) / (A_{\max} + A_{\min}). \quad (1.6)$$

Цей спосіб обчислення M часто застосовують на практиці для аналізу реалізації АМ-СГ.

У загальнішому випадку, коли МС складається з N гармонічних складових:

$$k \cdot u(t) = \sum_{i=0}^N U_i \cos(\Omega_i t + \Phi_i).$$

Матмодель МС набуває вигляду:

$$\begin{aligned} S_{AM}(t) &= \left[A_{m0} + \sum_{i=1}^N U_i \cos(\Omega_i t + \Phi_i) \right] \cos(\omega_0 t + \varphi_0) = \\ &= A_{m0} \left[1 + \sum_{i=1}^N M_i \cos(\Omega_i t + \Phi_i) \right] \cos(\omega_0 t + \varphi_0). \end{aligned} \quad (1.7)$$

В отриманому виразі $M_i = U_i / A_{m0}$ називаються парціальними КМ, які характеризують вплив кожної гармонічної складової МС з частотами $\Omega_1, \Omega_2, \dots, \Omega_N$ на загальну амплітуду МС.

ГК, описане формулою 1.7, називається складно-модульованим. Формула 1.5 є окремим випадком складно-модульованого коливання для $N = 1$.

1.2.1.2. ЧМ та ФМ-РС

У АМ-СГ амплітуда несучого СС модулюється інформаційним СГ $f(t)$, тому зміни величини амплітуди несучого СГ передають інформацію з $f(t)$. Оскільки СС характеризується трьома параметрами – амплітудою, частотою і фазою – відкривається можливість передати ту саму інформацію шляхом зміни частоти або фази несучої.

СС, за формулюванням, є безмежним коливанням із фіксованими амплітудою, частотою та фазою. Для моделювання реальних РС необхідно узагальнити поняття СС, дозволивши змінювати його амплітуду, частоту і фазу в часі. Поняття змінюваної амплітуди було впроваджено раніше при вивченні АМ. Розглянемо змінність частоти та фази.

Для розуміння процесу зміни частоти, введемо поняття МЧ. На рисунку 1.3 (а) зображено СС $\varphi(t)$, частота якого ω залишається сталою для $t < T$. У момент $t = T$ частота різко збільшується до 2ω , зберігається до $t = 2T$, а потім

повертається до ω_0 . Таким чином, $\varphi(t)$ є СС, що змінює частоту між ω_0 і $2\omega_0$ залежно від інтервалів часу:

- ω_0 для $2nT < t < (2n+1)T$.
- $2\omega_0$ для $(2n+1)T < t < (2n+2)T$, де n – цілочисельне значення.

Дослідимо випадок плавної коливання частоти, як наведено на рисунку 1.3 (г). Частота РС змінюється плавно та зі сталою швидкістю у межах ω_0 до $2\omega_0$ протягом інтервалу T . У цьому випадку в кожен момент часу МЧ є різною. Якщо говорити точно, РС $\varphi(t)$, який зображено на рисунку 1.3 (в) не можна представити у вигляді стандартного гармонійного коливання.

Для РС із поступовою зміною частоти вводиться узагальнене визначення СС:

$$\varphi(t) = A \cos \theta(t) , \quad (1.8)$$

де θ – фаза СС, яка є функцією часу.

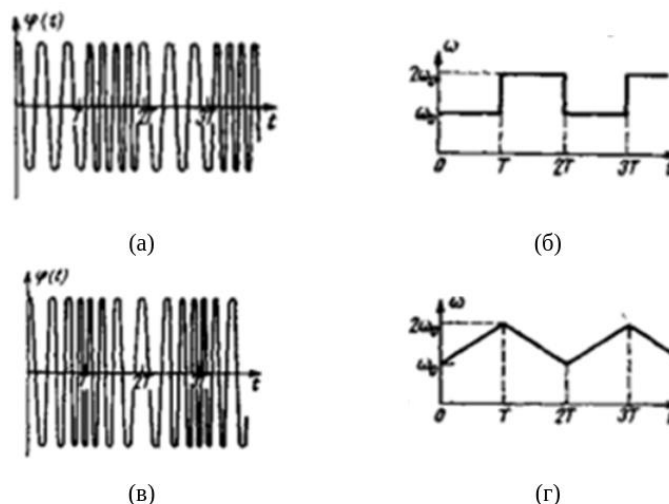


Рис. 1.3. – До поняття МЧ СГ: (а) – РС із раптовим переходом частоти; (б) – раптова зміна частоти; (в) – РС із поступовим варіюванням частоти; (г) – плавне варіювання частоти

У випадку РС зі сталою частотою (див. формулу 1.9):

$$\varphi(t) = A \cos(\omega_0 t + \theta_0), \quad (1.9)$$

Отже, $\theta(t) = \omega_0 t + \theta_0$ і $\omega_0 = d\theta/dt$.

У цьому випадку частота ω_0 залишається незмінною і визначається як похідна від фазового кута $\theta(t)$. Загалом, ця похідна може бути змінною. Введемо поняття МЧ $d\theta/dt$ (змінна у часі). Співвідношення між кутом $\theta(t)$ і МЧ ω_i виражається так:

$$\omega_i = d\theta/dt, \quad \theta = \int \omega_i dt \quad (1.10)$$

З огляду на це можна зрозуміти принцип передачі інформації, що перебуває у повідомленні $f(t)$, шляхом варіювання фази несучого СГ θ . Модуляція, під час якої відбувається зміна фази несучого СГ залежно від функції $f(t)$, називається КМ.

У КМ зазвичай застосовуються два основні типи: ФМ та ЧМ.

Якщо фазовий кут $\theta(t)$ змінюється пропорційно МС, тоді $f(t)$, тоді:

$$\theta(t) = \omega_0 t + \theta_0 + k_p f(t). \quad (1.11)$$

де k_p – константа пропорційності. Таку зміну фазового кута називають ФМ.

У цьому випадку РС: $A \cos[\omega_0 t + \theta_0 + k_p f(t)]$ є ФМ-НС.

МЧ ω_i РС ФМ визначається за формулою 1.12.

$$\omega_i = d\theta / dt = k_p df / dt \quad (1.12)$$

Це означає, що при ФМ МЧ змінюється прямо пропорційно швидкості зміни МС $f(t)$.

Якщо МЧ змінюється відповідно до самого МС $f(t)$, то така модуляція називається частотною. Для СГ з ЧМ:

$$\omega_i = \omega_0 + k_p f(t) \quad (1.13)$$

де ω_i – МЧ.

$$\theta(t) = \int \omega_i dt = \omega_0 t + k_f \int f(t) dt + \theta_0 \quad (1.14)$$

Таким чином, ЧМ-СГ можна представити у вигляді:
 $A \cos[\omega_0 t + \theta_0 + k_f \int f(t) dt]$.

Звідси випливає, що ФМ і ЧМ є різними типами КМ, але їхня відмінність полягає у способі зміни фази. При ФМ фаза змінюється пропорційно модулюючій функції $f(t)$, а при ЧМ – пропорційно інтегралу цієї функції.

Якщо проінтегрувати модулюючу ф-цію $f(t)$, а потім використати отримане коливання для ФМ несучої, утворюється ЧМ-РС. Аналогічно, якщо продиференціювати $f(t)$ та застосувати це коливання для модуляції частоти, отримуємо ФМ-РС.

Під час формування РС ЧМ одним із можливих способів є інтегрування $f(t)$ з подальшою ФМ несучої. У результаті ЧМ та ФМ тісно пов'язані: будь-яка зміна фази несучої спричиняє зміну частоти і навпаки.

Зважаючи на цю взаємозалежність, у подальшому розглядається лише ЧМ, оскільки всі отримані висновки справедливі й для ФМ. Важливо зазначити, що в КМ амплітуда РС залишається постійною.

Приклади РС ЧМ наведено на рисунку 5.1 (а, в).

Якщо позначити РС ЧМ й ФМ як $\varphi_{\text{ЧМ}}(t)$ та $\varphi_{\text{ФМ}}(t)$ відповідно, тоді:

$$\begin{cases} \varphi_{\text{ФМ}}(t) = A \cos[\omega_0 t + k_p f(t)] \\ \varphi_{\text{ЧМ}}(t) = A \cos[\omega_0 t + k_f \int f(t) dt] \end{cases}, \quad (1.15)$$

де $f(t)$ – модулююче повідомлення. ПФ θ_0 у формулах приймається рівною нулю без втрати загальності.

При ЧМ амплітуда НС залишається постійною $A_{m0} = \text{const}$, а МС $u(t)$ керує ПФ $\psi(t)$.

Є два основні варіанти модуляції:

– ЧМ.

У цьому випадку МЧ змінюється пропорційно МС щодо частоти НК:

$$\omega_0(t) = \omega_0 + k u(t). \quad (1.16)$$

ПФ $\psi(t)$ змінюється лінійно пропорційно інтегралу від МС:

$$\psi(t) = \int_0^t \omega(t) dt = \omega_0 t + \varphi_0 + k \int_0^t u(t) dt + \varphi_0. \quad (1.17)$$

Цей вид модуляції називається ЧМ.

Матмодель ЧМС:

$$s_{\text{ЧМ}}(t) = A_{m0} \cos \left[\omega_0 t + \varphi_0 + k \int_0^t u(t) dt \right]. \quad (1.18)$$

– ФМ.

Тут ПФ змінюється пропорційно МС щодо лінійного закону:

$$\psi(t) = \omega_0 t + \varphi_0 + ku(t). \quad (1.19)$$

З рівняння (1.19) випливає, що кутова МЧ змінюється пропорційно похідній МС відносно частоти ω_0 НК:

$$\omega(t) = \frac{d\psi(t)}{dt} = \omega_0 + k \frac{du(t)}{dt}. \quad (1.20)$$

Цей тип модуляції називається ФМ.

Матмодель ФМС:

$$s_{\text{ФМ}}(t) = A_{m0} \cos [\omega_0 t + \varphi_0 + ku(t)]. \quad (1.21)$$

ЧМС і ФМС тісно пов'язані, оскільки зміна частоти викликає зміну ПФ, і навпаки. У загальному випадку за осцилограмами неможливо точно визначити, чи це ЧМС, чи ФМС. Різниця стає очевидною, коли МС $u(t)$ змінюється стрибкоподібно: у ФМС спостерігаються стрибки миттєвих значень РС.

На рисунку 1.4 наведено осцилограми ЧМС і ФМС при гармонічній (а, б, в) і стрибкоподібній (г, д, е) зміні МС.

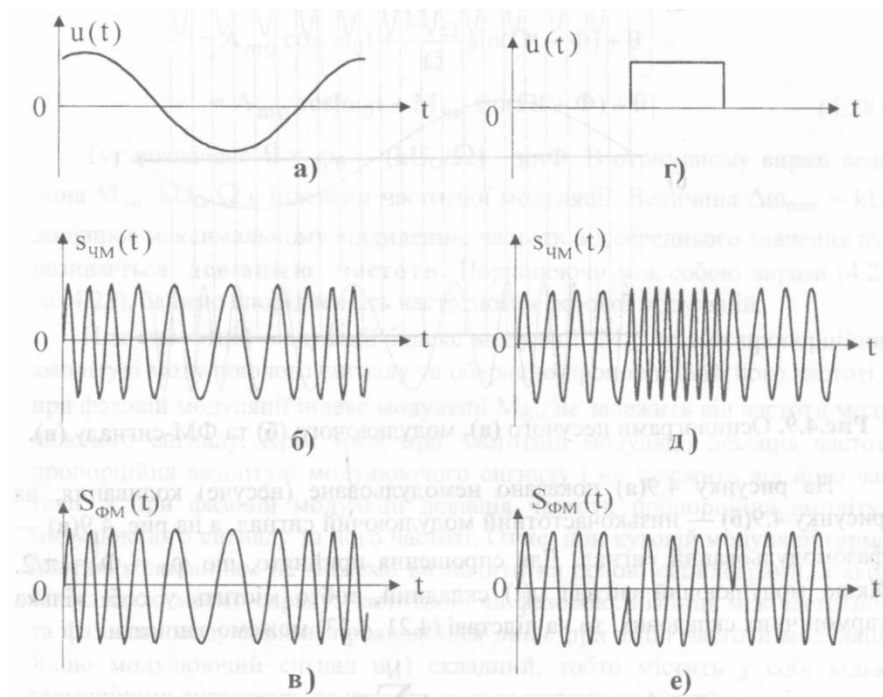


Рис. 1.4. – Осцилограми МС та ЧМ й ФМ-коливань при гармонічній (а, б, в) та стрибкоподібній (г, д, е) зміні МС.

Різниця між ЧМ і ФМ проявляється лише під час зміни частоти модуляції.

1.2.2. Статистичні моделі РС

При проектуванні РТС зв'язку (див. рисунок 1.5) КП РС зазвичай є фіксованою ланкою, до якої мають бути узгоджені джерела СГ і функціональні блоки.

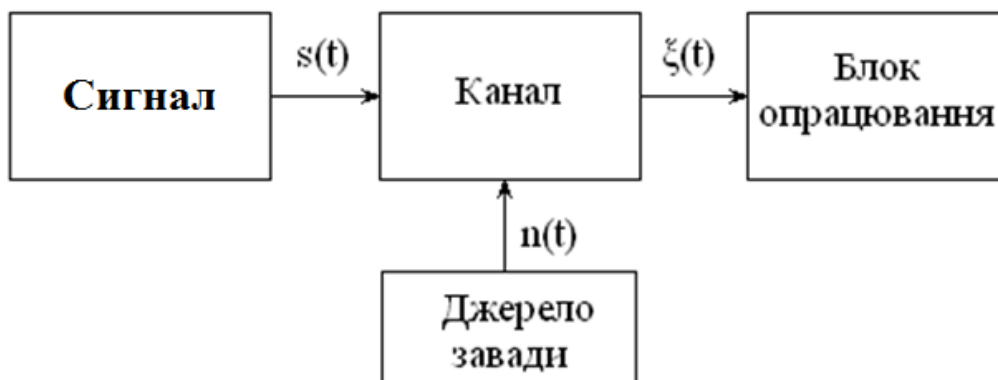


Рис. 1.5. – Загальний принцип проектування РТС зв'язку

Для аналізу можливих варіантів проектування таких РТС, передачі РС та прогнозування їх характеристик без проведення прямих експериментальних випробувань необхідно володіти детальними характеристиками РС.

Опис РС, який дозволяє розрахувати або оцінити будь-які його параметри, називається моделлю РС. Важливо відрізнити модель від часткового опису РС, що включає лише окремі характеристики, необхідні для вирішення конкретних завдань.

Основні вимоги до моделі:

- Простота – модель має бути зручною для використання.
- Точність – модель повинна відповідати експериментальним даним.

Однак реальні РС є складними, тому ці вимоги часто суперечать одна одній. Під час розробки моделі необхідно знайти розумний компроміс між її простотою та точністю.

Модель реального РС, якщо не враховувати внутрішні процеси в системі, зводиться до матопису РС на вході та виході каналу та зв'язків між ними.

Зв'язок між вхідним РС $s(t)$ та вихідним РС $\xi(t)$ можна описати через системний оператор:

$$\xi(t) = L\{s(t)\}. \quad (1.22)$$

Для моделювання КЗ потрібно визначити область V_x функціонального простору, яка відповідає допустимим вхідним РС.

Ця область описує характер РС, які можуть бути:

- Безперервними.
- Дискретними.
- Цифровими.
- Детермінованими.
- Стохастичними.

Аналогічно, необхідно задати область V_y для допустимих вихідних РС.

Матмодель РС – це сукупність:

- Системного оператора.
- Областей допустимих РС V_X та V_Y .

Класифікувати РС можна за властивостями їх матмоделей:

- Стаціонарні РС: параметри постійні в часі.
- Нестационарні РС: параметри змінюються залежно від моменту подачі вхідного РС.

Реальні КЗ в РТС зазвичай мають певну міру нелінійності. Однак більшість з них добре описуються лінійними моделями.

Однією з найпростіших моделей є РС з адитивним шумом. Такий РС описується у ЛК та проілюстрований на рисунку 1.6.

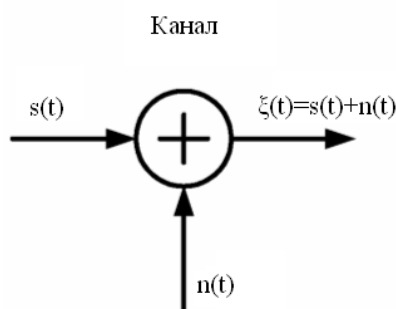


Рис. 1.6. – Модель СГ в КЗ СП РС

У даній моделі РС $s(t)$ піддається впливу лише адитивного шуму $n(t)$, описаного адитивною моделлю:

$$\xi(t) = s(t) + n(t). \quad (1.23)$$

Фізично адитивний шум зумовлений зовнішніми електричними перешкодами, елементами електроніки, підсилювачами системи тощо.

Коли шум переважно спричинений електронними компонентами й підсилювачами, його описують як БШГ.

Відповідно, така матмодель отримала назву "РС з АШГ". Завдяки своїй універсальності для фізичних КЗ й простоті матопису, вона стала стандартом у дослідженнях СП даних.

БШ – це стаціонарний стохастичний процес $n(i\Delta t)$, чия АКФ виражається через δ -функція Дірака. PSD $W(f) = \sigma^2$ є сталою й дорівнює дисперсії $n(i\Delta t)$. Таким чином, усі частоти БШ мають однакову PSD. Це ідеалізований процес з нескінченною енергією. У реальних умовах, при обмеженні спектру шуму частотним діапазоном, така ідеалізація дозволяє створювати ефективні фільтраційні алгоритми.

У радіотехніці та телекомунікаціях БШ застосовується для моделювання шумів за умови, що ефективна ширина спектру РС B_s значно менша від ширини спектру шуму B_n , а PSD шумів змінюється несуттєво в діапазоні частот РС. Термін БШ описує лише спектральні характеристики стохастичного процесу – будь-який процес із рівномірним спектром підпадає під це визначення незалежно від характеру розподілу.

Якщо спектр РС та шумів обмежений частотним діапазоном $0-B$, то PSD шумів має вигляд:

$$W_n(f) = \sigma^2, \quad 0 \leq f \leq B, \quad W_n = 0, \quad f > B. \quad (1.24)$$

КФ шумів описується формулою:

$$R_n(\tau) = \sigma^2 B \sin(2\pi B\tau) / 2\pi B_\tau. \quad (1.25)$$

Ефективний інтервал кореляції:

$$T_k = 2 \int_0^\infty |R_n(\tau)| d\tau / R_n(0). \quad (1.26)$$

Практичний інтервал кореляції визначають шириною головного максимуму функції $R_n(\tau)$, де сконцентрована основна частина енергетичного спектра шуму. При цьому $T_k = 1/B$, а $BT_k = 1$.

Із цих формул, а також на основі рисунку 1.7 наведено, що звуження частотного діапазону шумів призводить до їхньої кореляції. Чим менший діапазон частот, тим більший радіус кореляції шумів.

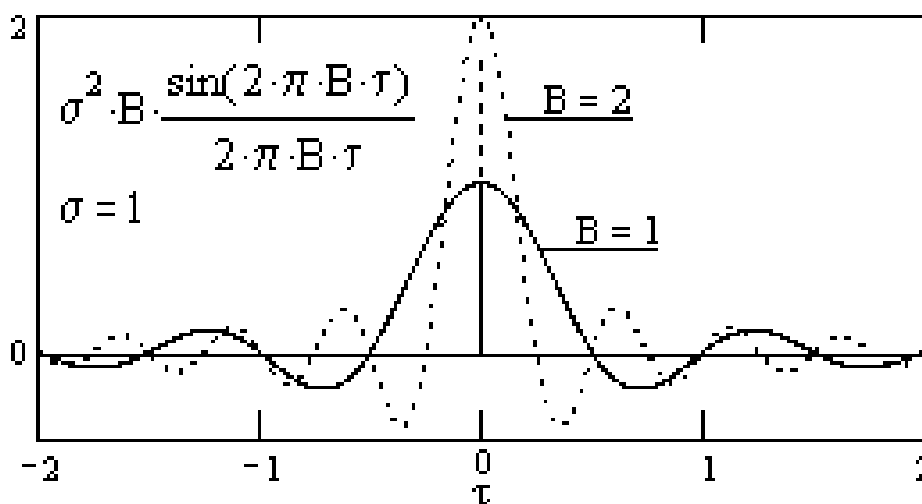


Рис. 1.7. – КФ БШ в частотному діапазоні 0-B.

Обмеження шуму частотним діапазоном рівнозначне його фільтрації смуговим фільтром відповідної ширини. КФ імпульсного відгуку такого фільтру згортатиметься з δ -функцією БШ.

Дискретну модель БШ $n(i\Delta t)$ можна описати як часову стохастичну послідовність δ -імпульсів $\delta(k\Delta t)$ з стохастичними амплітудами a_k :

$$n(i\Delta t) = \sum_{k=1}^N a_k \delta(i\Delta t - k\Delta t) \quad (1.27)$$

яка відповідає умовам стат. однорідності: стабільне середнє число імпульсів за од. часу та статистична незалежність кожного імпульсу від попереднього.

Такий імпульсний потік, званий пуассонівським, є некорельованим і має PDS з рівномірним розподілом:

$$W_n(\omega) = c^2 = N\sigma_a^2. \quad (1.28)$$

де N – кількість імпульсів у проміжку T , σ_a^2 – дисперсія амплітуд.

1.3. Концепція наукової роботи

Існуючі детерміновані підходи до моделювання РС у КЗ не враховують властивість стохастичності, яка є типовою для СГ у реальних РТС.

Статистичні моделі РС, ураховуючи властивість стохастичності, широко застосовуються при проектуванні алгоритмів функціонування РТС.

Реальні РС за своєю природою мають циклічний характер і є стохастичними через вплив як внутрішніх, так і зовнішніх умов. Таким чином, відомі й використовувані статистичні моделі не дозволяють повноцінно створити ефективні алгоритми для функціонування РТС з урахуванням усіх характеристик реальних РС.

Розробка нової моделі РС, яка враховує як морфологічні параметри, так і характеристики циклічності й стохастичності, залишається актуальною проблемою при аналізі алгоритмів обробки сигналів у реальних РТС.

1.4. Висновок до першого розділу

В першому розділі кваліфікаційної роботи проаналізовано наукові роботи різних авторів щодо досліджуваної проблематики.

Відомі детерміновані моделі РС у КЗ не враховують фактор стохастичності, що характерний для реальних РТС.

Статистичні моделі РС враховують стохастичність і широко застосовуються при розробці алгоритмів для РТС.

Реальні РС характеризуються періодичністю та стохастичною природою, яка обумовлена впливом внутрішніх і зовнішніх факторів. У зв'язку з цим існуючі статистичні моделі не забезпечують можливості повного створення ефективних алгоритмів, що враховують усі параметри сигналогенераторів.

Розробка нових моделей РС, які враховують морфологічні особливості, періодичність і стохастичні характеристики, є важливим науковим завданням для вдосконалення алгоритмів обробки реальних РТС.

РОЗДІЛ 2. ОСНОВНА ЧАСТИНА

Опис РС за допомогою матмоделей на кожному етапі аналізу базується на необхідності відображення ключових закономірностей об'єктів і явищ, що досліджуються, з їх подальшою формалізацією у матвигляді. Це дає змогу проводити теоретичне апіорне моделювання, здійснювати експерименти з РТС, обґрунтовувати алгоритми та інтерпретувати результати обробки РС.

2.1. Модель РС в каналах зв'язку із АМ

Під час розробки СПІ зазвичай виявляється, що спектр СГ, який необхідно передати, не збігається з частотами, які ефективно пропускає заданий КЗ. Також часто виникає потреба у одночасній передачі кількох СГ. Один із методів вирішення цієї проблеми – застосування ЧРК, коли різні СГ займають окремі частотні смуги, що не накладаються одна на одну.

Часто необхідно, щоб переданий РС був вузькосмуговим. Це означає, що ефективна ширина спектра: $\Delta f \ll f_0$.

Вказані причини обумовлюють потребу у трансформації РС таким способом, щоб ця смуга частот задовольняла задані вимоги, а на приймальній стороні СГ міг бути відновлений без спотворень.

Розглянемо часове та спектральне представлення АМ-СГ. На рисунку 2.1 показано: (а) СГ-носій $s_n(t)$; (б) інформаційний СГ $s_i(t)$; (в) АМС $s_{AM}(t)$.

Аналіз АМ-СГ зручно виконувати через мат. модель. Побудуємо модель для цього АМ-СГ. Вираз для СГ-носія задається формулою (2.1):

$$\omega_n = 2\pi f = \frac{2\pi}{T}, \quad (2.1)$$

де U_m – амплітуда СГ, ω_n – частота носія, φ_0 – ПФ.

Як інформаційний СГ використаємо гармонічний:

$$s_i(t) = U_\Omega \cos \Omega t, \quad (2.2)$$

де U_Ω – амплітуда, Ω – частота інформаційного СГ. При цьому $\Omega \ll \omega_n$.

АМ-СГ описується формулою (2.3), з урахуванням $\varphi_0 = 0$:

$$s_{AM}(t) = U(t) \cos \omega_n t \quad (2.3)$$

Огинаюча СГ:

$$U(t) = U_m + U_\Omega \cos \Omega t \quad (2.4)$$

Щоб уникнути спотворень при демодуляції, амплітуда носія повинна перевищувати амплітуду інформаційного СГ. Тоді огинаюча завжди залишатиметься вище осі часу t .

Підставляючи (2.4) у (2.3), отримуємо модель АМ-СГ:

$$s_{AM}(t) = (U_m + U_\Omega \cos \Omega t) \cos \omega_n t \quad (2.5)$$

СГ-носій – це гармонічний СГ із частотою, значно більшою за частоту спектра інформаційного СГ, як наведено на рисунку 2.1.

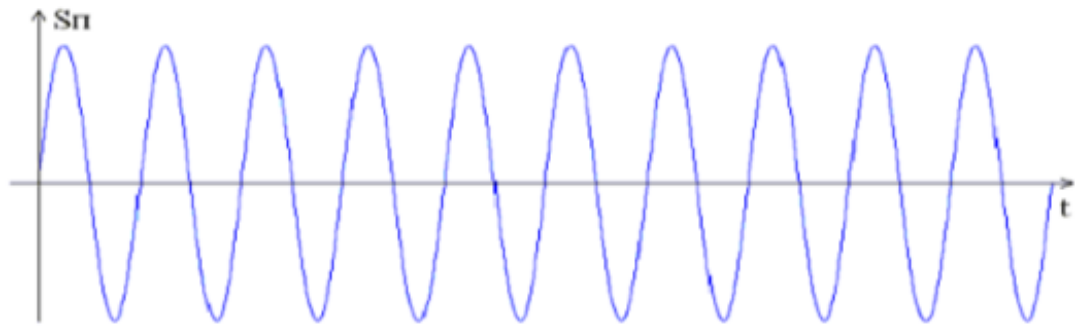


Рис. 2.1. – СГ-носій

Інформаційний СГ – безперервний (аналоговий), у цьому прикладі гармонічний СГ (див. рисунок 2.2).

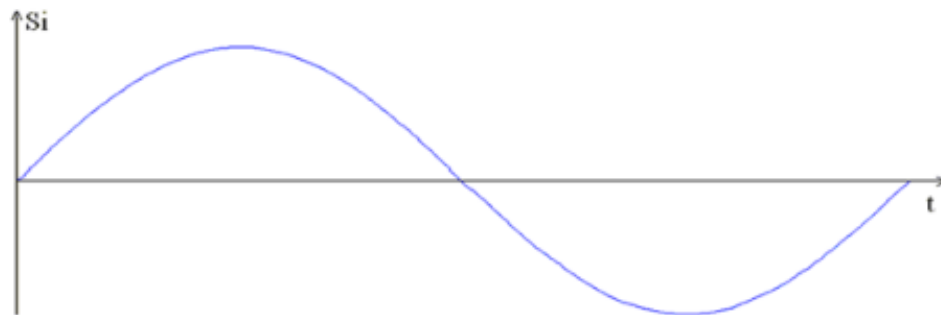


Рис. 2.2. – Інформаційний СГ

АМ-СГ формується так, що амплітуду носія змінює форма інформаційного СГ, як це наведено на рисунку 2.3.

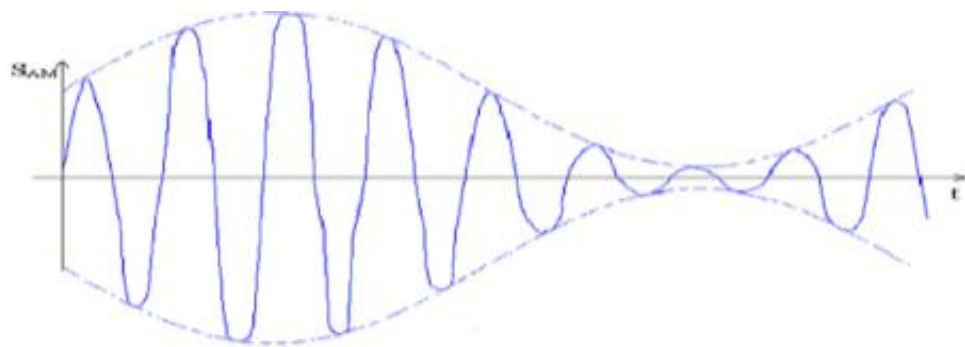


Рис. 2.3. – Формування АМ-СГ

Зазначимо, що під час накладання інформ. СГ на гармонічний СГ-носій відбувається перенесення спектру інформ. СГ в заданий частотний діапазон. Центральна частота цього діапазону збігається з частотою гармонічного носія.

Для уникнення спотворень при демодуляції необхідно врівноважити амплітуди інформ. СГ й СГ-носія. З цією метою до моделі (2.4) додається пропорційний коефіцієнт k , тобто:

$$s_{AM}(t) = (U_m + kU_{\Omega} \cos \Omega t) \cos \omega_{\Pi} t, \quad U_{\Omega} < U_m \quad (2.6)$$

Графічно спектри інформ. СГ та носія наведено на рисунку 2.4.

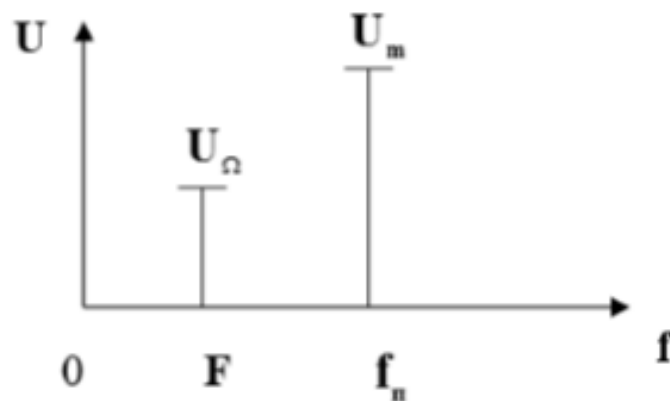


Рис. 2.4. – Частотне представлення: F – частота інформ. СГ, f_n – частота СГ-носія

Щоб "проаналізувати" спектр АМ-СГ, скористаємося його мат. моделлю (формула 2.6). Необхідно трансформувати цей вираз у вигляд, що представляє суму функцій типу $\sin/\cos$. Виконаємо перетворення формули (2.6):

$$s_{AM}(t) = U_m \left(1 + \frac{kU_{\Omega}}{U_m} \cos \Omega t \right) \cos \omega_{\Pi} t \quad (2.7)$$

Введемо параметр модуляції $M = \frac{kU_{\Omega}}{U_m}$, який є коефіцієнтом глибини модуляції АМ-СГ.

Для подальших перетворень виразу (2.7) застосуємо тригонометричну формулу:

$$\cos\alpha \cos\beta = \frac{1}{2} [\cos(\alpha + \beta) + \cos(\alpha - \beta)] \quad (2.7)$$

У результаті цих дій отримуємо вираз (2.8), що відповідає поставленим вимогам:

$$s_{AM}(t) = U_m \cos\omega_n t + \frac{U_m M}{2} \cos(\omega_n + \Omega)t + \frac{U_m M}{2} \cos(\omega_n - \Omega)t \quad (2.8)$$

Із виразу (2.8) випливає, що спектр АМ-СГ включає три гармоніки з частотами:

$$f_n = \frac{\omega_n}{2\pi}; \quad f_n + F = \frac{\omega_n + \Omega}{2\pi}, \quad f_n - F = \frac{\omega_n - \Omega}{2\pi} \quad (2.9)$$

Таким чином, спектральний склад АМ-СГ зображено на рисунку 2.5.

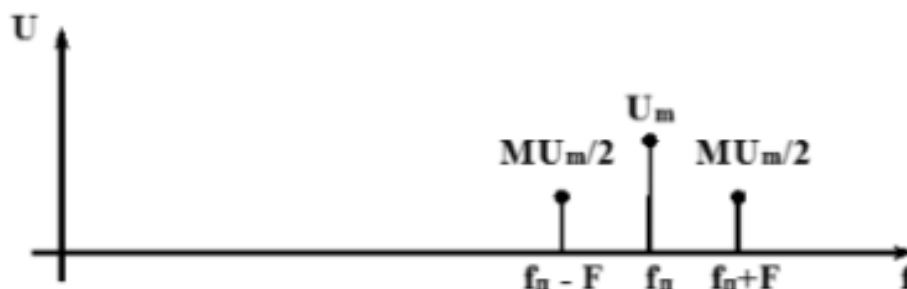


Рис. 2.5. – Спектр АМ-СГ для випадку гармонійного інформаційного СГ

Варто зазначити, що у спектрі відсутня гармоніка інформаційного СГ на частоті F .

2.2. Модель РС в КЗ із АМ під впливом різного роду завад РС в системах зв'язку

Джерела завад можуть бути як зовнішніми, так і внутрішніми стосовно РТС передачі.

Завада (див. рис. 2.6) описується стохастичною функцією часу.

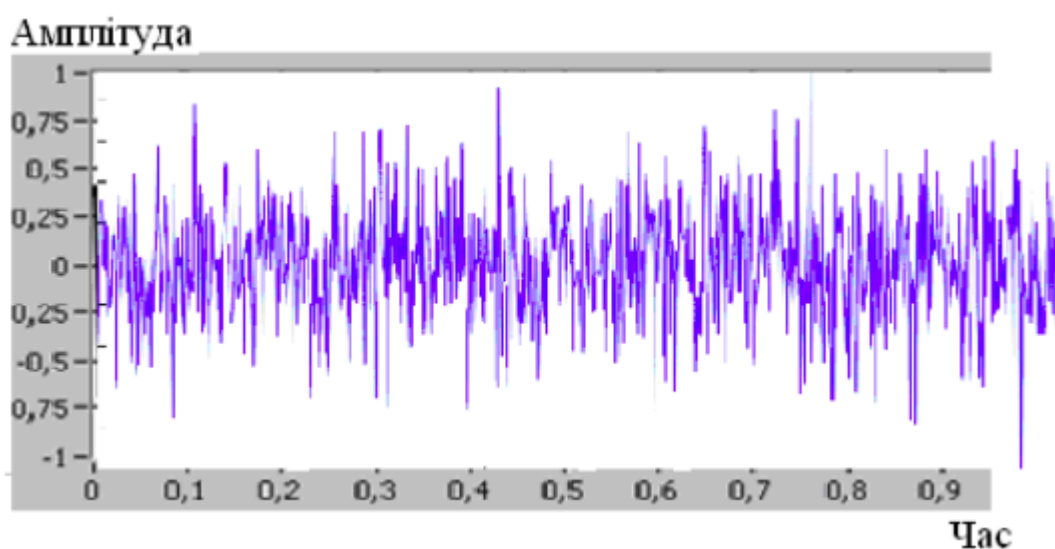


Рис. 2.6. – Завада у КЗ

У загальному вигляді вплив завади $n(t)$ на переданий АМ-РС $s(t)$ можна подати через оператор перетворення прийнятого СГ $\xi(t)$ у формі адитивної моделі:

$$\xi(t) = s(t) + n(t), \quad (2.10)$$

де $n(t)$ – завада в КЗ РТС, яка моделюється у вигляді БГШ – стац. шум, спектральні компоненти якого рівномірно розподілені по всьому частотному діапазону.

У реальній природі та радіотехніці "ідеальний" БШ (із однаковою PSD на всіх частотах) не існує через нескінченну потужність такого СГ. Проте до категорії БШ належать шуми, PSD яких є постійною або майже постійною у визначеному частотному діапазоні.

2.3. Структура РС

У процесі створення РТС необхідно мати модель РС, яка слугуватиме основою для проєктування системи, адже саме на основі вимог до цієї моделі формуються функціональні можливості РТС.

Фінальним етапом розробки є тестування, що забезпечує перевірку коректності роботи алгоритму ПЗ, інтегрованого в ядро системи.

Таким чином, моделювання РС, яке дозволяє забезпечити параметричну ідентифікацію з достовірним відтворенням характеристик, є важливим завданням для перевірки точності функціонування алгоритму ПЗ у складі РТС.

РС – це СГ, що передається РХ. РС функціонує в усіх діапазонах РХ із застосуванням різних видів модуляції (АМ, ЧМ, ФМ тощо). Для трансляції використовуються РП (радіопередавачі), а для прийому — РПМ (радіоприймачі).

Для моделювання розглянемо РС, що містить дані про споживання активної потужності електроенергії за 5 днів. Передача цих даних здійснюється через РК (радіоканал) та обробляється на ПЦ (приймальному центрі), оснащеному спеціалізованою ОТ (обчислювальною технікою).

На рисунку 2.7 представлено експериментальний передавальний РС $\xi(t)$, що складається з набору хвиль, які разом формують період РС. Це призводить до появи сукупності k -тих періодів (добових інтервалів) РС із постійними періодами $T_1 = T_2 = \dots = T_k$.

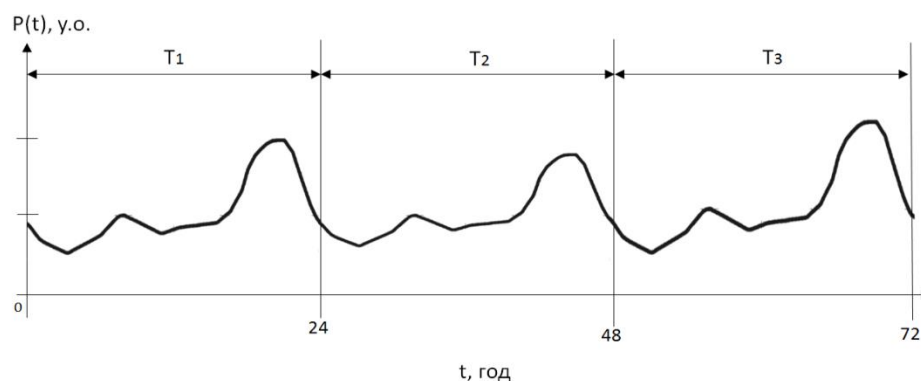


Рис. 2.7. – Графік споживання електроенергії (населення 400 тис. осіб)

З огляду на структуру РС (див. рис. 2.7) було розроблено алгоритм для імітаційного моделювання РС, який дозволяє відтворювати стохастичні та періодичні СГ.

2.4. Алгоритм імітаційного моделювання РС

Згідно з відомими методами моделювання, РС в межах одного періоду моделювали із застосуванням однієї функції, яка покривала весь часовий інтервал.

Однак така модель не відповідає експериментально отриманим даним РС. У цій роботі запропоновано підхід, де РС моделюється за допомогою функцій, визначених на окремих характеристичних часових зонах, схему яких наведено на рисунку 2.8.

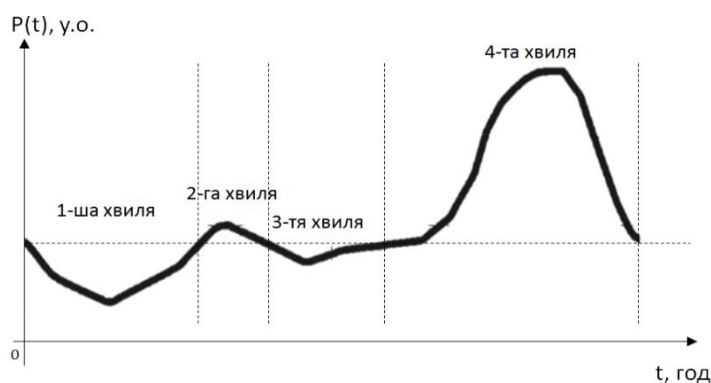


Рис. 2.8. – Розбиття РС на часові зони в межах одного періоду

Оскільки РС у межах періоду складається з n -ої кількості складових, що мають форму характерних кривих, кожену із цих складових моделюємо окремо. Далі всі складові об'єднуються в єдиний масив, який формує загальний РС. Алгоритм моделювання РС для одного періоду представлено на рисунку 2.9.

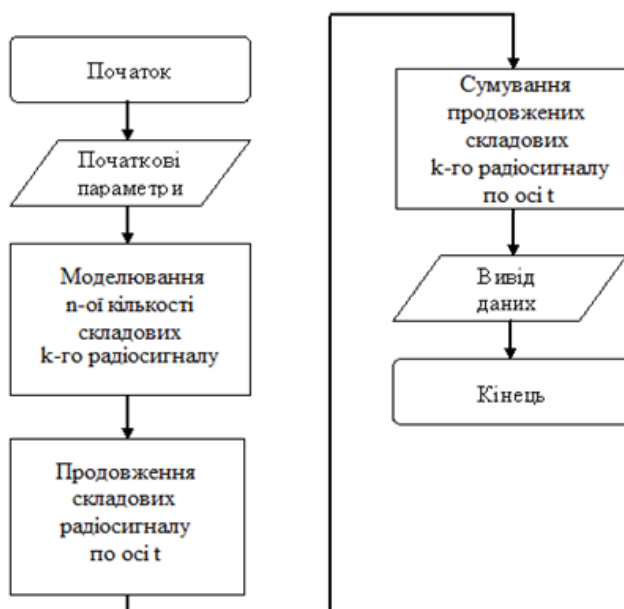


Рис. 2.9. – Алгоритм моделювання РС в межах одного періоду

Згідно з алгоритмом, який наведено на рисунку 2.9, спочатку моделюються n -хвилі РС у межах їхнього часового діапазону для k -го періоду, тобто від 0 до T_{nk} . Потім змодельовані хвилі розміщуються на часовій осі згідно з їх послідовністю, а проміжки часу, які не належать хвилям, заповнюються нулями відповідно до формули 3.1:

$$\xi_k(t) = \sum_{k \in Z} \tilde{\xi}_{nk}(t), \quad t \in [0, T_k), \quad T_1 = T_2 = \dots = T_k \quad (2.11)$$

де $\tilde{\xi}_{nk}(t)$ – продовжені на часовій осі хвилі РС $\xi_{nk}(t)$, $t \in [0, T_{nk})$. Заповнення часових проміжків, яким не належать хвилі, здійснюється за допомогою виразу:

$$\tilde{\xi}_{n\bar{k}}(t) = \begin{cases} \xi_{n\bar{k}}(t), & t \in [T_{(n-1)\bar{k}}, T_{n\bar{k}}) \\ 0, & t \notin [T_{(n-1)\bar{k}}, T_{n\bar{k}}) \end{cases}, \quad (2.12)$$

Тут $T_{n\bar{k}}$ – тривалість n -ої хвилі на k -му періоді РС, причому $T_{1\bar{k}} \neq T_{2\bar{k}} \neq \dots T_{\bar{k}m}$.

Для кращого розуміння формування РС, на рисунку 2.10 продемонстровано принцип його побудови в межах періоду.

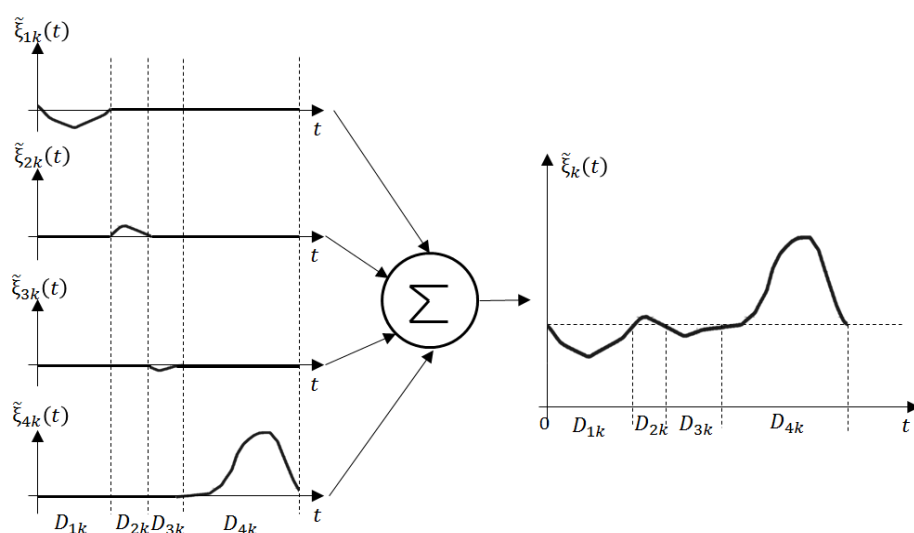


Рис. 2.10. – Принцип формування k -го РС у межах періоду

Згідно з принципом на рисунку 2.10, продовжені хвилі РС сумуються, що дає змогу отримати СГ, сформований у межах одного періоду.

Оскільки РС є періодичним, необхідно моделювати його й на інших інтервалах, тобто продовжити СГ на n -кратну кількість періодів. Окрім періодичності, РС має такі характеристики, як стохастичність амплітуд зубців та їх тривалості, що робить врахування цих факторів важливим завданням. Алгоритм побудови РС (як стохастичного, так і періодичного) представлено на рисунку 2.11.



Рис. 2.11. – Алгоритм формування РС з урахуванням періодичності та стохастичності

Відповідно до наведеного на рисунку 2.11 алгоритму, спочатку моделюється k -та кількість РС у межах періоду T_k (від 0 до T_k). Потім згенеровані РС розміщуються на часовій осі згідно з їх послідовністю, а часові проміжки, де відсутні СГ, заповнюються нулями за формулою:

$$\xi(t) = \sum_{k \in \mathbb{Z}} \tilde{\xi}_k(t), \quad t \in \mathbb{R} \quad (2.12)$$

де $\tilde{\xi}_k(t)$ – періодично продовжений РС на часовій осі $\tilde{\xi}_k(t), t \in \mathbb{R}$:

$$\tilde{\xi}_k(t) = \begin{cases} \xi_k(t), & t \in [T_{k-1}, T_k) \\ 0, & t \notin [T_{k-1}, T_k) \end{cases} \quad (2.13)$$

де $\xi_k(t)$ – РС у межах k -го періоду T_k (T_k), T_k – тривалість періоду, причому $T_1 \neq T_2 \neq \dots \neq T_k$.

На рисунку 2.12 наведено принцип формування періодичного та стохастичного РС для кращого розуміння структури СГ.

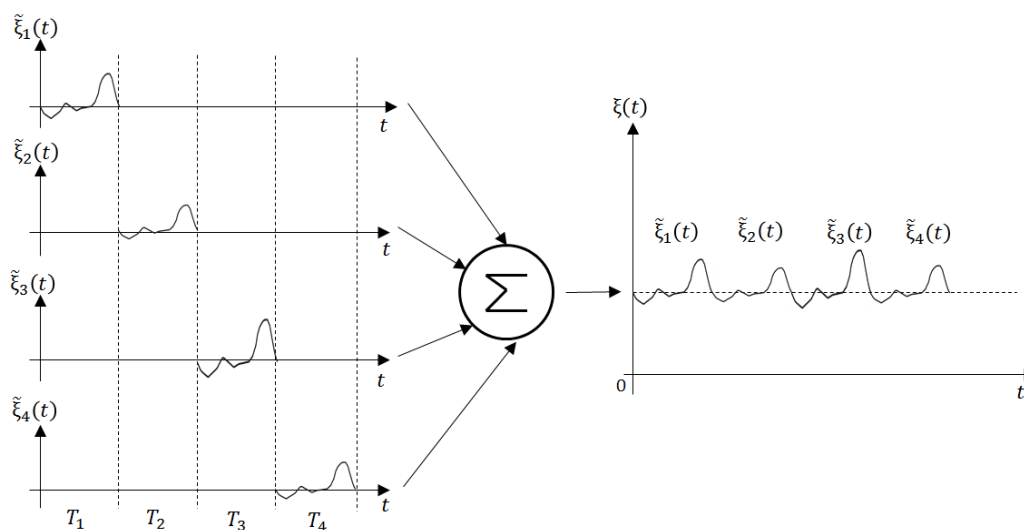


Рис. 2.12. – Принцип формування періодичного та стохастичного РС

У реалізації, представлений на рисунку 2.12, виконано АМ-модуляцію згідно з відомими алгоритмами, які реалізуються за формулою:

$$S_{AM}(t) = \xi(t) \cdot \cos(2 \cdot \pi \cdot f \cdot t + \varphi_0) \quad (2.14)$$

де $\xi(t)$ – амплітудні значення РС (див. рис. 2.12), f – частота НС; φ – ПФ НС.

На рисунку 3.7 наведено принцип побудови СГ в каналах АМ-модуляції.



Рис. 2.13. – Принцип побудови СГ в каналах АМ

2.5. Імітаційна модель РС

Згідно з рисунком 3.6, імітаційна модель описується так:

$$\xi(t) = \sum_{k \in \mathbf{Z}} \tilde{\xi}_k(t) = \sum_{k \in \mathbf{Z}} \sum_{n \in \mathbf{Z}} \tilde{\xi}_{nk}(t), \quad t \in \mathbf{R} \quad (2.15)$$

де $\tilde{\xi}_k(t)$ – продовження k -го періоду РС по числовій осі, $\tilde{\xi}_k(t), t \in \mathbf{R}$; $\tilde{\xi}_{nk}(t)$ – продовжені n -ні хвилі РС по часовій області, $\tilde{\xi}_{nk}(t), t \in [0, T_{nk})$.

Модель n -ї хвилі РС на k -му періоді T_k побудована як синусоїда з експоненційним згасанням, що враховує амплітуди хвиль, часові тривалості та їх стохастичність:

$$s_{nk}(t) = A_{nk} \sin(2 \cdot \pi \cdot t \cdot f_{nk}) \cdot e^{-t \cdot K_{nk}} \cdot L_{nk}, \quad t \in [0, T_{nk}) \quad (2.16)$$

де: n – номер хвилі, $n=1,2,\dots,N$; N – загальна кількість хвиль; T_{nk} – тривалість n -ї хвилі на k -му періоді; A_{nk} – амплітуда n -ї хвилі; f_{nk} – частота СК (півперіоду); K_{nk} – коефіцієнт загасання n -ї хвилі; L_{skj} – масштабні коефіцієнти для n -ї хвилі.

У моделі (див. формулу 2.16) амплітуди хвиль A_{nk} та їх тривалості T_{nk} визначені як сталі значення. Для додання стохастичної складової до виразу введемо такі зміни:

$$\xi_{nk}(t) = (A_{nk} + \psi_A) \sin(2 \cdot \pi \cdot (t + \psi_T) \cdot f_{nk}) \cdot e^{-t \cdot K_{nk}} \cdot L_{nk}, \quad t \in [0, T_{nk}) \quad (2.17)$$

де $\psi_A(M\{A\}, D\{A\})$ – ВВ амплітуди хвиль із $M\{A\}$ та $D\{A\}$ (характеризує відхилення), $\psi_T(M\{T\}, D\{T\})$ – ВВ тривалості хвиль із $M\{T\}$ та $D\{T\}$.

На рисунку 2.14 продемонстровано РС із ключовими інформативними параметрами, необхідними для задачі параметричної ідентифікації.

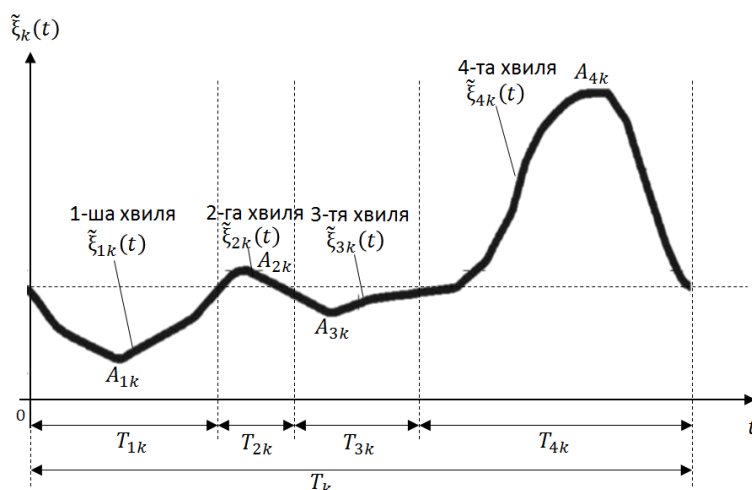


Рис. 2.14. – РС із ключовими інформативними параметрами

Отримані значення функцій (див. формула 2.17) для n -хвиль об'єднують у загальну функцію:

$$\xi_k(t) = \xi_{1k}(t_1) \cup \xi_{2k}(t_2) \cup \xi_{3k}(t_3) \dots \xi_{nk}(t_k), \quad (2.18)$$

де: $t_{1k} \in [0, T_{1k})$ – часовий інтервал 1-ї хвилі, $t_{2k} \in [0, T_{2k})$ – інтервал 2-ї хвилі, $t_{3k} \in [0, T_{3k})$ – інтервал 3-ї хвилі, $t_{nk} \in [0, T_{nk})$ – інтервал n -ї хвилі, $t \in [0, (T_{1k} + T_{2k} + T_{3k} + \dots + T_{nk}))$ – тривалість, рівна періоду T_k РС.

Вираз 2.18 переписується у вигляді:

$$\xi_k(t) = \sum_{n \in Z} \tilde{\xi}_{nk}(t), \quad t \in [0, T_k) \quad (2.19)$$

де $\tilde{\xi}_{nk}(t)$ – продовження n -х хвиль РС $\xi_{nk}(t)$, $\tilde{\xi}_{nk}(t)$, $t \in [0, T_{nk})$.

Цей вираз моделює імітаційну модель РС у межах k -го періоду T_k , яка враховує стохастичність амплітуд і тривалостей хвиль, тобто описує НСП (нестационарний стохастичний процес).

Підставляючи вирази (2.13) і (2.15) у (2.11) та враховуючи адитивний вплив завад згідно з (2.18), отримуємо загальну формулу моделі РС:

$$\xi(t) = \sum_{k=1}^{N_k} \left(\sum_{n=1}^{M_n} \left(\begin{cases} (A_{nk} + \psi_A) \sin(2 \cdot \pi \cdot (t + \psi_T) \cdot f_{nk}) \cdot e^{-t \cdot K_{nk}} \cdot L_{nk}, & t \in [T_{(n-1)k}, T_{nk}) \\ 0, & t \in [T_{nk}, T_{(n+1)k}) \end{cases} \right) + n(t) \right) \quad (2.20)$$

де: N_k – кількість періодів РС, M_n – кількість хвиль у межах k -го періоду.

Згідно із (3.10) алгоритм моделювання наступний:

- Задати масив амплітуд A_{nk} і тривалостей T_{kn} .
- Визначити N_k – кількість періодів, і M_n – хвиль у періодах.
- Встановити M та D для ВВ $\psi_A(M(A), D(A))$ та $\psi_T(M(T), D(T))$.

- Визначити коефіцієнти нахилу хвиль L_{nk} .
- Змодельовати n -ну кількість хвиль згідно з (3.7) для k -го періоду.
- Продовжити n -ні хвилі k -го періоду на всій часовій осі.
- Просумувати продовжені n -ні хвилі k -го періоду.
- Продовжити k -ту кількість періодів на всій часовій осі.
- Просумувати продовжені k -ті періоди РС.
- Провести АМ-модуляцію.
- Врахувати адитивний вплив зовнішньої завади.

2.6. Моделювання РС

Оскільки основними інформативними параметрами в РС є пікові значення хвиль та їх тривалість, як наведено на рисунку 3.7, вони є достатніми для тестування алгоритмів функціонування, враховуючи всі наведені параметри.

Детально розглянемо кожну хвилю. Хвилі СГ описуються як півперіоди синусоїди з плавним зростанням на піковій точці. За допомогою алгебраїчних рівнянь та базових функцій можна сформувати таку криву через добуток синусоїди та експоненційної функції.

Форму експоненційної функції (див. формула 2.21) подано на рисунку 3.9, а вигляд півперіоду синусоїди (див. формула 3.8) наведено на рисунку 3.10.

$$s_{11}(t) = e^{tK_1}, \quad (2.21)$$

де: t – час, K_1 – коефіцієнт, що визначають наростання експоненти.

Експоненційна функція (див. формула 2.21) має вигляд наростаючої кривої, причому нахил залежить від коефіцієнту K .

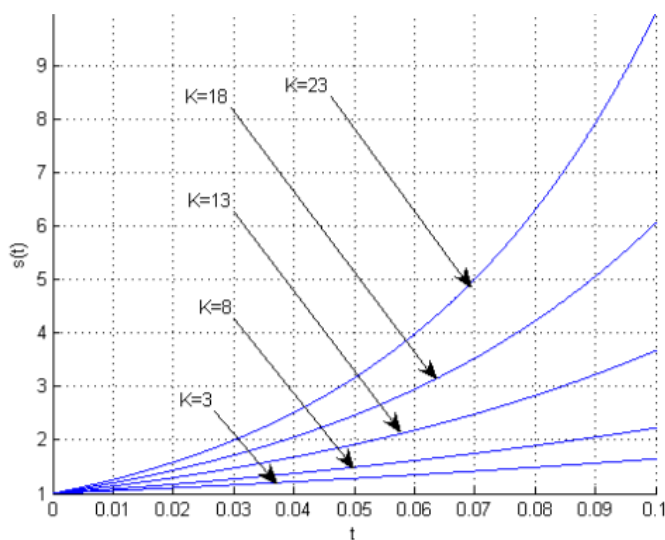


Рис. 2.15. – Реалізація експоненційної функції

Для створення коливного характеру кривої (див. рисунок 2.16) використовується синусоїдальна функція (півперіод) із позитивною або негативною амплітудою. Ця функція задається виразом (див. формула 2.22) і візуалізується на рисунку 3.10:

$$s_{12}(t) = A_2 \sin(2\pi f_2 t), \quad (2.22)$$

де A_2 – амплітуда, f_2 – частота коливань, t – час.

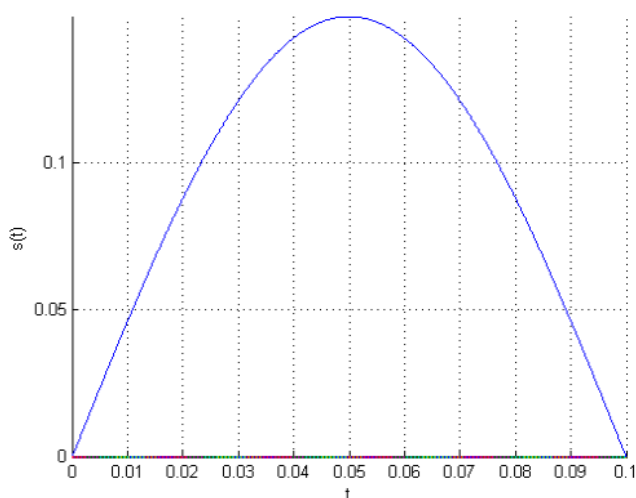


Рис. 2.16. – Реалізація півперіоду синусоїди з позитивною амплітудою

Об'єднання функцій (див. формула 2.21) та (див. формула 2.22) дає вираз (див. формула 2.23), що підтверджує доцільність формули (див. формула 2.16):

$$s(t) = s_{11}(t) \cdot s_{12}(t) = A \sin(2\pi f t) e^{-tK}, \quad (2.23)$$

де A – амплітуда, f – частота, K_2 – коефіцієнт наростання.

Результати моделювання за формулою (2.23) представлено на рисунку 2.17 для різних значень ($K=3, 8, 13, 18, 23$).

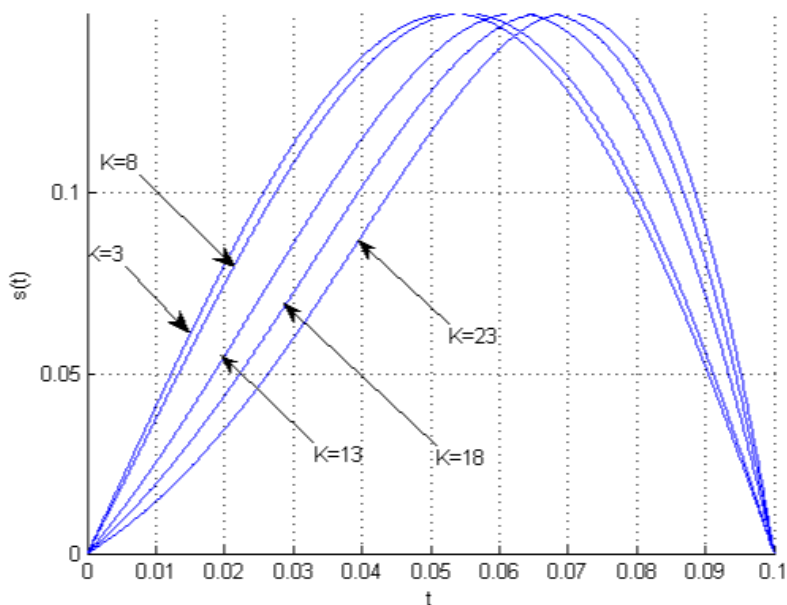


Рис. 2.17. – Реалізація добутку експоненти та синуса

1-ша хвиля є відображенням графіка (див. рисунок 3.12) із негативною амплітудою, тому її матопис відповідає виразу (див. формула 3.13). Формула 1-ї хвилі:

$$s_1(t) = A_1 \sin(2\pi f_1 t) e^{-tK_1}, \quad (2.24)$$

де K_1 – коефіцієнт згасання.

Результати моделювання за формулою (2.24) представлено на рисунку 2.18.

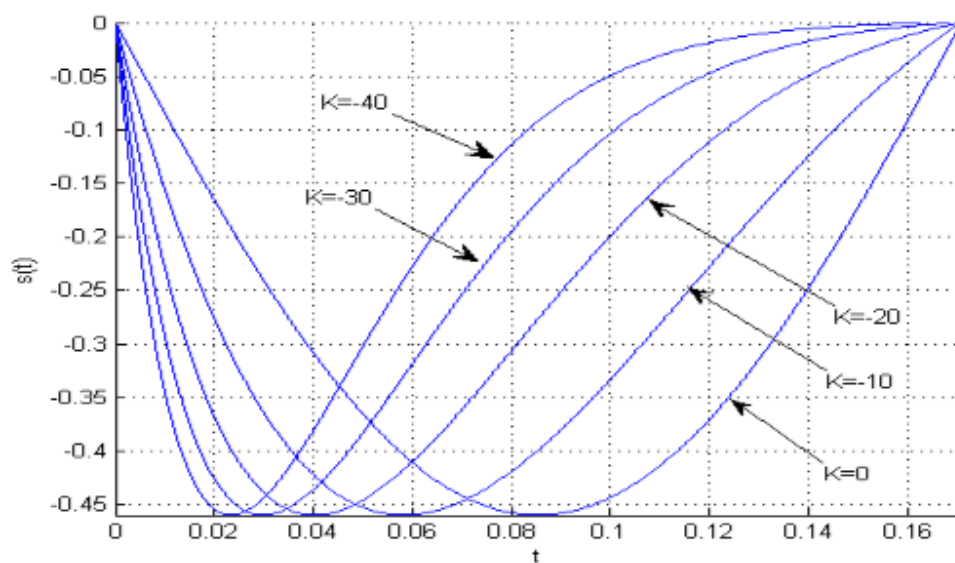


Рис. 2.18. – Моделювана 1-ша хвиля при різних коефіцієнтах

2-га хвиля РС аналогічна 1-й, але має позитивну амплітуду. Формула 2-ї хвилі:

$$s_2(t) = A_2 \sin(2\pi t f_2) e^{-tK_2}, \quad (2.25)$$

Результати наведено на рисунку 2.18.

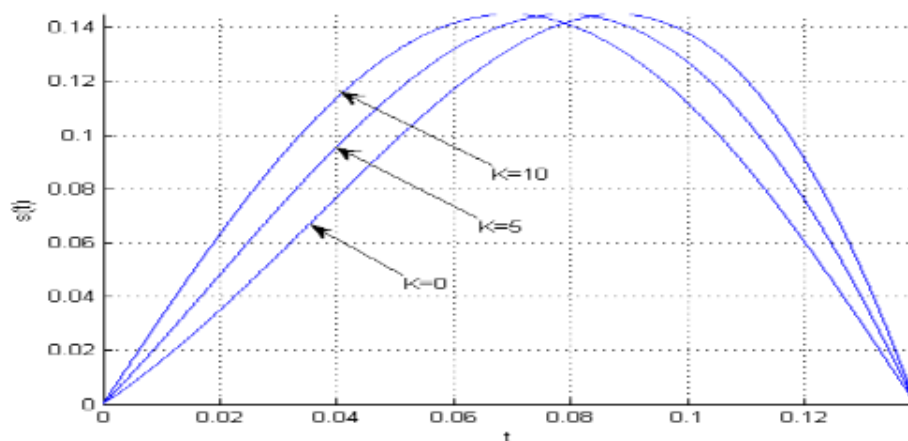


Рис. 2.18 – Моделювання 2-ї хвилі РС

3-тя хвиля подібна 1-й і описується формулою:

$$s_3(t) = A_3 \sin(2\pi f_3 t) e^{-tK_3}, \quad (2.26)$$

Результати подано на рисунку 2.19.

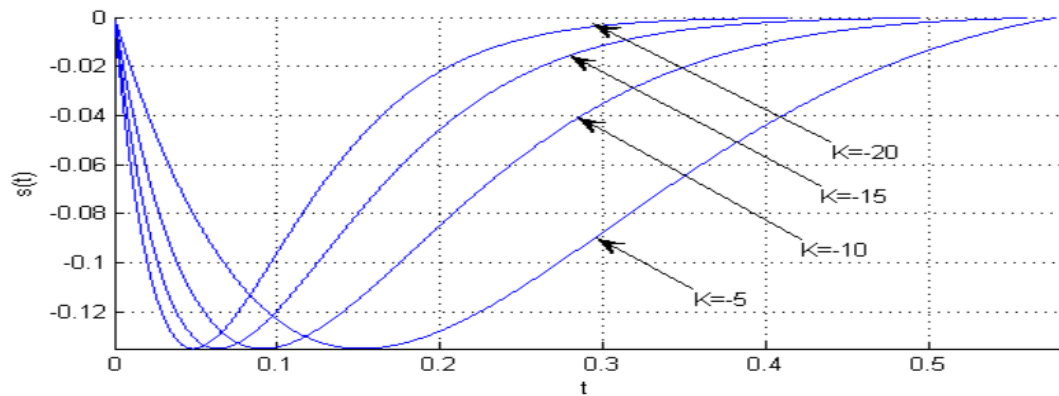


Рис. 2.19. – Модельована 3-я хвиля при різних коефіцієнтах

4-та хвиля аналогічна 2-й, але має експоненціальне затухання. Формула 4-ї хвилі:

$$s_4(t) = A_4 \sin(2\pi f_4 t) e^{-tK_4}, \quad (2.27)$$

Результати представлено на рисунку 2.2.

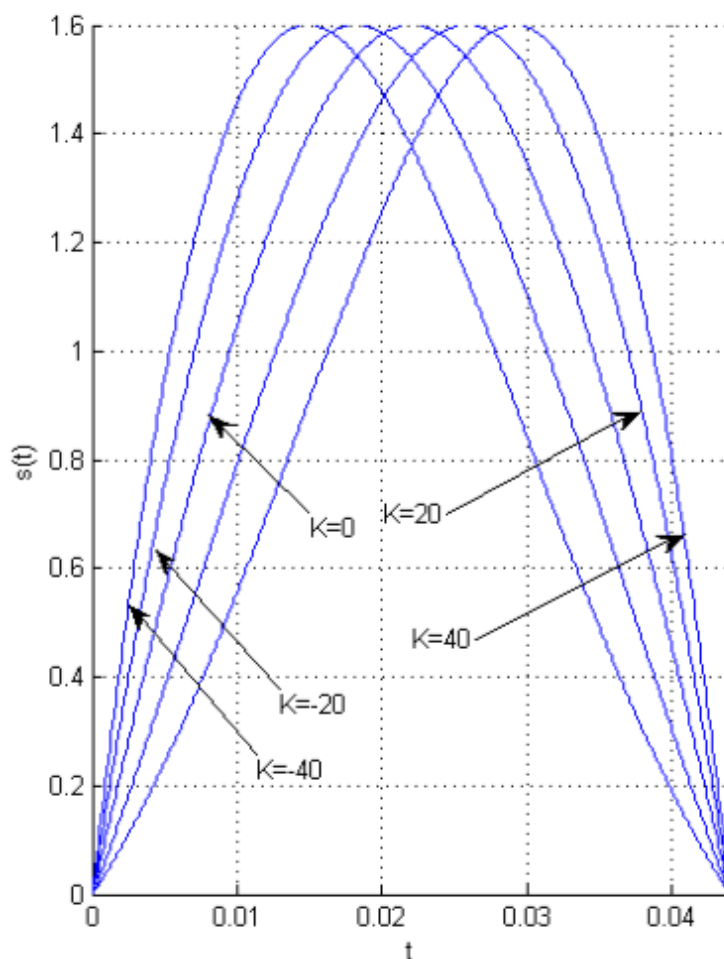


Рис. 2.20. – Реалізація змодельованої 4-ї хвилі РС

Об'єднання функцій 2.18-2.20 в одну, згідно алгоритму імітаційного моделювання (див. формула 2.20), враховує стохастичність амплітуд та тривалостей. Це забезпечує відтворення РС в межах одного періоду T , з підібраними коефіцієнтами K_1, K_2, K_3, K_4 .

З цього випливає, що в межах періоду T РС має поведінку синусоїди з експоненціальними затуханнями. Модель параметричної ідентифікації враховує всі важливі характеристики, такі як амплітуди хвиль та їх часові тривалості.

Оскільки РС є НСП (нестабільним і повторним процесом), моделювання на k -тих періодах дозволяє наблизити його до реального СГ.

2.7. Висновок до другого розділу

В другому розділі кваліфікаційної роботи розроблено матмодель РС як адитивну суміш корисного АМ- СГ та перешкод у вигляді БГШ, що застосовується для тестування алгоритмів функціонування радіотехнічних систем зв'язку.

Модель імітації РС для КЗ в РТС враховує механізм генерації СГ, поєднуючи стохастичність і повторюваність – характерні для реальних СГ. У межах періоду T СГ представлено у вигляді синусоїди з експоненційним загасанням на визначених часових інтервалах. Це дозволяє точно відтворювати РС за заданими параметрами, що необхідно для перевірки алгоритмів роботи РТС.

РОЗДІЛ 3. НАУКОВО-ДОСЛІДНА ЧАСТИНА

Проведення експериментів із застосуванням програмного пакету MATLAB дозволяє детально оцінити процес моделювання РС та перевірити точність і достовірність отриманих результатів.

3.1. Блок-схема програми імітаційного моделювання РС

Для реалізації експериментального моделювання РС необхідно розробити ПЗ, яка виконує імітацію РС. Основою цього процесу є створення послідовності операцій, представлених у вигляді блок-схеми, яка забезпечує логіку роботи ПЗ.

ПЗ складається з двох частин:

- Імітатор РС.
- ПЗ з GUI.

GUI виконує функцію зв'язку між користувачем та імітатором і забезпечує керування параметрами через зручні кнопки у GUI-вікні. Основним елементом є блок-схема генератора РС.

Алгоритм роботи ПЗ:

1. Створення масивів:

- Генерація масиву значень $PX(y_1, y_2, \dots, y_7)$.
- Створення масиву часових відліків $(i_1, i_2, \dots, i_7)$.

2. Об'єднання даних:

Отримані масиви поєднуються в один:

$$[y = [y_1, y_2, y_3, y_4, y_5, y_6, y_7], i = [i_1, i_2, i_3, i_4, i_5, i_6, i_7]]$$

3. Повторення для кількох періодів:

Процес повторюється k -разів, де k – кількість періодів.

4. АМ-модуляція:

Згенерований РС `s` піддається АМ, після чого до нього додається адитивний шум: $s=s+n$.

Ця блок-схема описує загальний алгоритм роботи ПЗ моделювання РС і є основою для створення програмного коду.



Рис. 4.1. – Блок-схема імітатора РС

ПЗ включає GUI, який дозволяє налаштовувати параметри моделювання. Його блок-схему, наведено на рисунку 4.2, забезпечує простий і зручний доступ до функцій програми.

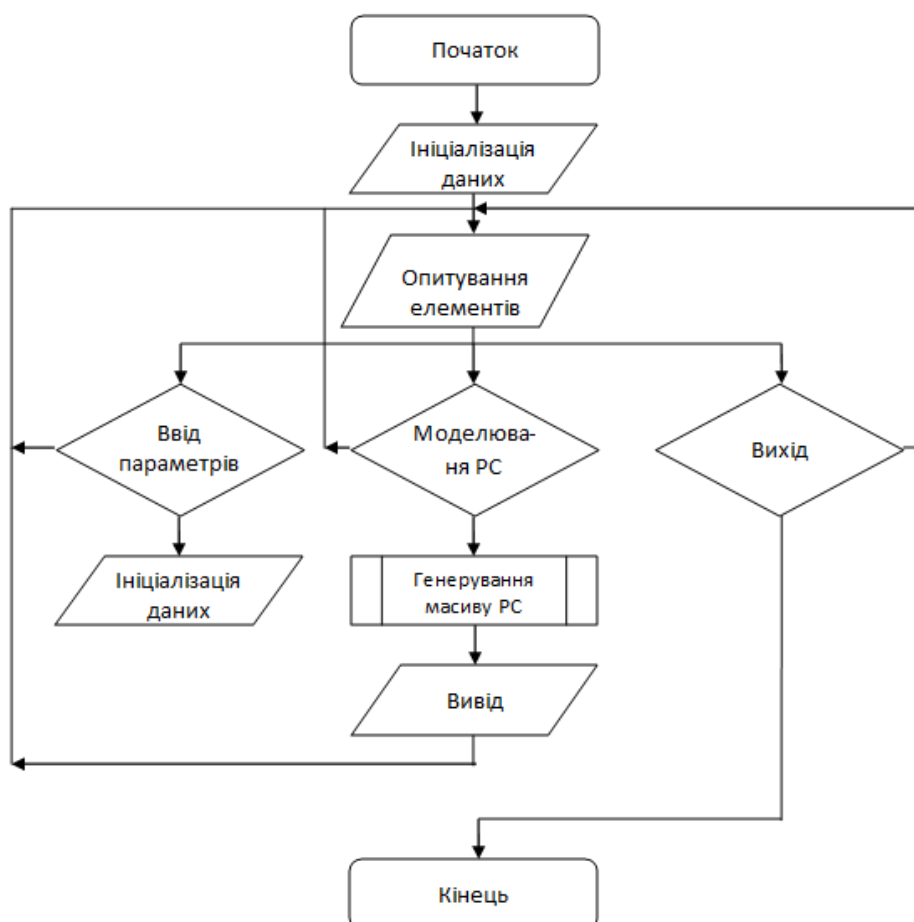


Рис. 4.2. – Блок-схема ПЗ з GUI

На основі цих блок-схем буде розроблено ПЗ імітатора та GUI.

3.2. Розробка ПЗ

Основною задачею цього етапу є програмна реалізація розроблених алгоритмів, яка включає:

- ПЗ імітації РС.
- GUI.

3.2.1. Розробка функції генератора РС

Основою ПЗ є імітатор РС, побудований на базі формул (див. формула 3.4, б), (див. формула 3.10) та блок-схеми (див. рисунок 4.1). Функція написана мовою програмування MATLAB і генерує масив значень РС відповідно до виразу (див. формула 3.10).

Синтаксис функції:

```
function [i, y] = imit(A, t, kk, dt, ddA, ddt, N);
```

Основні вхідні параметри:

- A – масив амплітуд РХ.
- t – масив тривалостей РХ.
- kk – коефіцієнти нахилу.
- dt – крок дискретизації.
- ddA – відхилення амплітуди.
- ddt – відхилення тривалостей.
- N – кількість періодів.

Перевірка вхідних параметрів. Функція 'nargin' перевіряє кількість введених аргументів. Якщо їх менше семи, виводиться повідомлення про помилку:

```
if nargin < 7
    error('Необхідно ввести не менше 7 аргументів');
end
```

Генерація амплітудних відхилень. Задаються стохастичні відхилення амплітуди РХ:

```
for k = 1:4
    rA(k) = normrnd(0, ddA);
    A(k) = A(k) + rA(k);
end
```

Якщо амплітуда першої РХ менша за нуль, змінюємо її на додатну:

```
if A(1) < 0
    A(1) = -A(1);
end
```

Генерацію відхилень тривалостей наведено у лістингу 4.1. Стохастичні відхилення тривалостей РХ розраховуються за умови, що $\text{ddt} > 0$:

Лістинг 4.1 – Генерація відхилень тривалостей

```
if ddt > 0
    for k = 1:4
        rt(k) = normrnd(0, ddt);
        t(k) = t(k) + rt(k);
        tN(k) = round(t(k) / dt);
        t(k) = tN(k) * dt;
    end
end
```

У лістингу 4.2 наведено програмний код для формування РС. Для кожної РХ генеруємо масив часу i , частоту коливань f та значення функції:

Лістинг 4.2 – Формування РС

```
i1 = 0:dt:t(1);
f1 = 2 * t(1);
y1 = sin(2 * pi * i1 / f1) .* exp(i1 * kk(1));
K = A(1) / max(y1);
y1 = y1 .* K;
```

Аналогічно обчислюються параметри для наступних РХ (див. лістинг 4.3):

Лістинг 4.3 – Обчислення параметрів наступних РХ

```
i2 = 0:dt:t(2);
f2 = 2 * t(2);
y2 = sin(2 * pi * i2 / f2) .* exp(i2 * kk(2));
K = A(2) / max(y2);
y2 = y2 .* K;
i2 = i2 + t(1);
```

Процедура повторюється для РХ y_3 , y_4 , змінюючи відповідно часові інтервали.

У лістингу 4.4 наведено об'єднання масивів. Масиви часу та значень функцій об'єднуються.

Лістинг 4.4 – Об'єднання масивів

```
y(1:length(y1)) = y1;
l1 = length(y); l2 = length(y) + length(y2) - 1; y(l1:l2) = y2;
l1 = length(y); l2 = length(y) + length(y3) - 1; y(l1:l2) = y3;
l1 = length(y); l2 = length(y) + length(y4) - 1; y(l1:l2) = y4;
i = (0:(length(y) - 1)) * dt;
```

Генерація кількох періодів. Для формування послідовності періодів РС операції виконуються у циклі:

```
for u = 1:N
    % моделювання u-го періоду
end
```

Отримані масиви `y` та `i` формують повний СГ. Масив у об'єднується із загальним масивом `x`. Таким чином, отримуємо повноцінну послідовність РС.

3.2.2. Розробка GUI для імітаційного моделювання РС

Для створення GUI використано утиліту GUIDE Quick Start, інтегровану в середовище MATLAB.

Створення діалогового вікна. Процес починається з введення команди `guide` у командному рядку MATLAB. Після цього відкривається стартове вікно (див. рисунок 4.3).

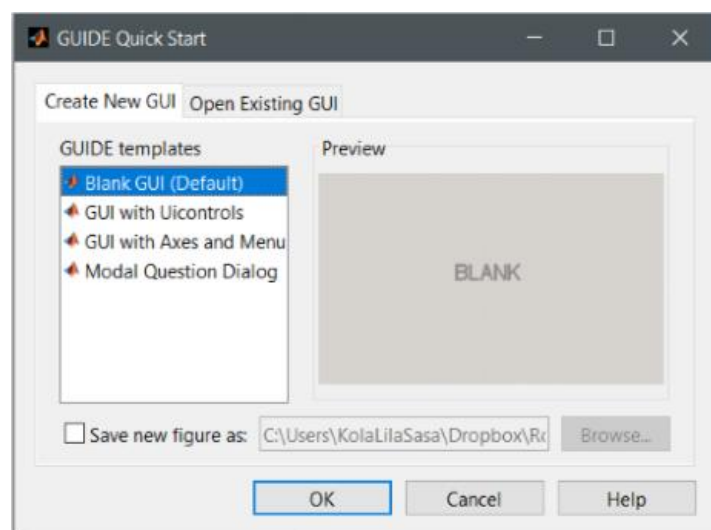


Рис. 4.3. – Стартове вікно GUIDE

Обравши опцію `Blank GUI (Default)` і натиснувши `OK`, відкриваємо робоче діалогове вікно (див рисунок 4.4), у якому розташовуються основні елементи керування.

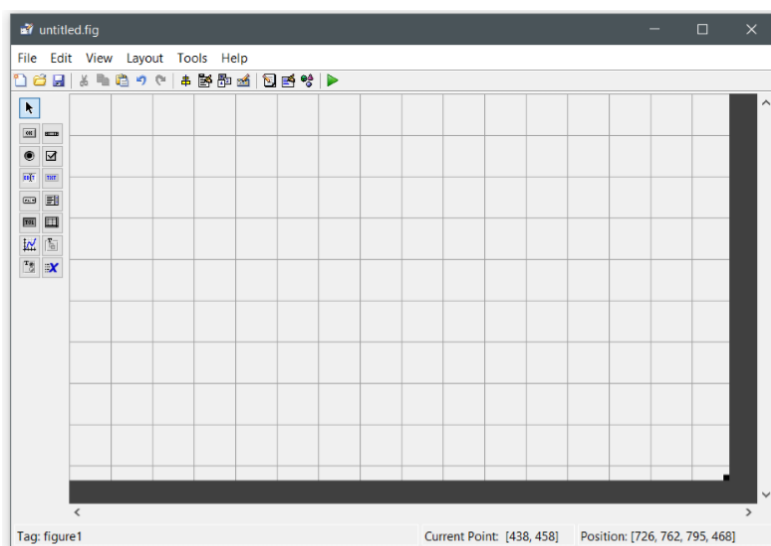


Рис. 4.4. – Робоче діалогове вікно

Діалогова форма. Використовуючи панель графічних об'єктів, було розроблено діалогову форму, яку наведено на рисунку 4.5.

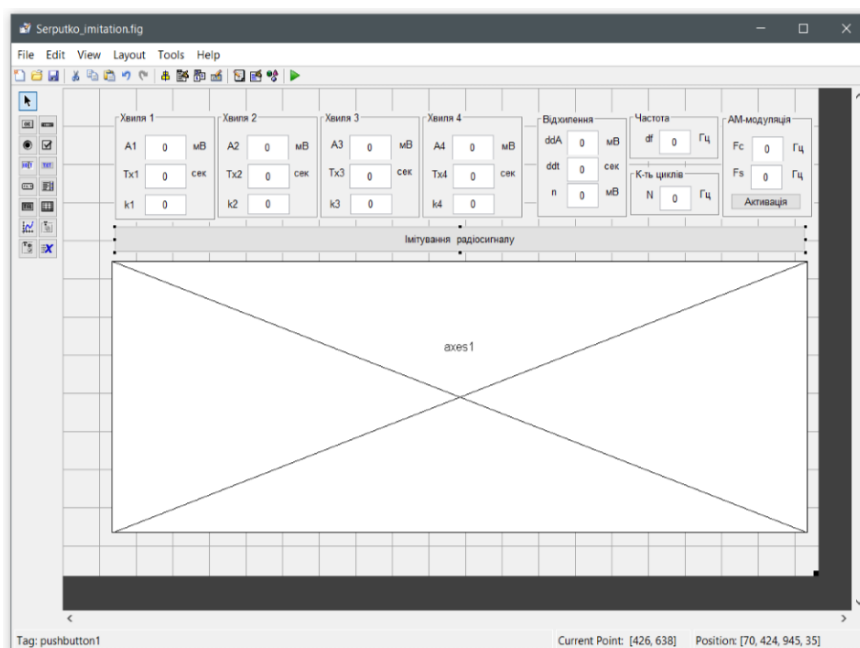


Рис. 4.5. – Діалогова форма з GUI

Для створення форми використано такі елементи:

- Edit Text – Поля для введення даних.
- Frame – Рамки для розділення областей.
- Text – Текстові підписи.
- Pushbutton – Кнопки дій.
- Axes – Графічне поле для виведення графіків.

Логіка роботи елементів. Кожен елемент має відповідну функцію. Наприклад, при натисканні на кнопку `Pushbutton1` викликається функція:

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

Програма запускає функцію `erg\_model\_OpeningFcn`, яка відповідає за ініціалізацію. Команда `set(a, b, c)` задає значення об'єктів `edit1-edit19`, де:

- `a` – Об'єкт.
- `b` – Параметр об'єкта.
- `c` – Значення параметра.

Для прикладу, об'єкт `handles.edit` із параметром `string` містить текстові значення, які відображаються на екрані.

Побудова графіка. Для вибору області виводу графіка використовується команда:

```
axes(handles.axes1);
```

Дані для побудови графіка зчитуються командами, які наведено у лістингу 4.5.

Лістинг 4.5 – Дані для побудови графіка

```
A(1) = str2num(get(handles.edit1, 'string'));
t(1) = str2num(get(handles.edit2, 'string'));
ddA = str2num(get(handles.edit13, 'string'));
ddt = str2num(get(handles.edit14, 'string'));
N = str2num(get(handles.edit16, 'string'));
Dn = str2num(get(handles.edit19, 'string'));
[x, t] = imit(A, t, k, dt, ddA, ddt, N);
x = x + normrnd(0, Dn, length(x), 1)';
```

Після розрахунків графік будується командою `plot`:

```
plot(t, x);
axis tight;
grid on;
```

Дані зберігаються у масиві `data` для подальшого використання:

```
data(1, :) = t;
data(2, :) = x;
set(handles.pushbutton1, 'userdata', data);
```

Збереження та завантаження даних. Для збереження даних використовується меню, створене через Menu Editor, яке наведено на рисунку 4.6.

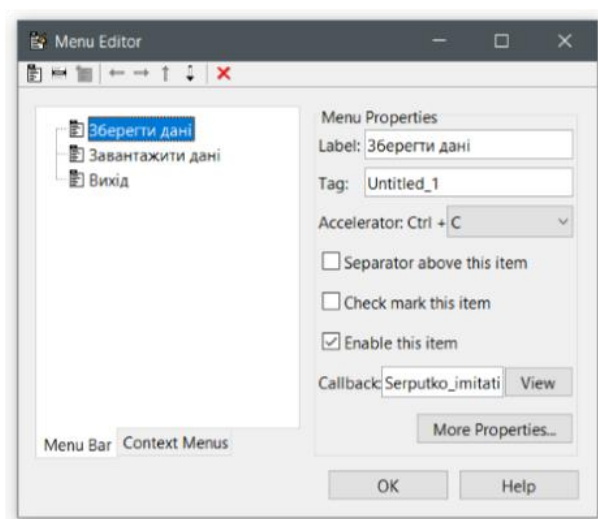


Рис. 4.6. – Розроблене меню

Основні функції меню:

- Зберегти дані.
- Завантажити дані.
- Вихід.

Діалогове вікно для завантаження файлу (див. рисунок 4.7) викликається функцією:

```
[filename, pathname] = uigetfile('*.dat', 'Завантажити дані моделі');
```

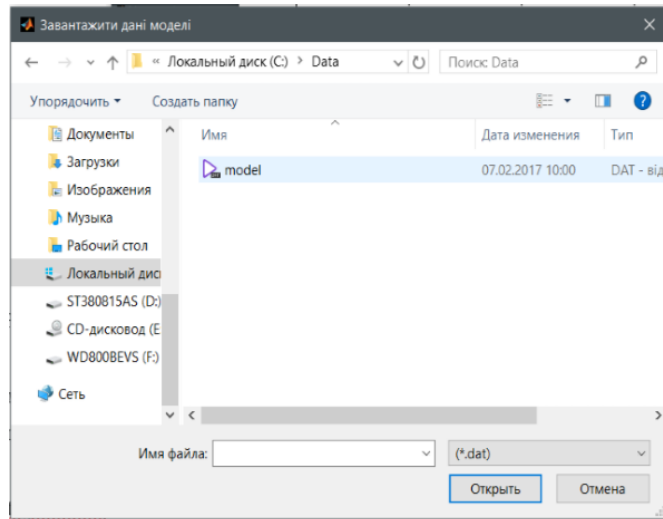


Рис. 4.7. – Діалогове вікно для завантаження файлу

Дані зчитуються через команду:

```
M = dlmread([pathname, filename]);
t = M(1, :); x = M(2, :);
plot(t, x);
```

Для збереження даних викликається вікно (див. рисунок 4.8):

```
[filename, pathname] = uiputfile('*.dat', 'Зберегти дані моделі');
dlmwrite([pathname, filename], data, ' ');
```

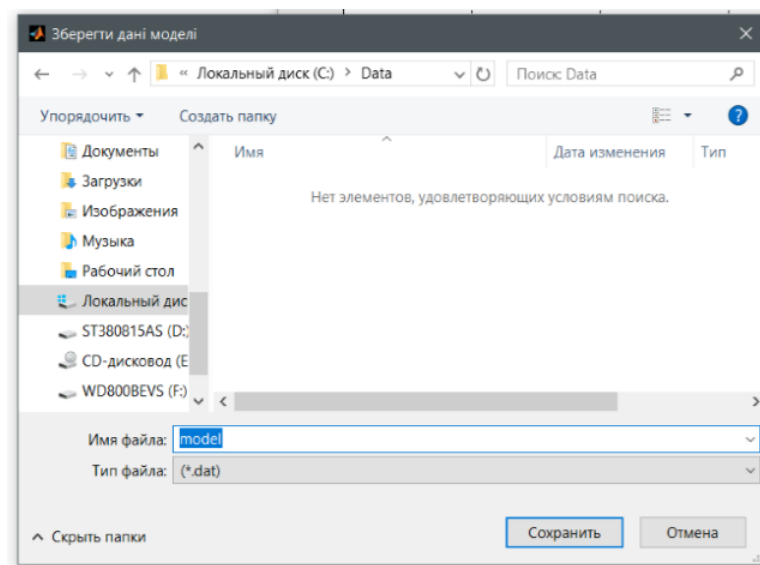


Рис. 4.8. – Вікно для збереження даних

Команда `dlmwrite` зберігає масив даних `data` у зазначений файл.

3.3. Результати імітаційного моделювання РС

Завдяки утиліті GUIDE Quick Start, яку інтегровано в MATLAB, створено програмний модуль з GUI, який зображено на рисунку 4.9. Цей модуль імітує РС на основі формули 3.10.

Спершу було згенеровано РС в межах одного циклу без впливу завад.

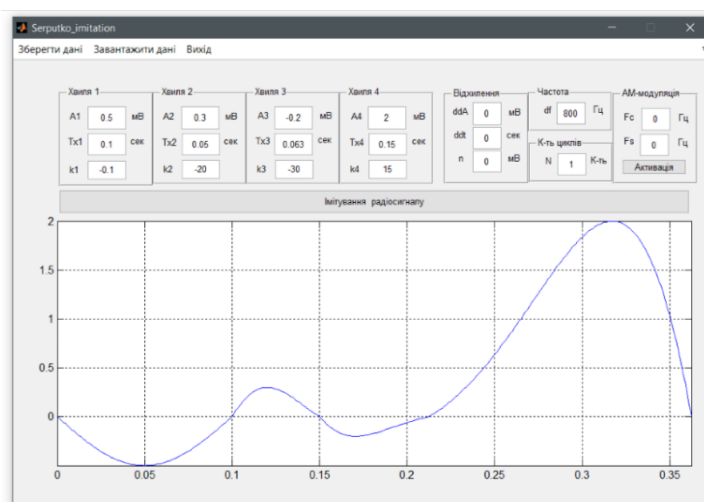


Рис. 4.9. – GUI для моделювання РС для одного циклу

Результати моделювання при різних кількості циклів. На рисунках 4.10-4.12 зображено кілька реалізацій РС для різних значень кількості циклів.

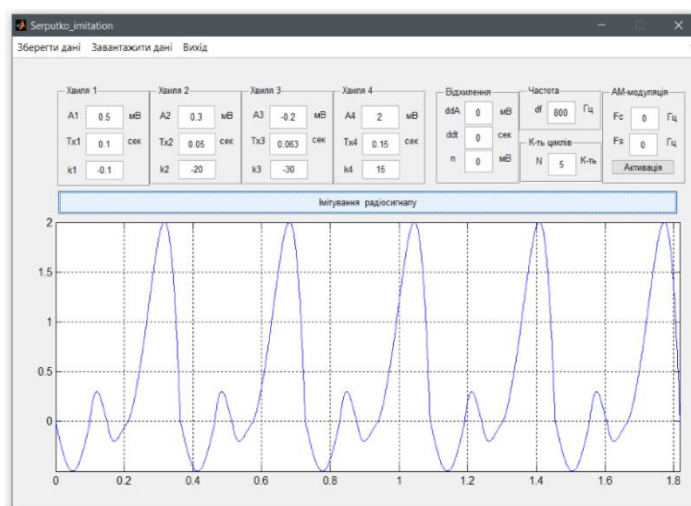


Рис. 4.10. – Симульований РС при N=5

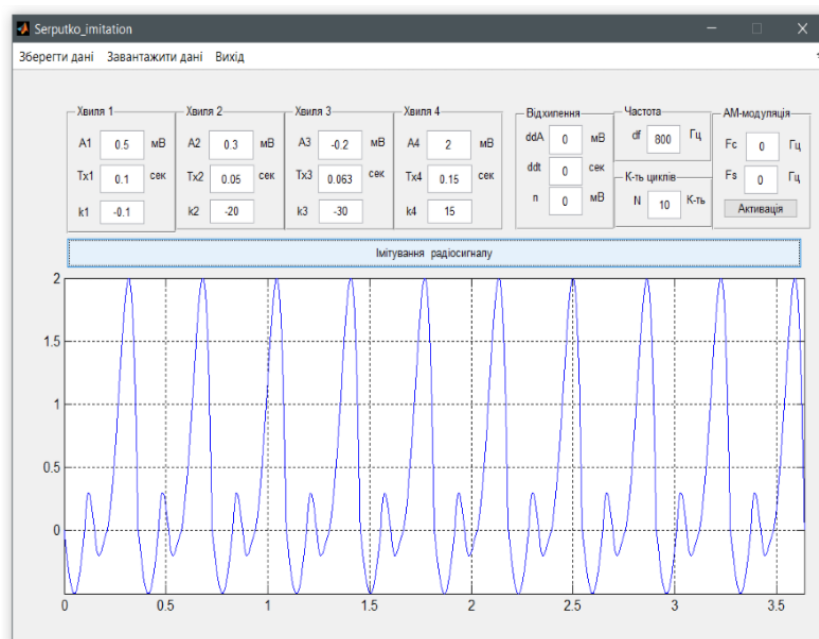


Рис. 4.11. – Симульований РС при N=10

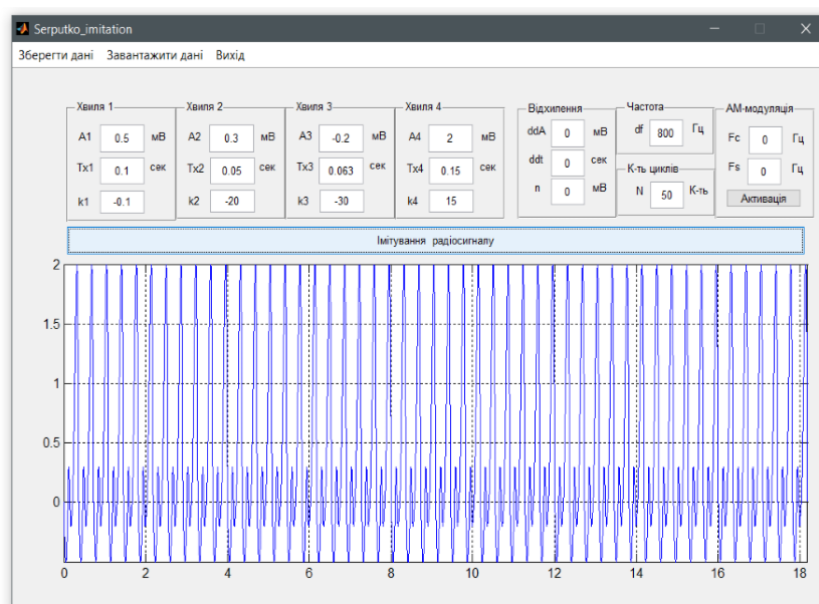


Рис. 4.12. – Симульований РС при N=50

Моделювання РС із стохастичними параметрами. На рисунках 4.13-4.14 наведено результати симуляції РС із стохастичними параметрами моделі.

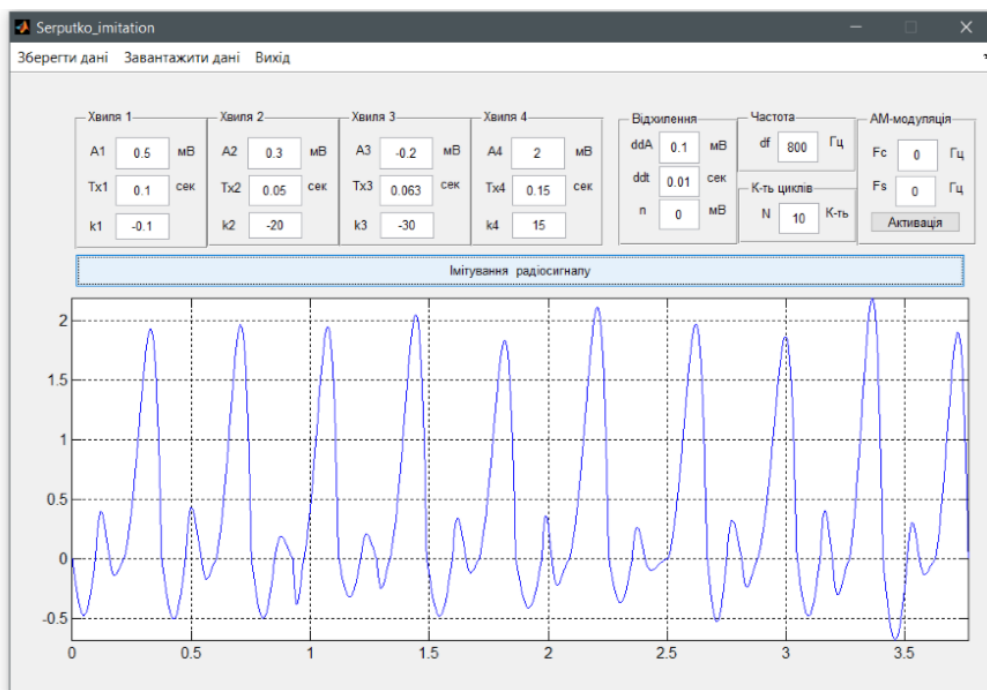


Рис. 4.13. – Симульований РС із стохастичними параметрами при N=10

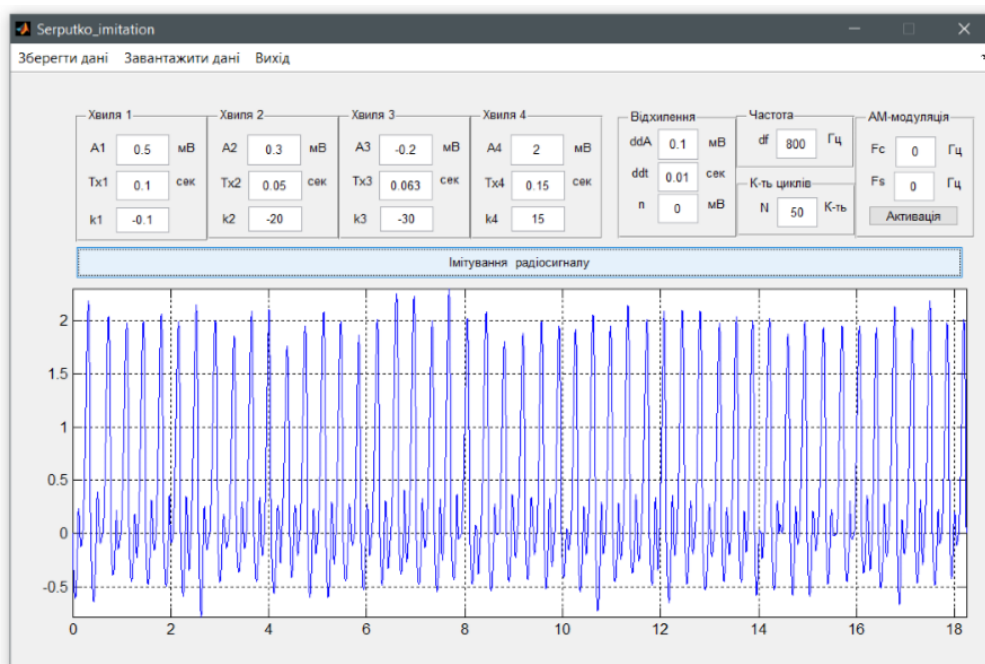


Рис. 4.14. – Симульований РС із стохастичними параметрами при N=50

Моделювання РС з адитивним впливом завад. Результати симуляції РС із стохастичними параметрами моделі та адитивними завадами різної інтенсивності наведено на рисунках 4.15-4.16.

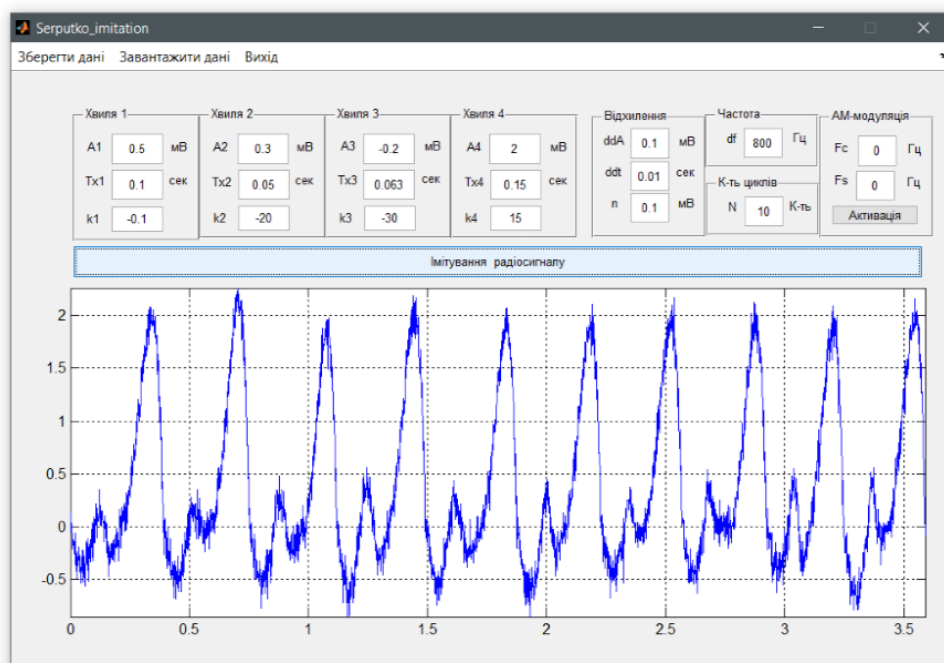


Рис. 4.15. – Симульований РС при $N=10$ та дисперсії шуму $0,1\text{мВ}^2$

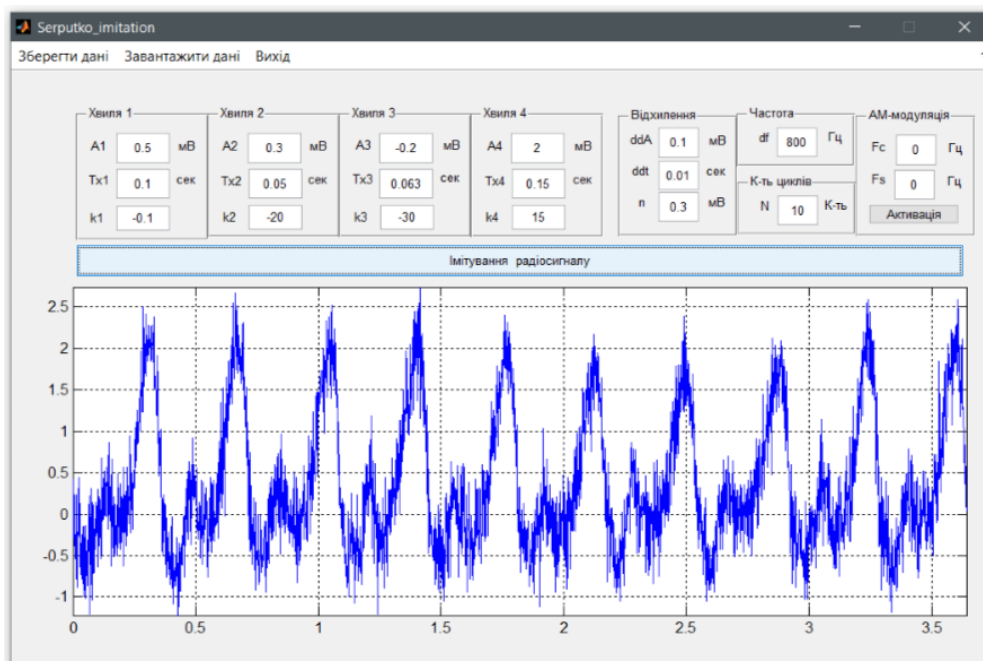


Рис. 4.16. – Симульований РС при $N=10$ та дисперсії шуму $0,3\text{мВ}^2$

Симуляція АМ РС. Результати моделювання АМ-РС із стохастичними параметрами моделі та адитивними завадами подано на рисунках 4.17-4.20.

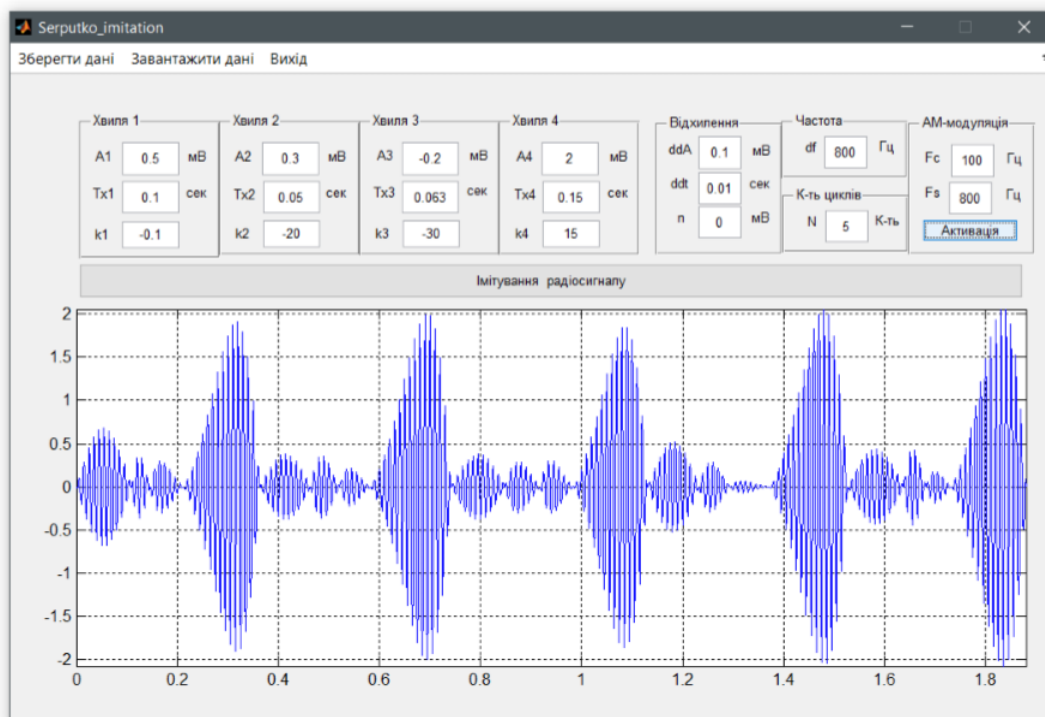


Рис. 4.17. – АМ-імітований РС при N=5 без завад

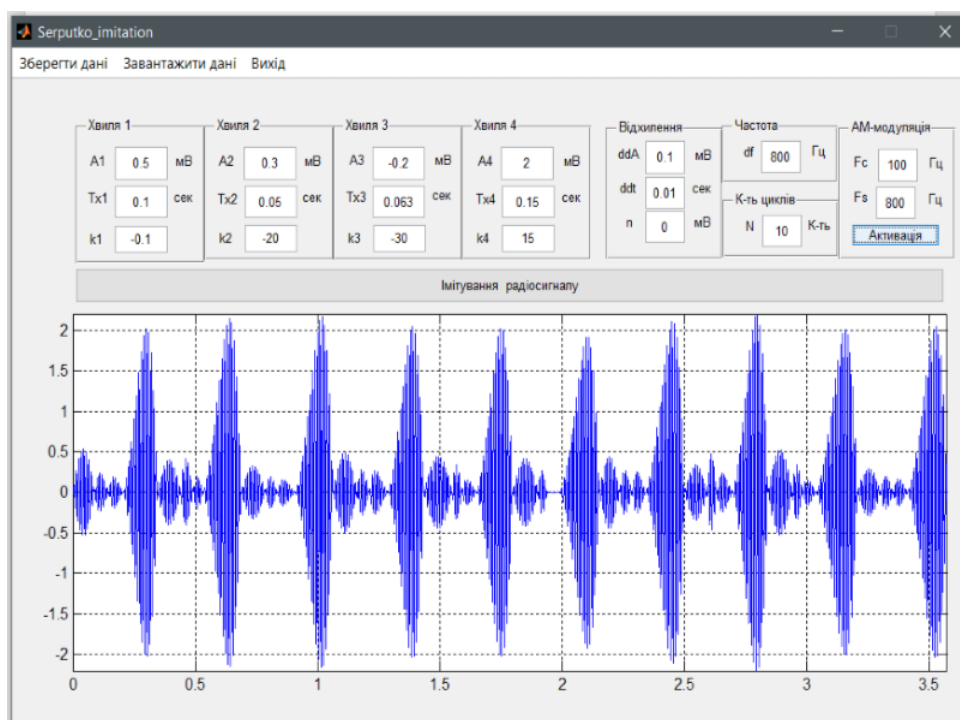


Рис. 4.18. – АМ-імітований РС при N=10 без завад

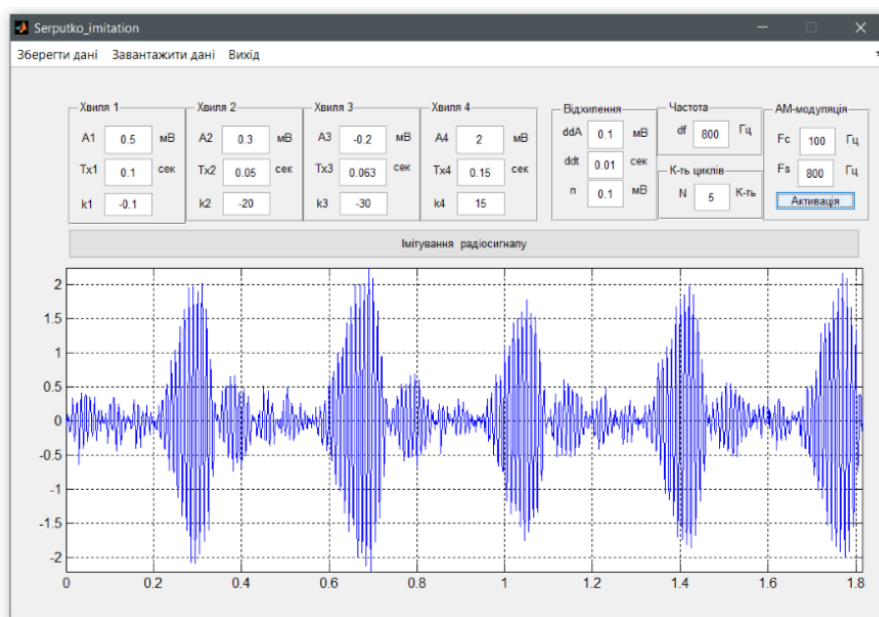


Рис. 4.19. – АМ-імітований РС при $N=5$, дисперсія шуму $0,1 \text{ мВ}^2$

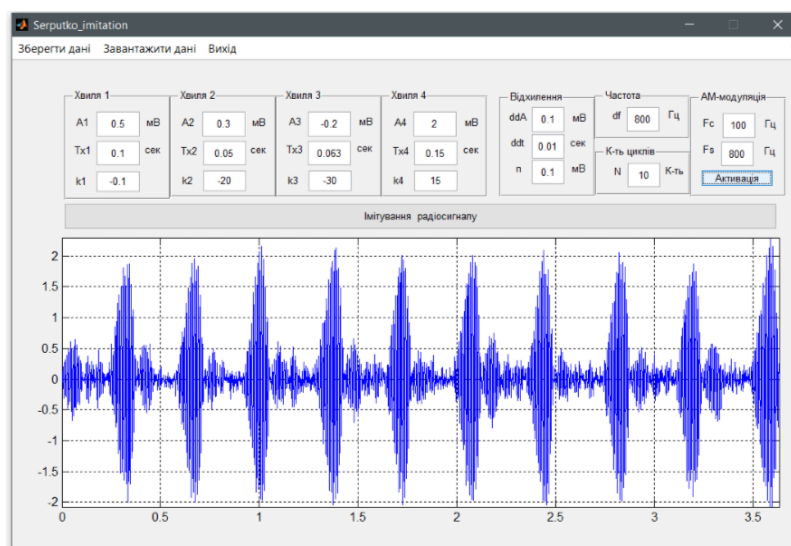


Рис. 4.20. – АМ-імітований РС при $N=10$, дисперсія шуму $0,1 \text{ мВ}^2$

3.4. Висновок до третього розділу

Розроблена модель симуляції РС враховує механізм генерації, поєднуючи стохастичність із повторюваністю – характерну для експериментальних СГ у КЗ. У середовищі MATLAB створено ПЗ з GUI, яка дозволяє легко генерувати будь-який тип РС.

РОЗДІЛ 4.

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

При плануванні та організації робочого місця дослідника, який виконує науково-дослідну роботу, згідно санітарних норм, існує ряд правил, які необхідно виконувати для забезпечення нормальної працездатності упродовж усього робочого дня. Монітори комп'ютерів орієнтовані так, щоб кількість бліків була мінімальна, а відстань до очей оператора оптимальна. На рис. 4.1 зображено схему організації розміщення робочого місця користувача ПК.

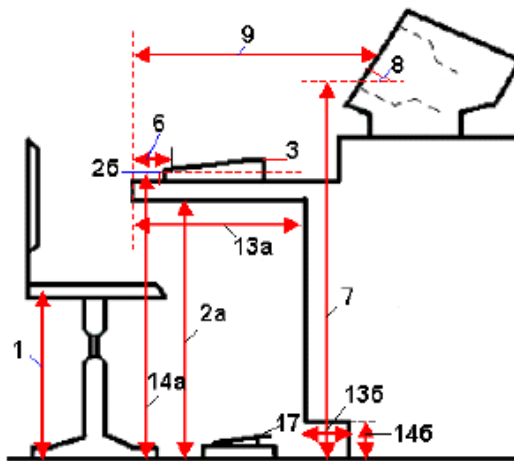


Рис. 4.1. – Основні просторові параметри робочого місця користувача ПК

На рис. 4.2 показані джерела електромагнітного випромінювання ПК.

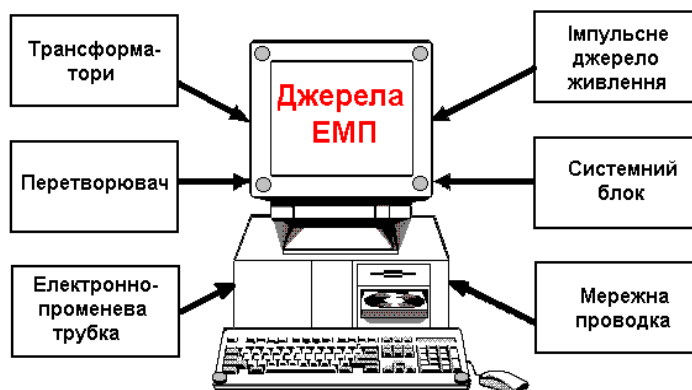


Рис. 4.2 – Джерела електромагнітного випромінювання ПК

Площу приміщень, в яких розташовують відеотермінали, визначають згідно з чинними нормативними документами з розрахунку на одне робоче місце, обладнане відеотерміналом: площа - не менше $6,0 \text{ м}^2$, обсяг - не менше $20,0 \text{ м}^3$, з урахуванням максимальної кількості осіб, які одночасно працюють у зміні.

Нормування параметрів середовища проводять з урахуванням характеру трудової діяльності, особливостей механізму впливу на організм робітників чинників зовнішнього середовища та ступеня важкості і напруженості праці. Зокрема, під час виконання робіт, які передбачають розробку концепцій нових програм, рівень шуму не повинен перевищувати 40 дБА, під час виконання операторської діяльності - 65 дБА. Рівень шуму під час роботи ЕОМ у дисплейних залах не перевищує 50 дБА, а рівні звукового тиску в октавних полосах частот з середньгеометричними значеннями 63, 125, 250, 5000, 1000, 2000, 4000 та 8000 Гц - відповідно повинні складати не більш, ніж 71, 61, 54, 49, 45, 42, 40 та 38 дБ [61].

Приміщення, що оснащені дисплеями (ділянки підготовки даних, дисплейні зали), мають північну або північно-східну орієнтацію. Коли у діючих обчислювальних центрах такі приміщення орієнтовані переважно на південні, південно-західні або південно-східні румби, передбачають обов'язкове розташування на вікнах сонцезахисних пристроїв (жалюзі, штори). В приміщеннях, які оснащені відеотерміналами, стіни пофарбовані у кольори пастельних тонів з коефіцієнтом відбиття, що знаходиться у межах 40-60%. Забарвленню поверхонь надано матову фактуру.

Робочі місця, які обладнані ПК, розташовують таким чином, щоб у поле зору користувача не потрапляли вікна та освітлювальні прилади. Тому відеотермінали розміщено під кутом $90-105^\circ$ до площини зовнішньої стіни. В поле зору користувача ПК не повинні потрапляти і інші поверхні робочого приміщення, які мають властивості віддзеркалювання. Співвідношення яскравості екрана та поверхонь, які оточують, повинно складати 5:1, причому,

безпосереднє покриття робочого стола повинно мати коефіцієнт відбиття 0,25-0,4.

Параметри мікроклімату в дисплейних залах забезпечують комфортне теплове самопочуття організму користувача ПК. Тому оптимальна температура повітря в цих приміщеннях становить 19-21 °С, (допустима – 18-22 °С), оптимальна відносна вологість – 40-60%, оптимальна швидкість руху повітря - 0,05-0,1 м/с.

З метою запобігання електротравм забороняється працювати з моніторами, які є незаземленими, а також з моніторами, у яких порушений зовнішній вигляд або під час роботи з'являються нехарактерні сигнали, нестабільне зображення на екрані.

4.2. Безпека в надзвичайних ситуаціях

Створення оптимальних комфортних умов у виробничих приміщеннях по виготовленню телекомунікаційних систем та окремих його деталей є складною задачею, вирішити яку можна наступними заходами та засобами:

- Удосконалення технологічних процесів та устаткування.
- Впровадження нових технологій та обладнання, які не пов'язані з необхідністю проведення робіт в умовах інтенсивного нагріву дасть можливість зменшити виділення тепла у виробничі приміщення. Наприклад, заміна гарячого способу обробки металу — холодним, нагрів полум'ям – індуктивним, горнових печей – тунельними.
- Рациональне розміщення технологічного устаткування. Основні джерела теплоти бажано розміщувати безпосередньо під аераційним ліхтарем, біля зовнішніх стін будівлі і в один ряд на такій відстані один від одного, щоб теплові потоки від них не перехрещувались на робочих місцях. Для охолодження гарячих виробів необхідно передбачити окремі приміщення. Найкращим рішенням є розміщення тепловипромінюючого обладнання в

ізолюваних приміщеннях або на відкритих ділянках.

- Автоматизація та дистанційне управління технологічними процесами. Цей захід дозволяє в багатьох випадках вивести людину із виробничих зон, де діють несприятливі фактори (наприклад автоматизоване завантаження печей в металургії, управління розливом сталі).

- Раціональна вентиляція, опалення та кондиціювання повітря. Вони є найбільш розповсюдженими способами нормалізації мікроклімату у виробничих приміщеннях. Так зване повітряне та водоповітряне душення широко використовується у боротьбі з перегріванням робітників в гарячих цехах.

Забезпечити нормальні теплові умови в холодний період року в надтогабаритних та полегшених промислових будівлях дуже важко і економічно недоцільно. Найбільш раціональним варіантом в цьому випадку є застосування променистого нагрівання постійних робочих місць та окремих ділянок. Захист від протягів досягається шляхом щільного закривання вікон, дверей та інших отворів, а також влаштуванням повітряних і повітряно-теплових завіс на дверях і воротах.

Раціоналізація режимів праці та відпочинку досягається скороченням тривалості робочої зміни, введенням додаткових перерв, створенням умов для ефективного відпочинку в приміщеннях з нормальними метеорологічними умовами. Якщо організувати окреме приміщення важко, то в гарячих цехах створюють зони відпочинку – охолоджувальні альтанки, де засобами вентиляції забезпечують нормальні температурні умови.

Для робітників, що працюють на відкритому повітрі зимою, обладнують приміщення для зігрівання, в яких температуру підтримують дещо вищою за комфортну. Застосування теплоізоляції устаткування та захисних екранів. В якості теплоізоляційних матеріалів широко використовуються: азбест, азбоцемент, мінеральна вата, склотканина, керамзит, пінопласт.

На виробництві застосовують також захисні екрани для відгородження джерел теплового випромінювання від робочих місць. За принципом захисту

щодо дії тепла екрани бувають відбиваючі, поглинаючі, відвідні та комбіновані. Хороший захист від теплового випромінювання здійснюють водяні завіси, що широко використовуються в металургії.

Використання засобів індивідуального захисту. Важливе значення для профілактики перегрівання мають індивідуальні засоби захисту. Спецодяг повинен бути повітро- та вологопроникним (бавовняним, з льону, грубововняного сукна), мати зручний покрій. Для роботи в екстремальних умовах застосовуються спеціальні костюми з підвищеною теплосвітловіддачею. Для захисту голови від випромінювання застосовують дюралеві, фіброві каски, повстані капелюхи; для захисту очей – окуляри – темні або з прозорим шаром металу, маски з відкидним екраном. Захист від дії зниженої температури досягається використанням теплового спецодягу, а під час опадів – плащів та гумових чобіт.

4.3. Висновок до четвертого розділу

У підрозділі з охорони праці проаналізовано питання вимог безпеки до лабораторних приміщень та обладнання для наукових досліджень, шкідливих і небезпечних факторів під час роботи з персональним комп'ютером при виконанні науково-дослідної роботи та забезпечення нормальних умов праці на робочому місці.

У підрозділі з безпеки в надзвичайних ситуаціях проаналізовано оптимальні комфортні умови у виробничих приміщеннях по виготовленню телекомунікаційної системи.

ЗАГАЛЬНІ ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну наукову задачу – розроблено імітаційну модель РС для моделювання та верифікації методів їх обробки.

Результати роботи:

1. Проведено аналіз існуючих моделей РС, на основі якого сформульовано основні вимоги:

- Врахування стохастичності та циклічності РС.
- Можливість ймовірнісного тестування алгоритмів РТС.

2. Розроблено нову модель РС, що перевершує відомі аналоги, завдяки:

- Включенню стохастичності та циклічності.
- Здатності точно відтворювати експериментальні РС за відомими параметрами.

- Забезпеченню параметричної ідентифікації.

3. Спроектовано ПЗ із GUI для моделювання РС, яке:

- Дозволяє тестувати алгоритми РТС.
- Забезпечує зручний і функціональний інструментарій для роботи.

4. Доведено, що створена модель:

- Повністю відтворює форму експериментальних РС за часовими та амплітудними параметрами.

- Гарантує достовірність моделювання та адекватність тестування алгоритмів РТС.

ПЕРЕЛІК ПОСИЛАНЬ

1. Бабак В. П. Детерміновані сигнали і спектри: навч. посіб./ В. П. Бабак, А. Я. Білецький. – К.: Техніка, 2003. – 454 с.
2. Бабак В. П. Обробка сигналів: підручник./ В. П. Бабак, В. С. Хандецький, Е. Шрюфер. – К: Либідь, 1999. – 250 с.
3. Жураковський Ю. П. Теорія сигналу та кодування [Текст] : підручник / Ю.П. Жураковський, В.П. Полторак. – К.: Вища шк., 2001. – 255 с.
4. Уривський Л. О. Основи прикладної теорії інформації для телекомунікацій [Текст] : дисертація доктора технічних наук : 05.12.02 / Л. О. Уривський; Київ, НТУУ «КПІ». - К. : 2009. - 377 с.
5. Couch L.W. Signal-to-noise ratio out of ideal FM detector for SSB FM plus Gaussian noise at the input//IEEE Trans. (Concise Papers). — 1969. — Vol. COM-17, № 1. P. 591-592.
6. Dahlman E. 3G Evolution: HSPA and LTE for Mobile Broadband [Text] / E. Dahlman. – Academic Press is an imprint of Elsevier, 2007. – 485 p.
7. Proakis J.G. Digital Communications, 4th ed., New York, McGraw-Hill, 2001.
8. Slyusar V.I., Smolyar V.G. The method of nonorthogonal frequency-discrete modulation of signals for narrow-band communication channels // Radio Electronics and Communications Systems C/C of Izvestiia – Vysshie Uchebnye Zavedeniia Radioelektronika. – 2004, VOL 47; PART 4, P. 40–41. – Allerton Press Inc. (USA).
9. Дунець В.Л., Хвостівська Л.В., Паляниця Ю.Б. Математичне, алгоритмічне та програмне забезпечення оцінювання завадозахищеності каналів зв'язку з балансною модуляцією. Збірник наукових праць Вісник НУ ВГП, серія технічні науки, випуск 4 (104), 2023. - С. 95-107. ISSN: 2306-5478.
10. Liliya Khvostivska, Mykola Khvostivskyy, Vasyl Dunets, Iryna Dediv. “Mathematical and Algorithmic Support of Detection Useful Radiosignals in Telecommunication Networks”. Proceedings of the 2nd International Workshop on

Information Technologies: Theoretical and Applied Problems, 22-24 November 2022, Ternopil, Ukraine, ITTAP 2022, 2022, pp. 314-318. ISSN 1613-0073

11. Дунець В.Л., Хвостівський М.О., Сверстюк А.С., Хвостівська Л.В. Математичне та алгоритмічно-програмне забезпечення опрацювання електрокардіосигналів при фізичному навантаженні у кардіодіагностичних системах: наукова монографія. Львів: Видавництво «Магнолія - 2006», 2022. 136 с. ISBN 978-617-574-242-6.

12. Безпека життєдіяльності (забезпечення соціальної, техногенної та природної безпеки: Навч. посібник/ В.В. Бєгун, І.М. Науменко. – К.: Видавництво УАННП “Фенікс”, 2004. – 328с.

13. Білявський Г. О. Основи екології [Текст] : підручник для студ. вищих навч. закладів / Г. О. Білявський, Р. С. Фурдуй, І. Ю. Костіков. – К. : Либідь, 2004. - 408 с. – ISBN 966-06-0289-8.

14. Жидецький В. Ц. Основи охорони праці [Текст] : підруч. / В. Ц. Жидецький ; М-во освіти і науки України. Наук.-метод. центр вищої освіти. Укр. акад. друкарства ; рец.: Г. Г. Гогіташвілі, І. І. Даценко, Б. С. Штангрет. — 3-тє вид., перероб. і доп. — Львів : Укр. акад. друкарства, 2006. — 336 с. — Бібліогр.: с. 329–330. — ISBN 966-8013-11-5

15. Йохна М.А., Стадник В.В. Економіка і організація інноваційної діяльності: Навч. посіб. – К.: Видавничий центр «Академія», 2005. – 400с.

ДОДАТКИ

Додаток А

Програма генератора радіосигналу

```

function [x t]=imit(AAA,ttt,kk,dt,ddA,ddt,N);
%
% Генератор сигналу
% imit(A,t,kk,dt,ddA,ddt);
% A - масив амплітуд хвиль
% t - масив тривалостей відповідних хвиль
% kk - кути нахилу хвиль
% dt - крок дискретизації
% ddA - дисперсія для амплітуди
% ddt - дисперсія для часових інтервалів
% Кількість періодів

if nargin<7, error('Необхідно ввести не менше 7-ми
аргументів'); end;

x=[];
for u=1:N
    t=ttt; A=AAA;
% Значень відхилень для амплітуди
for k=1:4
    rA(k)=normrnd(0,ddA);
    A(k)=A(k)+rA(k);
end;
if A(1)>0 A(1)=-A(1); end;

if ddt>0
% Значень відхилень для часових інтервалів
T=sum(t);
rt(1)=normrnd(0,ddt); t(1)=t(1)+rt(1);
tN(1)=round(t(1)/dt); t(1)=tN(1)*dt;
rt(2)=normrnd(0,ddt); t(2)=t(2)+rt(2);
tN(2)=round(t(2)/dt); t(2)=tN(2)*dt;
rt(3)=normrnd(0,ddt); t(3)=t(3)+rt(3);
tN(3)=round(t(3)/dt); t(3)=tN(3)*dt;
rt(4)=normrnd(0,ddt); t(4)=t(4)+rt(4);
tN(4)=round(t(4)/dt); t(4)=tN(4)*dt;
end;

% 1-ша функція
i1=0:dt:t(1); f1=2*t(1); y1=sin(2*pi*i1/f1).*exp(i1*kk(1));
K=A(1)/max(y1); y1=y1.*K;

% 2-га функція

```

```

i2=0:dt:t(2); f2=2*t(2); y2=sin(2*pi*i2/f2).*exp(i2*kk(2));
K=A(2)/max(y2); y2=y2.*K; i2=i2+t(1);

% 3-тя функція
i3=0:dt:t(3); f3=2*t(3); y3=sin(2*pi*i3/f3).*exp(i3*kk(3));
K=A(3)/max(y3); y3=y3.*K; i3=i3+t(1)+t(2);

% 4-та функція
i4=0:dt:t(4); f4=2*t(4); y4=sin(2*pi*i4/f4).*exp(i4*kk(4));
K=A(4)/max(y4); y4=y4.*K; i4=i4+t(1)+t(2)+t(3);

y(1:length(y1))=y1;
l1=length(y); l2=length(y)+length(y2)-1; y(l1:l2)=y2;
l1=length(y); l2=length(y)+length(y3)-1; y(l1:l2)=y3;
l1=length(y); l2=length(y)+length(y4)-1; y(l1:l2)=y4;
i=(0:(length(y)-1))*dt;

x=[x y];
y1=0; y2=0; y3=0; y4=0; y=0;
end;
t=0:length(x)-1; t=t.*dt;

```

Програма імітування радіосигналів із графічним інтерфейсом

```

function varargout = Serputko_imitation(varargin)
% SERPUTKO_IMITATION M-file for Serputko_imitation.fig
%
%           SERPUTKO_IMITATION, by itself, creates a new
SERPUTKO_IMITATION or raises the existing
%       singleton*.
%
%           H = SERPUTKO_IMITATION returns the handle to a new
SERPUTKO_IMITATION or the handle to
%       the existing singleton*.
%
%
SERPUTKO_IMITATION('CALLBACK', hObject,eventData,handles,...) calls
the local
%       function named CALLBACK in SERPUTKO_IMITATION.M with the
given input arguments.
%
%           SERPUTKO_IMITATION('Property','Value',...) creates a new
SERPUTKO_IMITATION or raises the
%       existing singleton*. Starting from the left, property
value pairs are
%       applied to the GUI before intarface_EKS_OpeningFunction
gets called. An
%       unrecognized property name or invalid value makes property
application
%       stop. All inputs are passed to
Serputko_imitation_OpeningFcn via varargin.
%
%       *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%       instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help
Serputko_imitation

% Last Modified by GUIDE v2.5 07-Feb-2017 09:40:59

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Serputko_imitation_OpeningFcn, ...

```

```

        'gui_OutputFcn',
@Serputko_imitation_OutputFcn, ...
        'gui_LayoutFcn',    [] , ...
        'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Serputko_imitation is made visible.
function Serputko_imitation_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Serputko_imitation (see
VARARGIN)

% Choose default command line output for Serputko_imitation
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Serputko_imitation wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% Встановлення початкових параметрів

N=0; x=0;
set(handles.edit1,'string','0.5');
set(handles.edit2,'string','0.1');
set(handles.edit3,'string','0.3');
set(handles.edit4,'string','0.05');
set(handles.edit5,'string','-0.2');
set(handles.edit6,'string','0.063');
set(handles.edit7,'string','2');
set(handles.edit8,'string','0.15');

set(handles.edit9,'string','-0.1');
set(handles.edit10,'string','-20');
set(handles.edit11,'string','-30');
set(handles.edit12,'string','15');

```

```

set(handles.edit13,'string','0');
set(handles.edit14,'string','0');

set(handles.edit15,'string','800');

set(handles.edit16,'string','1');

axes(handles.axes1);

A(1)=str2num(get(handles.edit1,'string'));
A(2)=str2num(get(handles.edit3,'string'));
A(3)=str2num(get(handles.edit5,'string'));
A(4)=str2num(get(handles.edit7,'string'));

t(1)=str2num(get(handles.edit2,'string'));
t(2)=str2num(get(handles.edit4,'string'));
t(3)=str2num(get(handles.edit6,'string'));
t(4)=str2num(get(handles.edit8,'string'));

k(1)=str2num(get(handles.edit9,'string'));
k(2)=str2num(get(handles.edit10,'string'));
k(3)=str2num(get(handles.edit11,'string'));
k(4)=str2num(get(handles.edit12,'string'));

ddA=str2num(get(handles.edit13,'string'));
ddt=str2num(get(handles.edit14,'string'));

dt=1/str2num(get(handles.edit15,'string'));

N=str2num(get(handles.edit16,'string'));

Dn=str2num(get(handles.edit19,'string'));

%k=[0 8 20 -15 -40 0 -10];
[x t]=imit(A,t,k,dt,ddA,ddt,N);
x=x+normrnd(0,Dn,length(x),1)';
data(1,:)=t;
data(2,:)=x;
set(handles.pushbutton1,'userdata',data);
plot(t,x);
axis tight;
grid on;

% --- Outputs from this function are returned to the command line.
function varargout = Serputko_imitation_OutputFcn(hObject,
eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

```

```
varargout{1} = handles.output;
```

```
function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of
edit1 as a double
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of
edit2 as a double
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```

if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%          str2double(get(hObject,'String')) returns contents of
edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%          str2double(get(hObject,'String')) returns contents of
edit4 as a double

% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.

```

```

if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%          str2double(get(hObject,'String')) returns contents of
edit5 as a double

% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%          str2double(get(hObject,'String')) returns contents of
edit6 as a double

% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.

```

```

if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%          str2double(get(hObject,'String')) returns contents of
edit7 as a double

% --- Executes during object creation, after setting all
properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%          str2double(get(hObject,'String')) returns contents of
edit8 as a double

% --- Executes during object creation, after setting all
properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.

```

```

if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%          str2double(get(hObject,'String')) returns contents of
edit9 as a double

% --- Executes during object creation, after setting all
properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%          str2double(get(hObject,'String')) returns contents of
edit10 as a double

% --- Executes during object creation, after setting all
properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%         str2double(get(hObject,'String')) returns contents of
edit11 as a double

% --- Executes during object creation, after setting all
properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%         str2double(get(hObject,'String')) returns contents of
edit12 as a double

% --- Executes during object creation, after setting all
properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)
% hObject      handle to edit13 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%          str2double(get(hObject,'String')) returns contents of
edit13 as a double

% --- Executes during object creation, after setting all
properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit13 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)
% hObject      handle to edit14 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
%          str2double(get(hObject,'String')) returns contents of
edit14 as a double

% --- Executes during object creation, after setting all
properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit14 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%         str2double(get(hObject,'String')) returns contents of
edit13 as a double

% --- Executes during object creation, after setting all
properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename, pathname] = uiputfile('.dat', 'Зберегти дані моделі');
if length(filename)>=1
dir=[pathname filename]
data=get(handles.pushbutton1,'userdata');
dlmwrite(dir,data,' ');
end;

% -----
----
function Untitled_2_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

axes(handles.axes1);
[filename, pathname] = uigetfile('*.dat', 'Завантажити дані моделі');
if length(filename)>=1
director=[pathname filename]
M=DLMREAD(director);
len=length(M);
t=0; x=0;
t=M(1,:);
x=M(2,:)
%t=M(1:len/2); x=M(len/2+1:len);
data=M;
plot(t,x);
axis tight;
grid on;
set(handles.pushbutton1,'userdata',data);
end;

% -----
----
function Untitled_3_Callback(hObject, eventdata, handles)
% hObject      handle to Untitled_3 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Натиснута кнопка

A(1)=str2num(get(handles.edit1,'string'));
A(2)=str2num(get(handles.edit3,'string'));
A(3)=str2num(get(handles.edit5,'string'));
A(4)=str2num(get(handles.edit7,'string'));

t(1)=str2num(get(handles.edit2,'string'));
t(2)=str2num(get(handles.edit4,'string'));
t(3)=str2num(get(handles.edit6,'string'));
t(4)=str2num(get(handles.edit8,'string'));

k(1)=str2num(get(handles.edit9,'string'));
k(2)=str2num(get(handles.edit10,'string'));
k(3)=str2num(get(handles.edit11,'string'));
k(4)=str2num(get(handles.edit12,'string'));

ddA=str2num(get(handles.edit13,'string'));

```

```

ddt=str2num(get(handles.edit14,'string'));

dt=1/str2num(get(handles.edit15,'string'));

N=str2num(get(handles.edit16,'string'));

Dn=str2num(get(handles.edit19,'string'));

[x t]=imit(A,t,k,dt,ddA,ddt,N);
x=x+normrnd(0,Dn,length(x),1)';

data(1,:)=t;
data(2,:)=x;

set(handles.pushbutton1,'userdata',data);

axes(handles.axes1);
plot(t,x);
axis tight;
grid on;

function edit15_Callback(hObject, eventdata, handles)
% hObject      handle to edit13 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%          str2double(get(hObject,'String')) returns contents of
edit13 as a double

% --- Executes during object creation, after setting all
properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit13 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit16_Callback(hObject, eventdata, handles)
% hObject      handle to edit14 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit14 as text
%          str2double(get(hObject,'String')) returns contents of
edit14 as a double

% --- Executes during object creation, after setting all
properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
%          str2double(get(hObject,'String')) returns contents of
edit14 as a double

% --- Executes during object creation, after setting all
properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit43_Callback(hObject, eventdata, handles)
% hObject    handle to edit43 (see GCBO)

```

```

% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit43 as text
%          str2double(get(hObject,'String')) returns contents of
edit43 as a double

% --- Executes during object creation, after setting all
properties.
function edit43_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit43 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit42_Callback(hObject, eventdata, handles)
% hObject      handle to edit42 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit42 as text
%          str2double(get(hObject,'String')) returns contents of
edit42 as a double

% --- Executes during object creation, after setting all
properties.
function edit42_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit42 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit41_Callback(hObject, eventdata, handles)
% hObject      handle to edit41 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit41 as text
%         str2double(get(hObject,'String')) returns contents of
edit41 as a double

% --- Executes during object creation, after setting all
properties.
function edit41_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit41 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit40_Callback(hObject, eventdata, handles)
% hObject      handle to edit12 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%         str2double(get(hObject,'String')) returns contents of
edit12 as a double

% --- Executes during object creation, after setting all
properties.
function edit40_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit12 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.

```

```

%           See ISPC and COMPUTER.
if         ispc         &&         isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit39_Callback(hObject, eventdata, handles)
% hObject      handle to edit11 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%           str2double(get(hObject,'String')) returns contents of
edit11 as a double

% --- Executes during object creation, after setting all
properties.
function edit39_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit11 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if         ispc         &&         isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit38_Callback(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%           str2double(get(hObject,'String')) returns contents of
edit10 as a double

% --- Executes during object creation, after setting all
properties.
function edit38_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)

```

```

% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit37_Callback(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%         str2double(get(hObject,'String')) returns contents of
edit9 as a double

% --- Executes during object creation, after setting all
properties.
function edit37_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
xx=get(handles.pushbutton1,'userdata');
x=xx(2,:);
Fc=str2num(get(handles.edit17,'string'));
Fs=str2num(get(handles.edit18,'string'));
dt=1/str2num(get(handles.edit15,'string'));

```

```

xmod=ammod(x,Fc,Fs);
t=(0:(length(xmod)-1)).*dt;
axes(handles.axes1);
%plot(t,x);
%hold on;
plot(t,xmod);
%hold off;
axis tight;
grid on;
function edit19_Callback(hObject, eventdata, handles)
% hObject      handle to edit19 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit19 as text
%         str2double(get(hObject,'String')) returns contents of
edit19 as a double
% --- Executes during object creation, after setting all
properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit19 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

Додаток В

Теза конференції (копія)

УДК 621.373

Канд. техн. наук, доцент, В.Л. Дунець, Д.А. Стащук

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СІГНАЛІВ ДЛЯ ТЕСТУВАННЯ АЛГОРИТМІВ РОБОТИ РАДІОТЕХНІЧНИХ СИСТЕМ

L.V. Hvosnivska Ph.D., V.L. Dunets, A.P. Serputko

SIMULATION MODELING OF SIGNALS FOR TESTING ALGORITHM OF RADIOSYSTEMS

В сучасних ринкових умовах спостерігається значний прогрес у розвитку радіотехнічних систем (РТС) і широке їх застосування у різних галузях діяльності людини. Процедура проектування РТС включає у свою структуру етап експериментального налагодження і випробувань РТС шляхом відпрацювання алгоритмів обробки радіосигналів [1].

Натурні дослідження (випробування та налагодження) РТС зумовлені складністю інтерпретації отриманих результатів (отримують шляхом обробки радіосигналів) в умовах експерименту із зміни умовами та вимагають великих фінансових і часових затрат.

Одним із методів, який уможливорює процедуру якісного попереднього дослідження роботи РТС на базі діючих алгоритмів обробки радіосигналів із мінімальними затратами експерименту є метод імітаційного моделювання [2].

Аналізом наукових праць встановлено, що імітаційне моделювання здійснюється на базі імітаторів, в ядрі є яких лежить математична модель радіосигналів в каналах зв'язку у вигляді емпіричного, детермінованого та випадкового представлень.

Емпіричні моделі адекватно описують радіосигнали та дають змогу апроксимувати результати експериментальних вимірювань шляхом вимірювання радіосигналу в реальних РТС з його відновленням за результатами експерименту [3]. Така модель не дає змогу проводити дослідження РТС в широкому діапазоні умов (без впливу різного роду факторів). Детерміновані моделі описують фундаментальні уявлення про поширення радіосигналів [Соколова А.В., Борзова А.Б., Сухаревського О.І., Васильця В.А., Корнєєва Ю.А.] в геометричних та електричних середовищах з високим ступенем адекватності імітації. Проте модель такого типу не урахує у своїй структурі фактору випадковості, що є притаманним для умов реального експерименту.

Статистичні моделі описують радіосигнали як випадкові процеси у вигляді ймовірнісних моделей [4]. Перевагою відомих статистичних моделей [Введенського Б.А., Кловського Д.Д., Галкіна А.П., Самойлова А.Г., Басса Ф.Г., Фукса І.М., Кларка Р.Х. (Clarke R.H.), Потапова А.А.] є можливість адекватного опису радіосигналів із урахуванням фактору випадковості. Моделі такого типу знайшли найбільшого застосування при моделюванні і розробці імітаторів радіосигналів в РТС із різними середовищами їх передачі. Оскільки реальні радіотехнічні сигнали характеризують окрім випадковості та властивістю періодичності, тому відомі статистичні моделі не відображають ці властивості в поєднанні.

Тому розроблення імітаційної моделі радіосигналів із урахуванням властивостей випадковості із періодичності є актуальною науковою задачею при адекватному дослідженні алгоритмів обробки реальних РТС.

Реальні радіосигнали за своєю структурою є складними сигналами з n -ми кількостями складових з певною випадковістю та повторюваністю (періодичністю) T в часі.

Із урахування структури радіосигналів розроблено алгоритм його моделювання, який зображено на рис.1.

Згідно до алгоритму (рис. 1) спочатку моделюються n складових радіосигналу в межах їх часового діапазону k -го періоду, відповідно від 0 до T_{nk} , потім змодельовані складові розташовуються на часовій осі в залежності від їх послідовного розміщення, а часові області якими вони не належать заповнюються нулями згідно виразу:

Рис.1. Алгоритм моделювання



радіосигналів

$$\tilde{z}_k(t) = \sum_{n \in \mathbb{N}} \tilde{z}_{nk}(t), \quad t \in [0, T_k), \quad T_1 \neq T_2 \neq \dots \neq T_k, \quad (1)$$

$\tilde{z}_{nk}(t)$ - продовжені по часовій області складові радіосигналу $z_{nk}(t)$, $t \in [0, T_{nk})$ (заповнення нулями часових областей, яким не належать складові):

$$\tilde{z}_{nk}(t) = \begin{cases} z_{nk}(t), & t \in [T_{(n-1)k}, T_{nk}) \\ 0, & t \notin [T_{(n-1)k}, T_{nk}) \end{cases}, \quad (2)$$

де T_{nk} - тривалість n -ої складової на k -му періоді радіосигналу, $T_{1k} \neq T_{2k} \neq \dots \neq T_{kn}$.

Імітаційну модель n -ої складової радіосигналу на одному k -му періоді T_k побудовано у вигляді синусоїди із експонентційним затуханням на характерних часових інтервалах, яка враховує амплітуди складових A_{nk} та їх часові тривалості T_{nk} з елементами випадковості ψ_A , ψ_T :

$$\tilde{z}_{nk}(t) = (A_{nk} + \psi_A) \sin(2 \cdot \pi \cdot (t + \psi_T) \cdot f_{nk}) \cdot e^{-t \cdot K_{nk}} \cdot L_{nk}, \quad t \in [0, T_{nk}) \quad (3)$$

Імітаційна модель (3) адекватно описує радіосигналу із урахуванням складності його структури, елементів випадковості та повторюваності, що є характерним при дослідженні алгоритмів роботи реальних РТС.