

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна

Факультет (навчально-науковий інститут) радіофізики,
біомедичної електроніки та комп'ютерних систем

Кафедра прикладної електродинаміки

До захисту допущено кафедрою
прикладної електродинаміки
протокол № ____ від _____
в.о. завідувача кафедри

_____ Вадим ПЛАХТІЙ
(підпис) (ім'я, прізвище)
« ____ » _____ 202_р.

Кваліфікаційна робота
здобувача _____ другого (магістерського) _____ рівня вищої освіти
(першого (бакалаврського) / другого (магістерського))

ЧАСОВІ ДІАГРАМИ НАПРЯМЛЕНОСТІ ВИПРОМІНЮВАЧІВ
ІМПУЛЬСНИХ ПОЛІВ ДЛЯ СИСТЕМИ ПОЗИЦІОНУВАННЯ

(назва роботи)

Спеціальність (спеціалізація) 105 Прикладна фізика та наноматеріали
(код та найменування спеціальності; спеціалізації спеціальності - за наявності)

Освітня програма _____ Радіофізика і електроніка
(назва освітньої програми)

Виконавець _____ Олександра Задорожня
(підпис) (ім'я, прізвище)

Науковий керівник _____ Вадим Плахтій
(підпис) (ім'я, прізвище)

РЕФЕРАТ

Дипломний проєкт містить 28 рисунків, 53 сторінки, 25 бібліографічних посилань.

Проведено моделювання надширокосмугового комбінованого вібраторно-щілинного випромінювача типу Клевіна з метою отримання часових залежностей електричної компоненти електромагнітного поля в дальній зоні. Визначено межу дальньої зони для використовуваного імпульсного сигналу, яка становить приблизно 30 см.

Розроблено програмні засоби для автоматичного формування сітки проб та збору результатів моделювання в електродинамічному симуляторі CST Microwave Studio методом FDTD. Отримані часові залежності були структуровані, інтерпольовані та представлені у вигляді тривимірних графіків, що відображають просторову еволюцію імпульсного сигналу.

Продемонстровано можливість використання випромінювача типу Клевіна для задач позиціонування на основі надширокосмугових сигналів завдяки характерним змінюванням форми часових залежностей у різних точках простору.

**НАДШИРОКОСМУГОВА АНТЕНА, ЕЛЕКТРОМАГНІТНЕ ПОЛЕ,
ПОЗИЦІОНУВАННЯ, НЕСТАЦІОНАРНЕ ВИПРОМІНЮВАННЯ,
ВИПРОМІНЮВАЧ КЛЕВІНА**

ABSTRACT

The diploma project contains 28 figures, 43 pages, and 25 bibliographic references.

A simulation of a combined vibrator–slot ultra-wideband emitter of the Klevin type was performed in order to obtain the time-domain waveforms of the electric field component in the far-field region. The far-field boundary for the applied excitation pulse was determined to be approximately 30 cm.

Software tools were developed to automate the placement of field probes and data extraction within the CST Microwave Studio electromagnetic simulator using the FDTD method. The obtained time-domain signals were structured, interpolated, and visualized as three-dimensional plots representing the spatial evolution of the emitted pulse.

The results demonstrate the feasibility of using the Klevin-type ultra-wideband radiator for positioning tasks, as the time-domain signatures exhibit distinguishable variations depending on the observation point in space.

ULTRA-WIDEBAND ANTENNA, ELECTROMAGNETIC FIELD, POSITIONING, NON-STATIONARY RADIATION, CLAVIN RADIATOR.

ЗМІСТ

ЗМІСТ	4
1. Вступ	5
Анени для імпульсних сигналів	9
Напрявленість імпульсних антен у часовій області.....	12
Методи аналізу та моделювання часових діаграм.....	14
Застосування в системах позиціонування	14
1. Моделювання комбінованого надширокозмугового випромінювача типу Клевіна.....	16
2. Попередня обробка часових залежностей амплітуди електричної компоненти електромагнітного поля для різних кутів спостереження.....	27
3. Результати числового моделювання	30
4. Висновки	35
ЛІТЕРАТУРА.....	37
Додаток 1	40
Додаток 2	42
Додаток 3	43
Додаток 4.....	45
Додаток 5.....	48
Додаток 6.....	51

1. ВСТУП

Зараз, коли ІТ та штучний інтелект розвиваються так швидко точне позиціонування стає все важливішим. Воно використовується в багатьох сферах: від робототехніки та Інтернету речей (IoT) до радіолокаційних комплексів і систем безпеки. У робототехніці високоточне визначення координат є дуже важливим для автономної навігації та взаємодії з навколишнім середовищем. У сфері IoT позиціонування дозволяє інтегрувати сенсори та мобільні пристрої у єдину мережу, що дає змогу краще контролювати та керувати всім у “розумних” будинках, виробничих лініях чи міських інфраструктурах. Радіолокаційні системи, у свою чергу, використовують точне визначення просторових параметрів для виявлення та відстеження об’єктів навіть у складних умовах, таких як туман, багатопрореневість чи наявність перешкод.

Для точного позиціонування зазвичай використовують імпульсні поля надширокосмугового діапазону (UWB). [1] Вони працюють за рахунок надкоротких імпульсів тривалістю наносекунди, що забезпечує надзвичайно широку смугу частот. Завдяки цьому досягається висока точність вимірювання відстані та координат, яка може становити кілька сантиметрів. Імпульсні сигнали мають низьку дисперсію та добре зберігають форму при поширенні, що дозволяє ефективно відокремлювати прямий сигнал від відбитих. Це особливо важливо для позиціонування у приміщеннях, де багатопрореневість є основною проблемою. Крім того, UWB-технологія характеризується енергоефективністю, адже короткі імпульси з низькою потужністю зменшують рівень інтерференції з іншими системами зв’язку.

Імпульсні сигнали мають кілька ключових властивостей, які роблять їх корисними для позиціонування. По-перше, ширина спектра дозволяє досягати високої просторової роздільної здатності, що забезпечує можливість розрізняти близько розташовані об’єкти. По-друге, часові характеристики імпульсів дають змогу реалізувати точні вимірювання затримки сигналу, що є основою для визначення координат. По-третє, здатність розрізняти різні об’єкти робить

імпульсні системи ефективними навіть у складних середовищах із великою кількістю відбиттів. [1]

Тривалість імпульсу обернено пропорційна ширині смуги передавання, тому чим ширша смуга, тим точніше можна визначити момент приходу сигналу. Якщо час приходу імпульсу відомий із малою невизначеністю, стає можливим точне оцінювання відстані, яку пройшов сигнал, а отже, і визначення координат передавача за допомогою триангуляційних методів. Для систем із потенційною смугою пропускання 7,5 ГГц часова роздільна здатність може становити близько 133 пс, що відповідає просторовій похибці приблизно 4 см. Навіть при помірній смузі 500 МГц часова невизначеність збільшується до 2 нс, що еквівалентно просторовій похибці близько 60 см. [2] Таким чином, UWB-технологія дозволяє реалізувати субметрову точність визначення положення, особливо за умови комбінування вимірів від декількох приймачів. Остаточна точність координат залежить як від сумарних похибок вимірювання відстаней, так і від специфічних похибок обраної методики позиціонування. Наразі для позиціонування UWB використовується переважно визначення часу обертання (ToF), визначення часу приземлення (TDoA) та визначення площини приземлення (AoA). Перші два методи працюють незалежно, тоді як останній метод AoA зазвичай поєднується з ToF або TDoA. [3]

Принципи роботи радіосистем позиціонування

Позиціонування на основі ToF

Метод ToF ґрунтується на вимірюванні відстані. UWB-трекер ініціює процес визначення дистанції до кожної опорної станції (стаціонарного приймально-передавального вузла, який використовується як точка відліку для визначення координат). Після отримання даних про відстань розраховується позиція об'єкта.

У найпростішому (нульовому) режимі достатньо однієї опорної станції. Для одновимірного позиціонування необхідно щонайменше дві станції, для

двовимірного зазвичай використовують три й більше, хоча в окремих спеціалізованих режимах можливе застосування лише двох. Тривимірне позиціонування потребує щонайменше чотирьох опорних станцій. [3]

У випадках одновимірного позиціонування станцію часто розташовують над об'єктом, і визначення координати здійснюється лише за її допомогою.

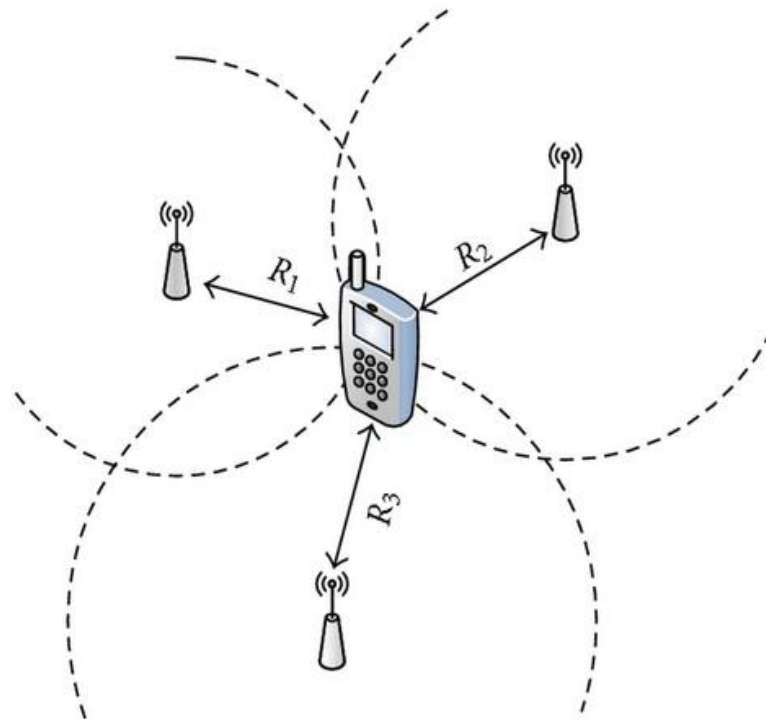


Рисунок 1 - Представлення техніки локалізації ToA [4]

Позиціонування на основі TDoA

Метод TDoA передбачає визначення місця розташування цілі шляхом вимірювання різниці часу приходу сигналів на дві різні опорні станції або антенно-блокові системи. На відміну від ToF, тут не використовується абсолютний час проходження сигналу, що знижує вимоги до синхронізації між джерелом сигналу та приймачами, але водночас підвищує вимоги до синхронізації між самими станціями. Для отримання двох незалежних TDoA необхідно щонайменше три опорні станції, а позиція мобільного трекера визначається як точка перетину гіпербол, сформованих цими вимірюваннями. [3]

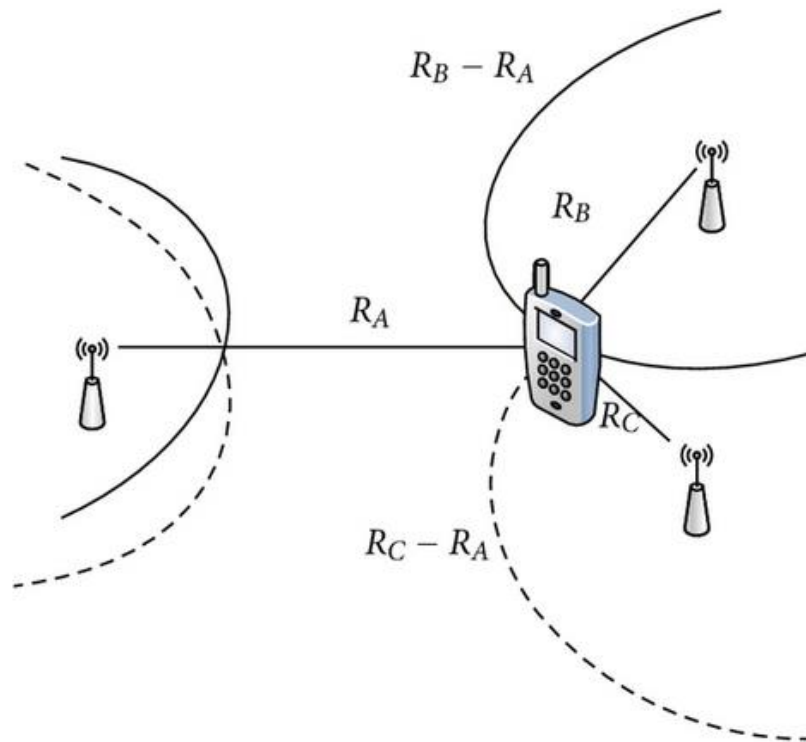


Рисунок 2 - Представлення методу локалізації TDoA [4]

Позиціонування за кутом приходу (AoA)

Метод AoA базується на аналізі різниці фаз сигналів. Його рідко застосовують окремо, адже точність визначення координат зменшується зі збільшенням відстані від опорної станції через обмежену кутову роздільну здатність. Проте AoA може комбінуватися з ToF, що дозволяє здійснювати позиціонування навіть за допомогою однієї станції. [3]

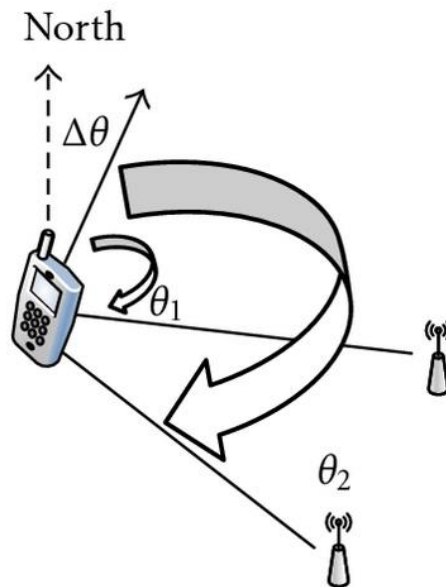


Рисунок 3 - Представлення методу локалізації AoA.[4]

Антени для імпульсних сигналів

Важливу роль у функціонуванні систем на основі імпульсних сигналів відіграють антени. Для UWB-систем застосовуються спеціалізовані конструкції, зокрема TEM рупорні антени, які забезпечують широку смугу пропускання та низьку дисперсію. Вони оптимізовані для випромінювання коротких імпульсів і дозволяють мінімізувати втрати енергії та відбиття. Правильне узгодження апертури антени сприяє покращенню коефіцієнта відбиття, посилення та рівномірному розподілу енергії сигналу. Це безпосередньо впливає на точність позиціонування та стійкість системи до зовнішніх перешкод.

Рупорні антени використовують для формування та випромінювання надкоротких імпульсів. Зазвичай вони мають пірамідальну або конічну форму. На високих частотах антени збуджуються від хвилеводу, а на нижчих - від коаксіальної лінії з чвертьхвильовим штирем, що ускладнює подачу потужних повторюваних імпульсів. Оскільки різні спектральні складові випромінюються з фазового центра, який зміщується всередині рупора, імпульс розтягується в часі.

Це знижує ефективність роботи в перехідних режимах. Крім того, для частот нижче 1 ГГц потрібні антени значних розмірів. [5]

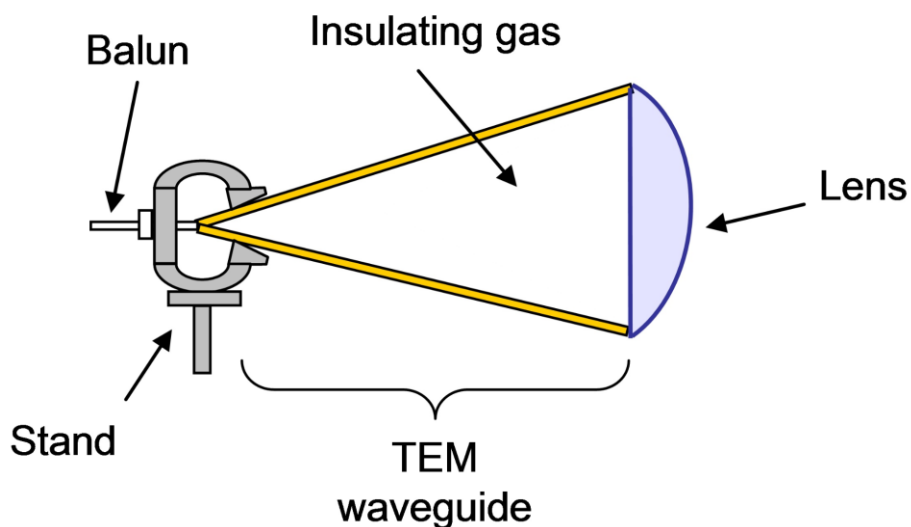


Рисунок 4 - Схематичний розріз рупора TEM [5]

Одним із найбільш придатних антенних елементів для випромінювання надширокопasmових імпульсних сигналів ще є антена типу Vivaldi, або TSA (Tapered Slot Antenna). Ця конструкція ґрунтується на використанні розширюючої щілини, форма якої визначає ширину робочої смуги частот. Завдяки плавному експоненційному розкриттю щілини забезпечується надширока смуга пропускання, що є ключовою вимогою для передавання ультракоротких імпульсів без суттєвих спотворень. [6]

Антени TSA характеризуються високим коефіцієнтом посилення та вузьким головним променем, що дозволяє формувати стабільну та добре спрямовану діаграму випромінювання.



Рисунок 5 - Модель TSA антени [6]

Коли антени мають достатньо великі розміри, їх можна віднести до класу антен з поширенням хвилі вздовж конструкції. Біконічна антена складається з двох конусів, розташованих один навпроти одного своїми вершинами (або основами — у варіанті дисконічної антени).[5] Конуси виготовляють або з листового металу шляхом згортання, або у вигляді каркаса з металевих стрижнів. Підведення енергії здійснюється у точці суміщення вершин конусів. Один конус фактично працює як половина біконічної антени, встановленої над заземленою площиною. Робочий частотний діапазон таких антен охоплює приблизно область від 300 МГц до 18 ГГц.

Фазова швидкість (або фазова затримка) визначає швидкість поширення окремої частотної компоненти сигналу. Вона залежить лише від швидкості хвилі на конкретній частоті. Тобто, поняття фазової швидкості стосується лише однієї частотної складової сигналу. [7]

Групова швидкість (або затримка) описує поширення всього сигналу, а не окремої частоти. Вона визначає швидкість руху хвильової форми, що складається з багатьох частотних компонентів із різними фазовими швидкостями. Зазвичай розглядають криву групової швидкості в певному

діапазоні, яка показує різницю фаз між складовими. На практиці інженери виражають це як часову затримку, тому групова затримка вимірюється у часі. Групову затримку можна визначати різними способами. Один із них полягає у вимірюванні діелектричної проникності та розрахунку швидкості хвилі на різних частотах. Також застосовують точкові вимірювання діелектричної проникності з подальшим розрахунком групової затримки за широкосмуговою моделлю.

Фазова стабільність відноситься до стану або якості, за якої фаза сигналу залишається постійною без коливань, зумовлених часом або змінами навколишнього середовища.[8]

Напрявленість імпульсних антен у часовій області

Частотна діаграма направленості (Radiation Pattern) - визначає розподіл енергії випромінювання антени у просторі для гармонічного сигналу на певній частоті. Використовується для оцінки підсилення, спрямованості та ефективності антени у вузькій смузі частот.

Часова діаграма направленості (time-domain radiation pattern) — це розподіл поля (електричного чи магнітного) у просторі та в часі для заданого імпульсу $E(\theta, \varphi, t)$. Ця характеристика показує, що у кожному напрямку (θ, φ) імпульс може мати різну форму, затримку, амплітуду і тривалість.

Таким чином, часова діаграма спрямованості дозволяє оцінити не лише напрямок випромінювання, а й часові особливості поширення імпульсу. Це критично важливо для систем позиціонування, де точність визначається різницею приходу сигналів у різних точках простору.

Для аналізу часової діаграми антени використовують нормовану імпульсну відповідь, яка показує зміну форми сигналу в різних напрямках незалежно від амплітуди. Часова діаграма описує, як антена спотворює імпульс,

і містить дані про групову затримку, дисперсію та фазову стабільність, на відміну від класичної частотної діаграми. [8]

У аналізі в часовому просторі застосовується концепція віртуального джерела, що враховує зміщення точки випромінювання залежно від частоти та напрямку. Це важливо для точних вимірювань і калібрування систем позиціонування. Форма імпульсної відповіді визначається дисперсією, шириною смуги, струмом у елементах антени, резонансами та поляризацією. Чим менше спотворень, тим точніше антена передає сигнал, що критично для UWB-систем позиціонування, радіолокації та зв'язку. Часова діаграма враховує не лише просторовий розподіл енергії, а й часову структуру сигналу. У різних напрямках імпульс приходить із різним зсувом, що впливає на точність визначення координат. [8]

Імпульс може зазнавати розтягнення, зміни фронту чи додаткових коливань через неоднорідність частотної характеристики та взаємодію з середовищем. Дисперсія та обмежена смуга пропускання призводять до деформацій, згладжування та зниження часової роздільності.

Однією з проблем при вимірюванні діаграм спрямованості антен у дальній зоні є багатопроменевість (multipath) — наявність відбитих сигналів, які спотворюють результати. Ці відбиті компоненти мають різні довжини шляху, тому їх можна відокремити, якщо освітлювати антену короткими імпульсами. Альтернативний підхід полягає у проведенні вимірювань у частотній області та застосуванні перетворення Фур'є для переходу до часової області. У цьому випадку інтерференційні компоненти можна видалити за допомогою часового вікна (time-domain gate), після чого результати знову перетворюються у частотну область для отримання покращених діаграм спрямованості у режимі безперервної хвилі (CW). [10]

Практичний приклад реалізації такого підходу наведено у дослідженнях Time Domain Antenna Pattern Measurements Andrew M. Predoehl , де було

модернізовано дальньопольовий антенний полігон і впроваджено систему збору та обробки даних у часовій області. Результати показали значне покращення точності вимірювань діаграм спрямованості після застосування часової обробки. [10]

Спотворення імпульсу у різних напрямках є наслідком комплексної взаємодії геометрії антени, її частотних характеристик та умов поширення хвилі. Для систем позиціонування це означає, що часові діаграми спрямованості повинні враховувати не лише просторовий розподіл енергії, а й часову структуру сигналу, щоб мінімізувати похибки у визначенні координат.

Методи аналізу та моделювання часових діаграм

Метод кінцевих різниць у часовій області (FDTD) [11] використовується для моделювання хвильових процесів у часі, а перетворення Фур'є дозволяє переходити до частотної області. У обчислювальній електромагнетичці він став популярним завдяки простій реалізації: електричні та магнітні поля обчислюються на двох зміщених сітках із застосуванням центрального диференціювання рівнянь Максвелла, що забезпечує точність другого порядку.

Метод особливо ефективний для задач з неоднорідними матеріалами, тому широко використовується у біомедичних дослідженнях та для аналізу широкосмугових антен. Одне моделювання з імпульсним збудженням дає змогу отримати частотну характеристику антени. Крім того, FDTD добре масштабується на паралельних обчисленнях і може значно прискорюватися за рахунок використання графічних процесорів.

Застосування в системах позиціонування

Багато виробників систем позиціонування в реальному часі застосовують готові надширокосмугові (UWB) модулі, що знижує витрати та спрощує розробку.

Qorvo DW1000 – один із перших інтегрованих UWB-чипів, який забезпечує точність до 10 см, роботу на шести каналах (3,5–6,5 ГГц), швидкість до 6,8 Мбіт/с та дальність понад 200 м у LoS-умовах. Підтримує TWR і TDoA-алгоритми та інтерфейс SPI. [12]



Рисунок 6 - Qorvo DW1000

Qorvo DW3000 – вдосконалена версія, сумісна зі стандартами IEEE 802.15.4-2011/4z. Має кілька варіантів (3110–3220), інтегровану антену, стійкість до багатопроменевих завад, підтримку каналів 5 і 9, швидкість 850 кбіт/с та 6,8 Мбіт/с, інтерфейс SPI до 38 МГц і сумісність із Apple U1/U2. [12]

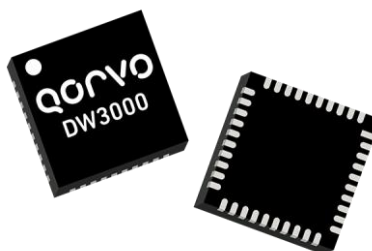


Рисунок 7 - Qorvo DW3000

1. МОДЕЛЮВАННЯ КОМБІНОВАНОГО НАДШИРОКОСМУГОВОГО ВИПРОМІНЮВАЧА ТИПУ КЛЕВІНА

Першим буде розглянуто моделювання надширокосмугового випромінювача типу Клевіна. В основі його є прототип випромінювач Клевіна, який поєднує випромінювачі магнітного та електричного типу, а саме щілину в екрані або хвилеводі та вібратори, проте частіше всього йде мова про вузькосмугову антену. Таке розташування вібраторів з щілиною дає змогу помітно збільшити ефективність випромінювання щілини [13], модель якого зображена на рис. 8

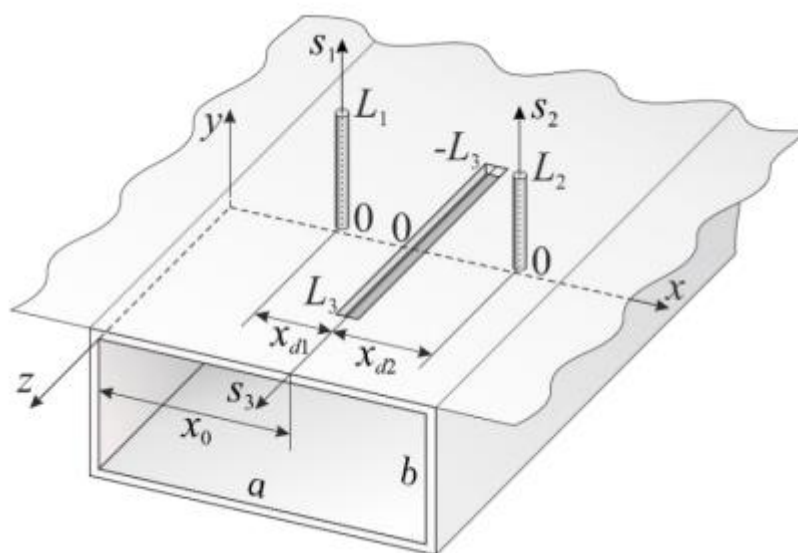


Рисунок 8 - Вібраторно-щілинний випромінювач на основі порожнистого прямокутного хвилеводу [13]

В роботі [14] був запропонований надширокоусмуговий варіант вібраторно-щілинного випромінювача, один із його варіантів зображений нижче на рис. 9.

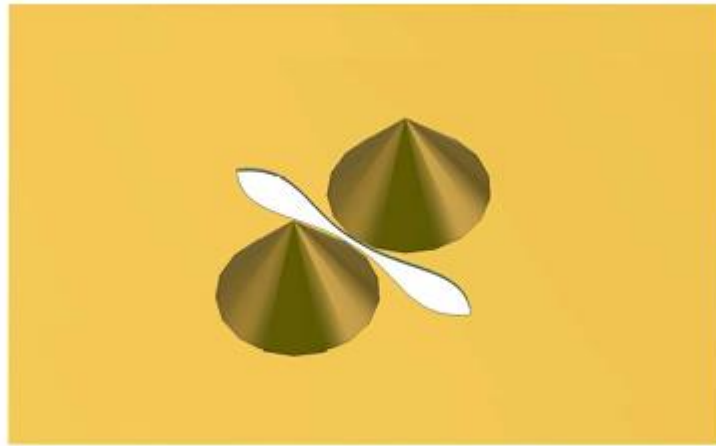


Рисунок 9 - Надширокосмуговий варіант вібраторно-щілинного випромінювача [14]

Він був розроблений на основі комбінації надширокосмугового випромінювача магнітного типу, а саме надширокосмугової щілини запропонованої Барнсом [15], а вібратори були замінені на конуси, основи яких лежать на екрані (рис. 10).



Рисунок 10 - Надширокосмугова магнітна щілинна антена Барнса [16]

Такий випромінювач також був розглянутий в складі антенної решітки [9-10] (рис. 11).

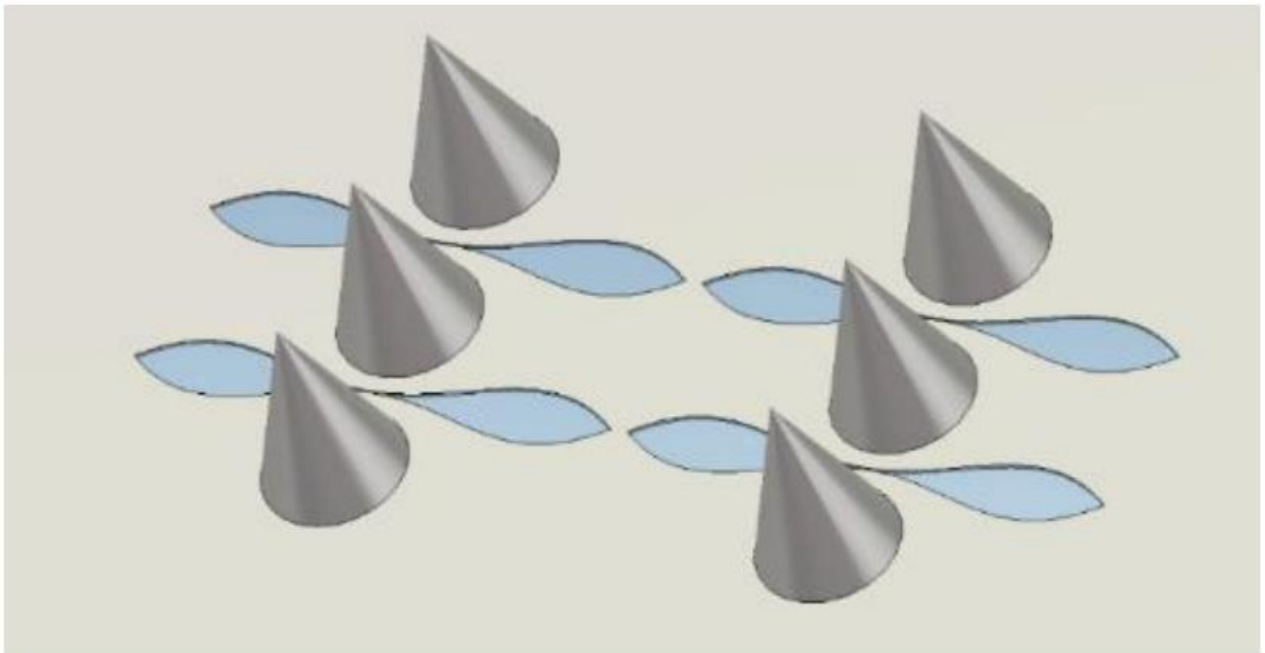


Рисунок 11 - Надширокосмугова антенна решітка [18]

В даній дипломній роботі був змодельований останній із варіантів розроблений науковцями [19], який також був розглянутий в складі антенної решітки. Проте для задачі позиціонування нам достатньо одного елемента.

Моделювання було здійснено в часовому просторі методом FDTD за допомогою студентської версії програмного пакету CST Microwave studio [20]. Зображення моделі цього випромінювача представлено на рис. 12.

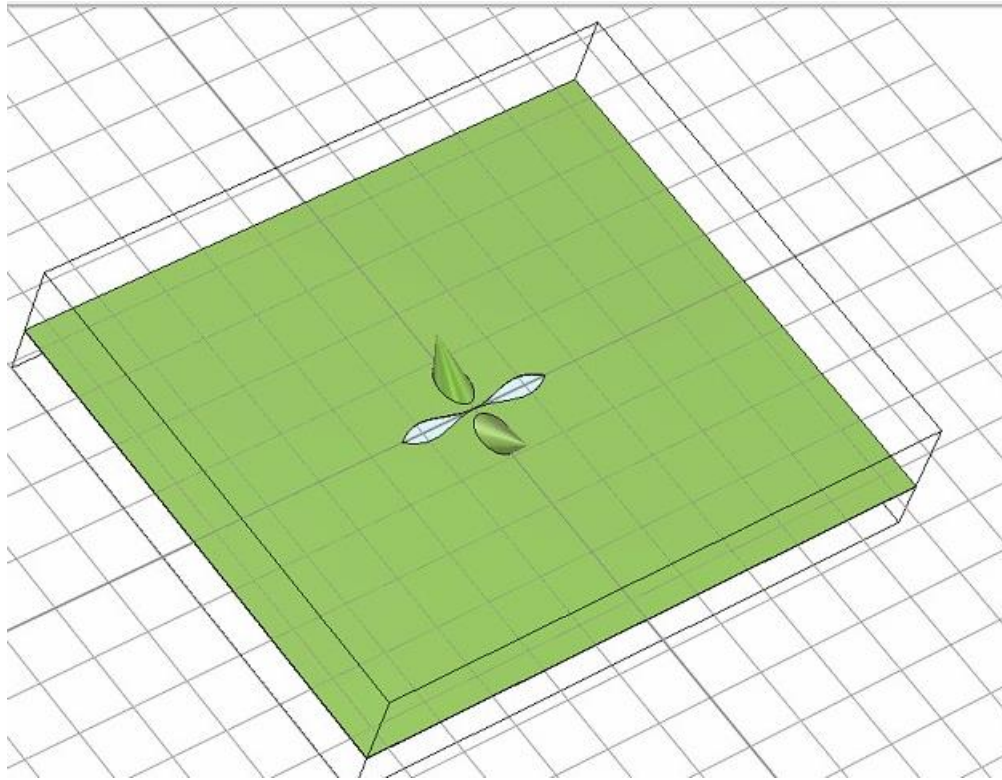


Рисунок 12 - Оптимізований надширокосмуговий вібраторно-щільовий випромінювач типу Клевіна.

Цілю цього моделювання є отримання часових залежностей амплітуди електричної компоненти електромагнітного поля під багатьма кутами спостереження, а саме в секторі кутів $\theta = 0 - 90^\circ$ та $\varphi = 0 - 90$ градусів.

CST Microwave Studio дає можливість це зробити автоматично за допомогою макросів. Для цього треба зайти в Macros – Solver – Monitors and Probs – Define Probs from ASCII-file. Це рішення потребує текстового файлу із наступною структурою (рис. 13):

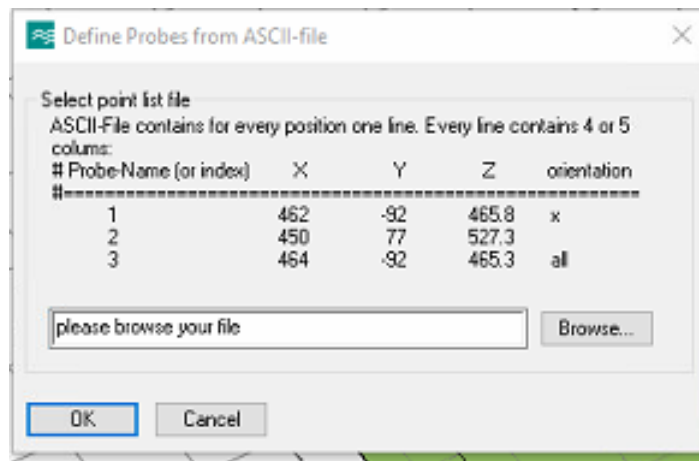


Рисунок 13 - Створення проб електричного поля за допомогою макросів.

Для автоматизації побудови такого файлу була використана програма на мові Python, вона наведена в Додатку 1. Цей код створює два файли, в одному записує координати проб в декартовій системі координат, в іншому паралельно записує значення кутів θ та φ в сферичній системі координат. Це буде необхідно в подальшому для побудови графіків та застосування класифікатором.

Розміри випромінювача були взяті із публікації [19] і повторивши проект бачимо, що по рівню $KСХН = 3$ мінімальна робоча частота цього випромінювача складає 1,55 ГГц. Залежність КСХН представлена на рис. 14.

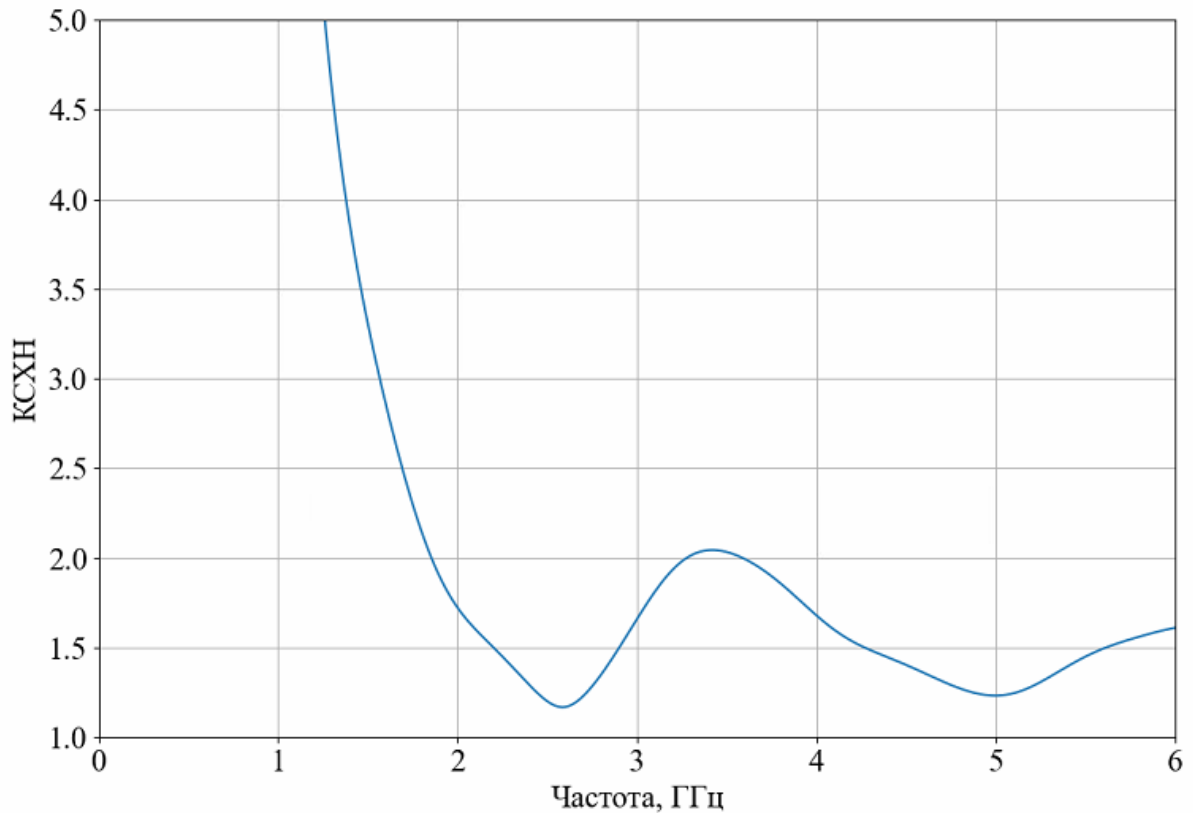


Рисунок 14 - Частотна залежність параметру КСХН надширокосмугового вібраторно-щільового випромінювача типу Клевіна

Для коректного вимірювання часових залежностей компонент електричного поля необхідно визначити дальню зону для даного випромінювача. Для вузькосмугових систем частіше всього застосовують наступне визначення [21]:

$$R < \frac{4D^2}{\lambda} \quad (1)$$

де D – лінійний розмір антени, λ – довжина хвилі.

Проте в нас надширокосмугова імпульсна антена, і для неї було запропоновано визначення межі дальньої зони в роботі Л.Г. Содіна [22] у наступному вигляді:

$$R < \frac{D^2}{c\tau_e} \quad (2)$$

де τ_e - деяка ефективна тривалість імпульсу.

Для нашого випадку розміром D будемо вважати довжину щілини, яка складає 0,14 м, а тривалість імпульсу будемо брати з із задачі позиціонування [15-16] на імпульсних електричних полях, для яких і рахується даний випромінювач: $\tau_e = 0,22$ нс. Ефективна тривалість збуджуючого імпульсу була розрахований за найбільшою тривалістю, яка була розглянута в роботах [15-16] по рівню амплітуди імпульса 0,707. На рис. 15 він вказаний як сигнал 2,5 ГГц.

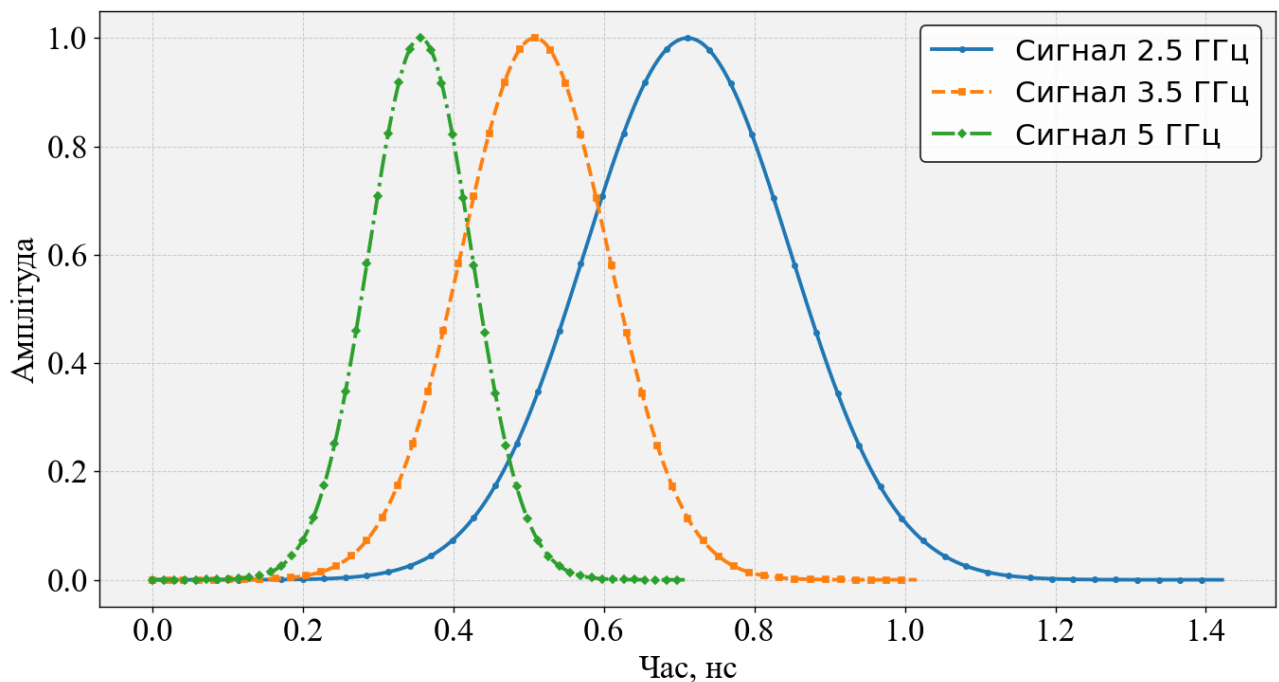


Рисунок 15 - Часові залежності збуджуючих імпульсів, які були використані в роботі для системи позиціонування на логоперіодичних антенах [24]

Враховуючи всі значення вище, розрахуємо межу дальньої зони для цієї антени $R < 30$ см. Як результат, довжина радіус-вектора пробників електричного поля відносно центру антени було взята рівною 40 см. Також важливо відмітити,

що для цього випромінювача буде розглянуто амплітуду електричну компоненту поля Y , яка направлена вздовж електричної компоненти цієї комбінованої антени, а саме в поперек щілини та вздовж напрямку розташування конусів (рис. 16).

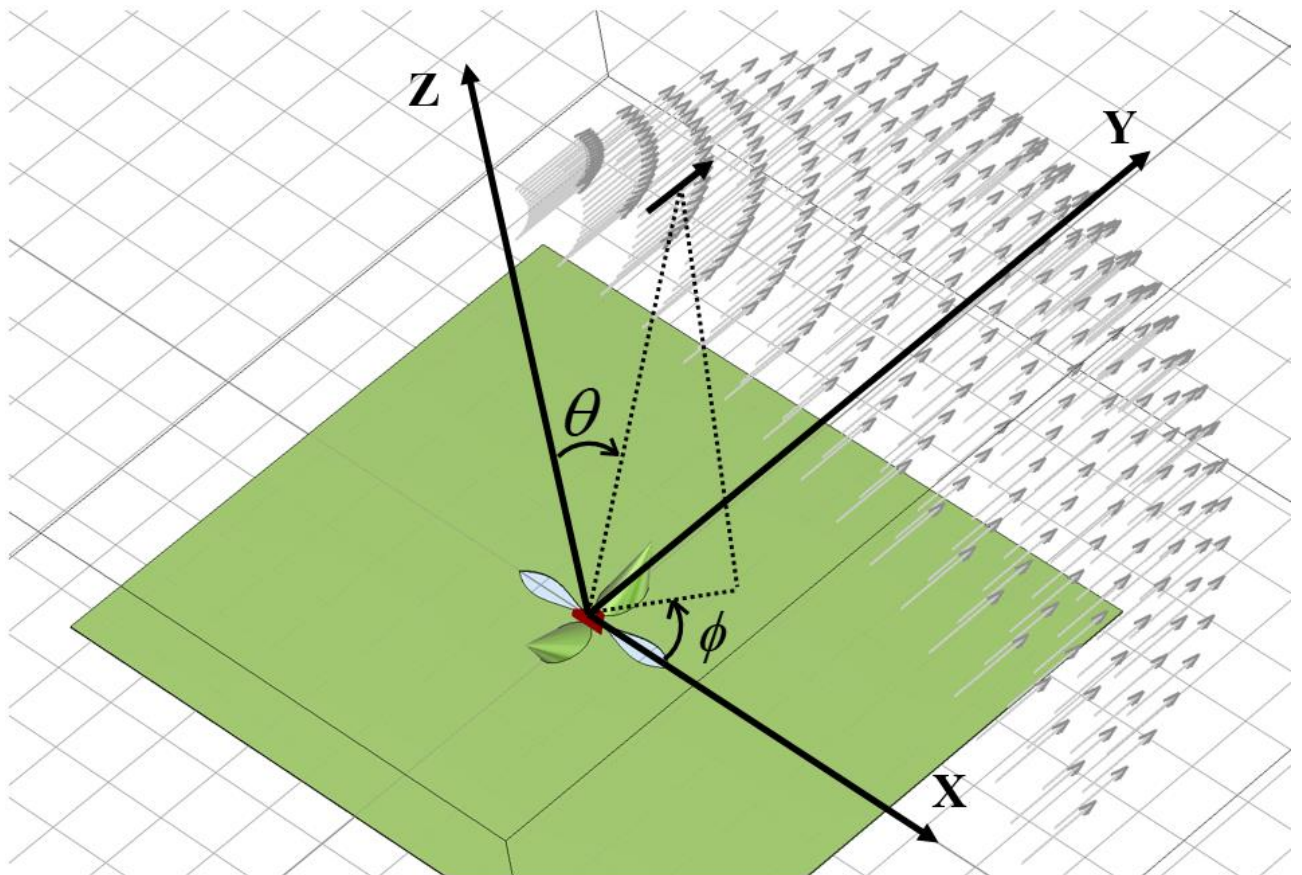


Рисунок 16. Постановка задачі

Збуджуючий імпульс, яким буде протестована антена виглядає натсупним чином – це Гаусовий імпульс з тривалістю 0,22 нс (рис. 17).

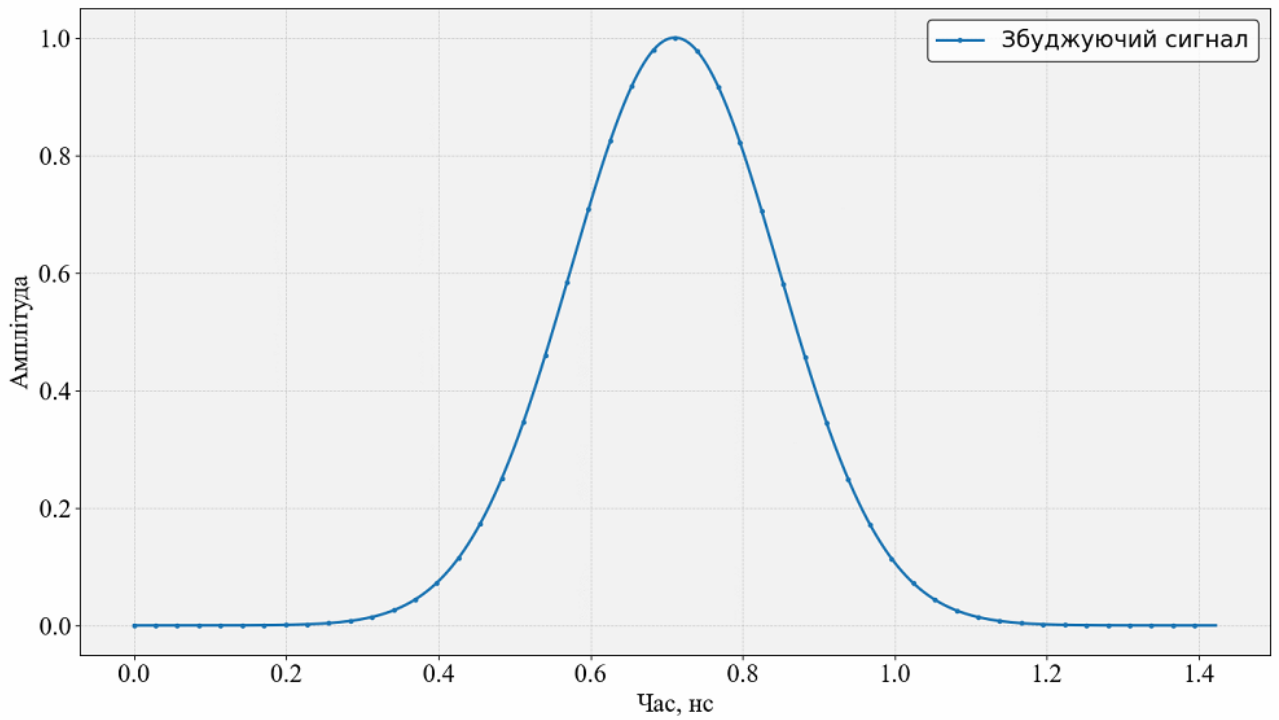


Рисунок 17. Часова залежність збуджуючого сигналу

В результаті розрахунку можемо поглянути на всі часові залежності отримані від проб електричного поля (рис. 18), які ми розташували навколо антени. Їх тут близько трьох сотень.

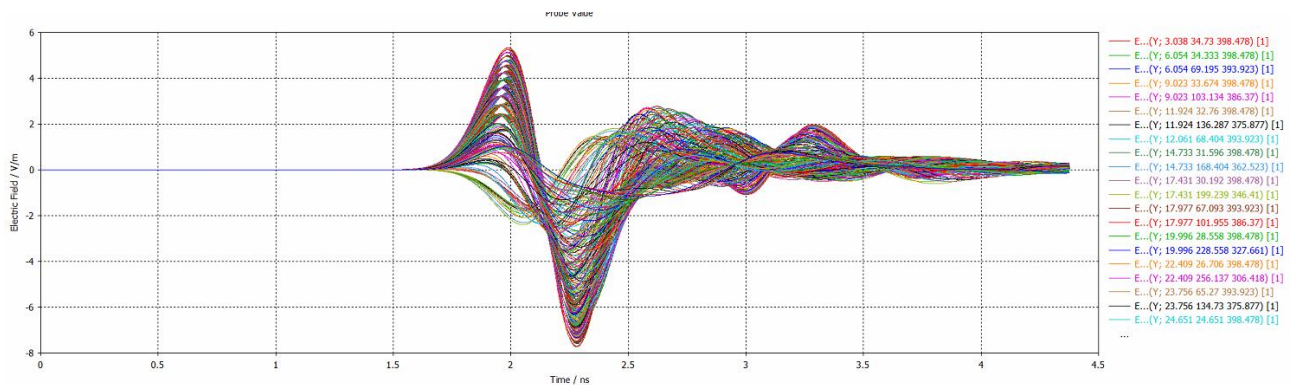


Рисунок 18 - Часові залежності для різних кутів спостереження

Для автоматичного експорту всіх розміщених раніше проб електричного поля необхідно скористатись макросами в CST Microwave studio, а саме написати програму на мові VBA (Visual Basic for Applications) (Додаток 2). Для цього потрібно знайти меню Macros- Open VBA macro editor (рис. 19)

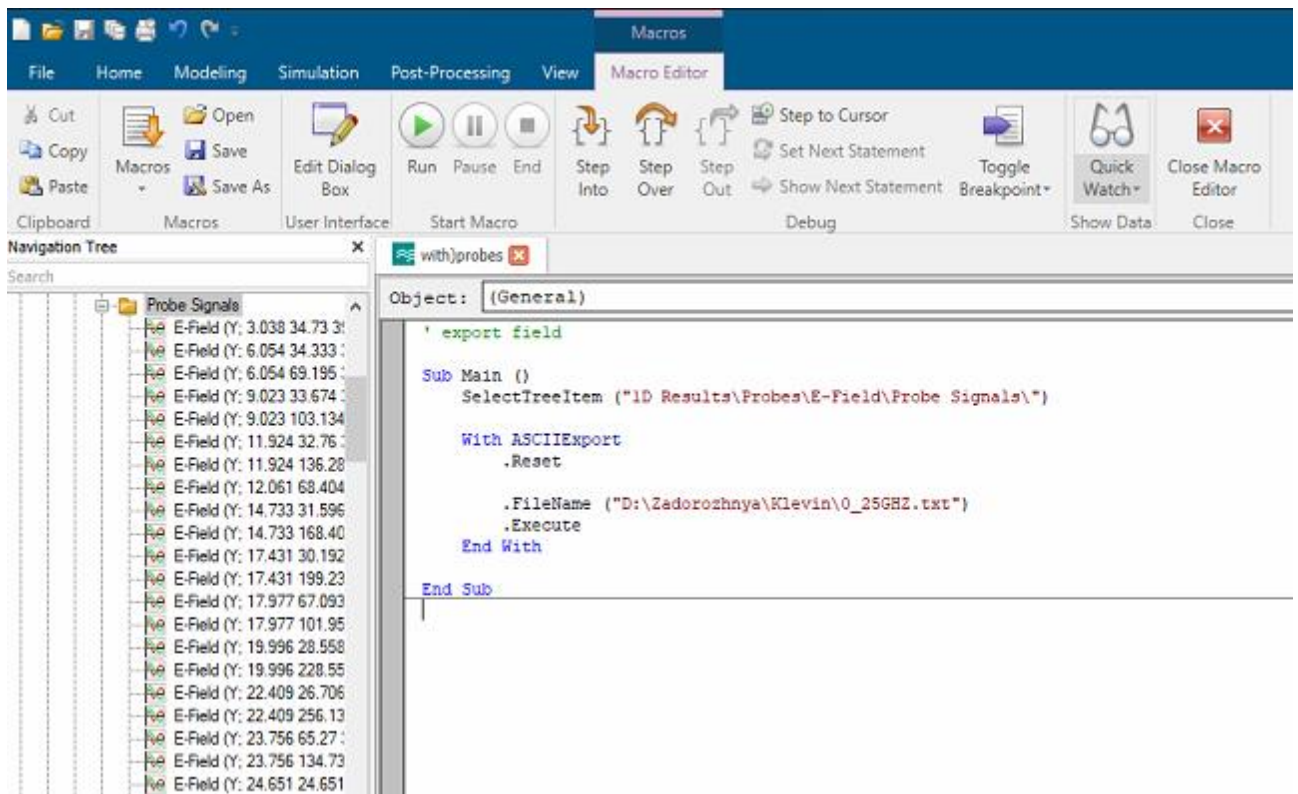


Рисунок 19 - Вигляд VBA редактора в CST Microwave Studio

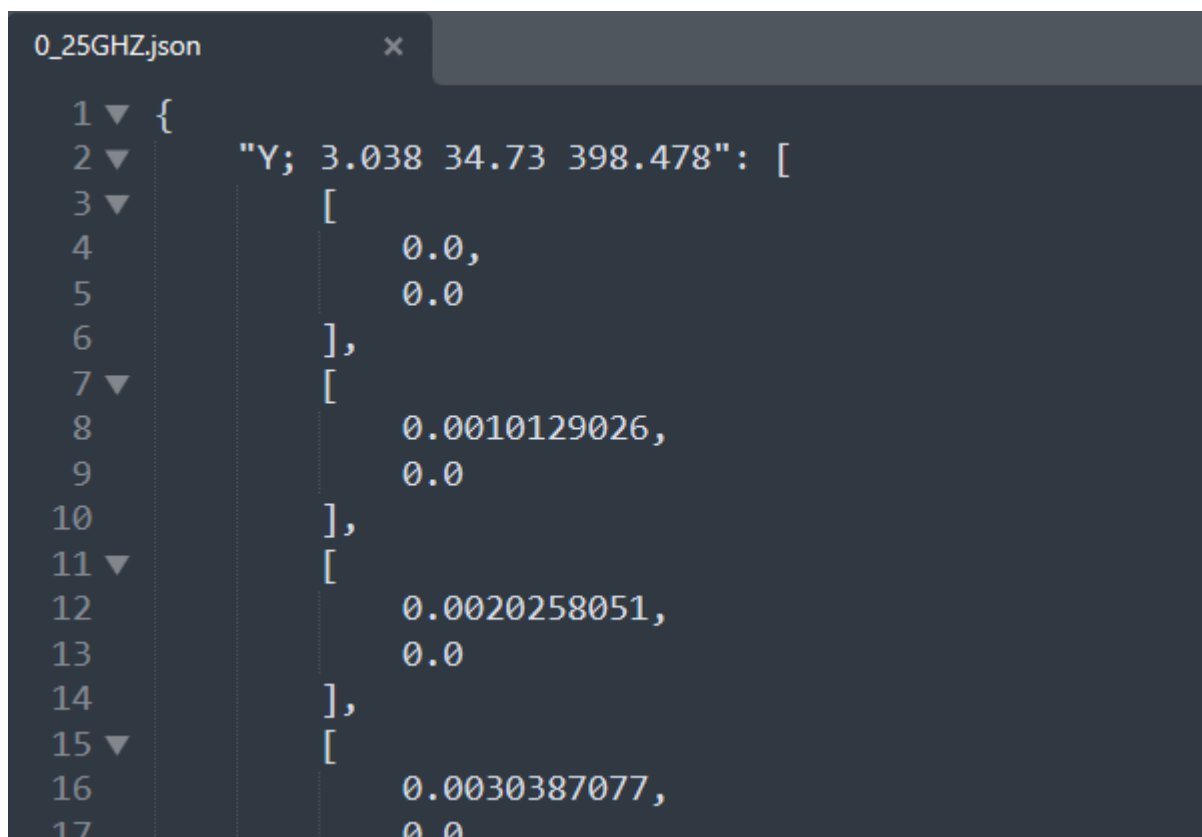
В результаті ми отримаємо великий файл з розширення .txt розміром в 70 Мб. Структура його представлена нижче, всі часові залежності додаються в текстовий файл одна за одною (рис. 20). Ці данні потребують попередньої обробки.

```
0_25GHZ_тест.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
-----
Time / ns      E-Field (Y; 3.038 34.73 398.478) [1]/real
-----
0              0
0.0010129026  0
0.0020258051  0
0.0030387077  0
0.0040516103  0
0.0050645128  0
0.0060774154  0
0.007090318   0
0.0081032206  0
-              -
-              -
-              -
-----
Time / ns      E-Field (Y; 6.054 34.333 398.478) [1]/real
-----
0              0
0.0010129026  0
0.0020258051  0
0.0030387077  0
-----
Windows (CRLF)
```

Рисунок 20 - Приклад структури експортованого масиву часових залежностей амплітуди електричного поля

2. ПОПЕРЕДНЯ ОБРОБКА ЧАСОВИХ ЗАЛЕЖНОСТЕЙ АМПЛІТУДИ ЕЛЕКТРИЧНОЇ КОМПОНЕНТИ ЕЛЕКТРОМАГНІТНОГО ПОЛЯ ДЛЯ РІЗНИХ КУТІВ СПОСТЕРЕЖЕННЯ

Для подальшої роботи з цими даними їх необхідно структурувати. Було запропоновано наступну структуру даних, яка в результаті зберігається в форматі Json. Для реалізації цього був написаний парсер, який наведений в Додатку 3. Результатом є JSON-структура виду ключ–значення, де ключем є значення з декартової системи координат у форматі строки, а значенням - масив упорядкованих пар чисел, що представляють часові залежності сигналу (рис 21).



```
0_25GHZ.json x
1 {
2   "Y; 3.038 34.73 398.478": [
3     [
4       0.0,
5       0.0
6     ],
7     [
8       0.0010129026,
9       0.0
10    ],
11    [
12     0.0020258051,
13     0.0
14    ],
15    [
16     0.0030387077,
17     0.0
```

Рисунок 21 - Фрагмент запропонованої структури даних

Далі потрібно інтерполювати дані. Програма, завдяки якій це було зроблено наведено в Додатку 4. Для подальшої зручної роботи з часовими залежностями вони мають мати однакову довжину та однаковий крок по часу. За

умовчуванням програма CST Microwave Studio надає масив даних, у якого що довжина масиву по часу, що крок по часу змінюється від розрахунку до розрахунку. Тому існує необхідність інтерполювати всі часові залежності, щоб привести їх до нормального вигляду. Для цього був використана програма, яка наведена в Додатку 4. В нашому випадку, тривалість розрахунку становила приблизно 4,3 нс. Загальний час моделювання можна точно дізнатись із графіку залежності енергії, яка знаходиться в розрахуноковому об'єму (рис. 22). Для задач, які розв'язуються методом FDTD одним із критеріїв зупинки розрахунку є значення в дБ енергії, що наявна в розрахуноковому просторі, де 0 дБ каже, про те, що енергія повністю в проекті, а зупинкою розрахунку може бути значення, наприклад -30 дБ, що каже про те, що енергія або поглинулась в матеріалах в просторі, або була поглинута ідеальними поглиначами на межах проекту. Крок по часу був обраний 0,02 нс.

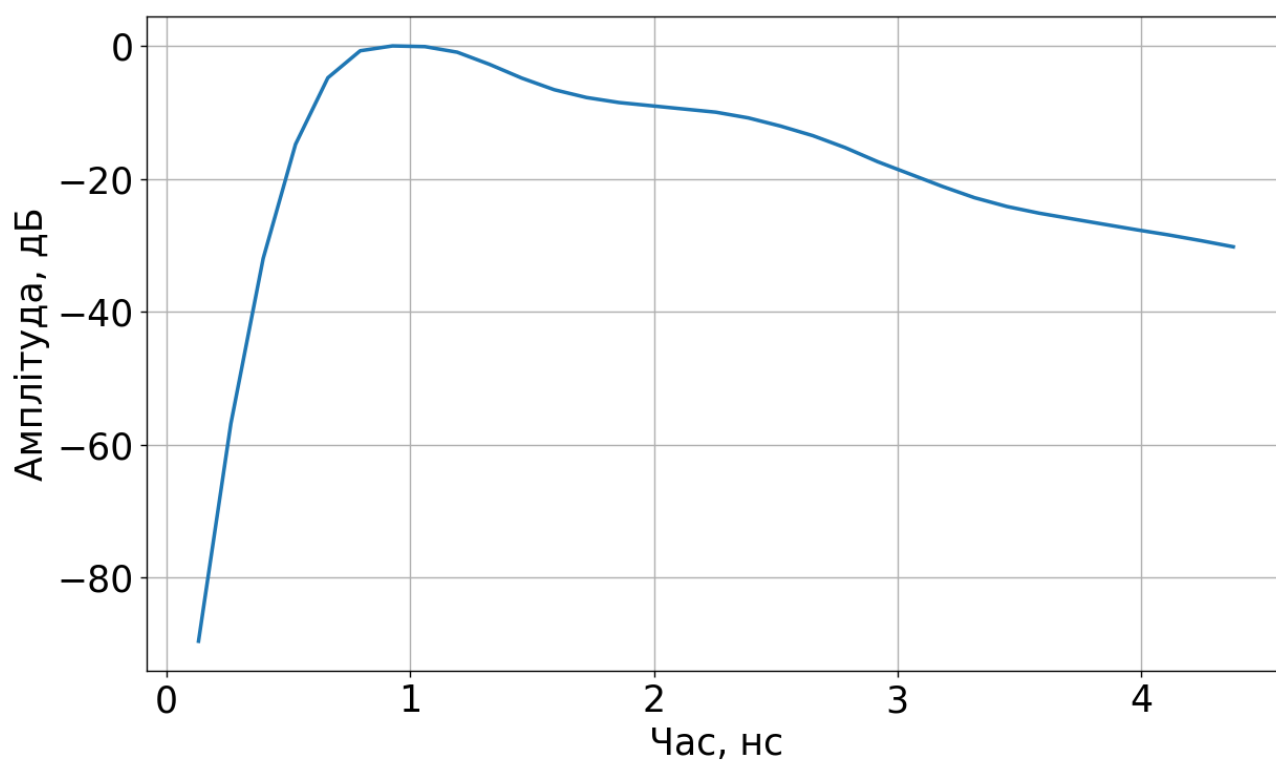


Рисунок 22 - Залежність енергії, яка знаходиться в розрахуноковому об'ємі від часу

Після інтерполяції необхідно в структурі даних виконати парсинг ключів і замінити декартові координати на кути в сферичній системі координат θ та φ . Для цього будуть використані вже інтерпольована дані та файли «легенди», які були створені ще на етапі формування координат поля в текстовий файл (Додаток 1). Цей парсер наведений в Додатку 5. Результатом роботи є структура даних у вигляді словника, з парами ключ-значення, де ключ це комбінації кутів в сферичній системі координат, а значення це часові залежності у вигляді списків зі значенням часу та амплітуди електричної компоненти поля (рис. 23).

```
1 {
2     "85 5": [
3         [
4             0.02,
5             0.0
6         ],
7         [
8             0.04,
9             0.0
10        ],
11        [
12            0.06,
13            0.0
14        ],
15        [
16            0.08,
17            0.0
18        ]
19    ]
20 }
```

Рисунок 23 - Фрагмент обробленої структури даних

3. РЕЗУЛЬТАТИ ЧИСЛОВОГО МОДЕЛЮВАННЯ

Маючи дану структуру даних маємо змогу в повній мірі будувати часові залежності - діаграми спрямованості імпульсного випромінювача. Всі наступні графіки були виконані за допомогою програми, яка наведена в Додатку 6

Спочатку поглянемо, як змінюється часова форма амплітуди електричної компоненти електромагнітного поля при $\varphi = 45^\circ$ та $\theta = 5^\circ - 85^\circ$ (рис. 24). Видно, що в більшості своїй при куті $\varphi = 45^\circ$ і зміні куті θ помітна сильна зміна часової форми післяімпульсних коливань. Проте форма головного імпульса майже не змінюється.

3D поверхня: залежність від θ при $\varphi = 45$

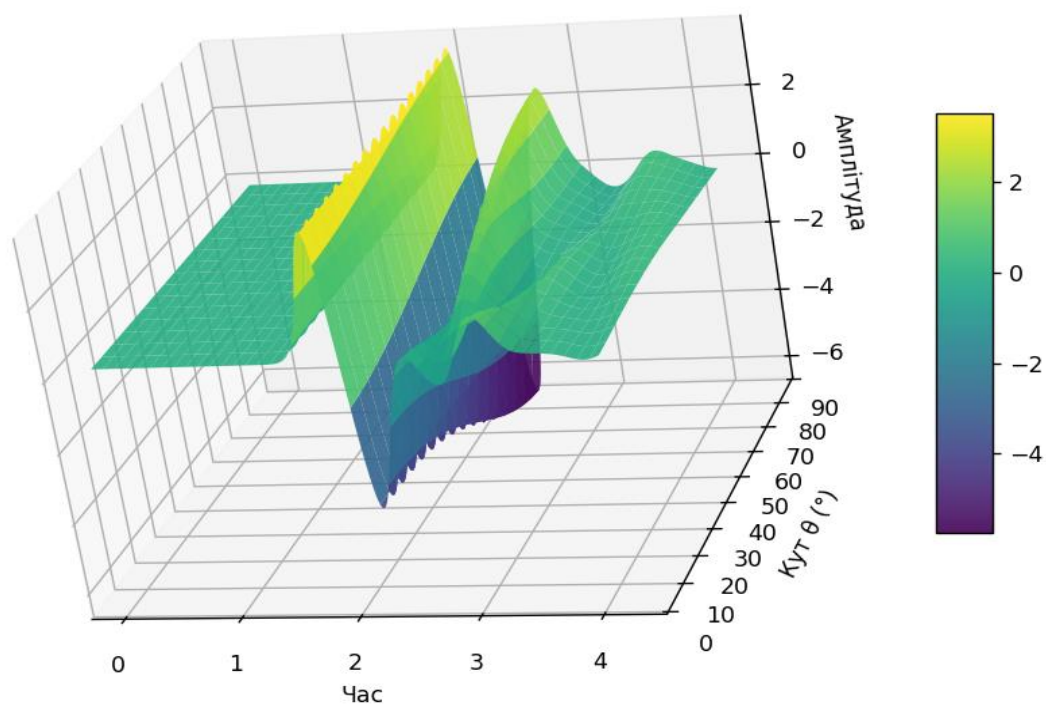


Рисунок 24 - Часові залежності амплітуди електричної компоненти електромагнітного поля на відстані 40 см від випромінювача та при кутах спостереження $\varphi = 45^\circ$ та $\theta = 5^\circ - 85^\circ$

Тепер навпаки: $\theta = 45^\circ$ та $\varphi = 5^\circ-85^\circ$ (рис. 25). При зміні кута φ помітна зміна не тільки післяімпульсних коливань, а й самої форми основног оімпульсу.

3D поверхня: залежність від φ при $\theta = 45$

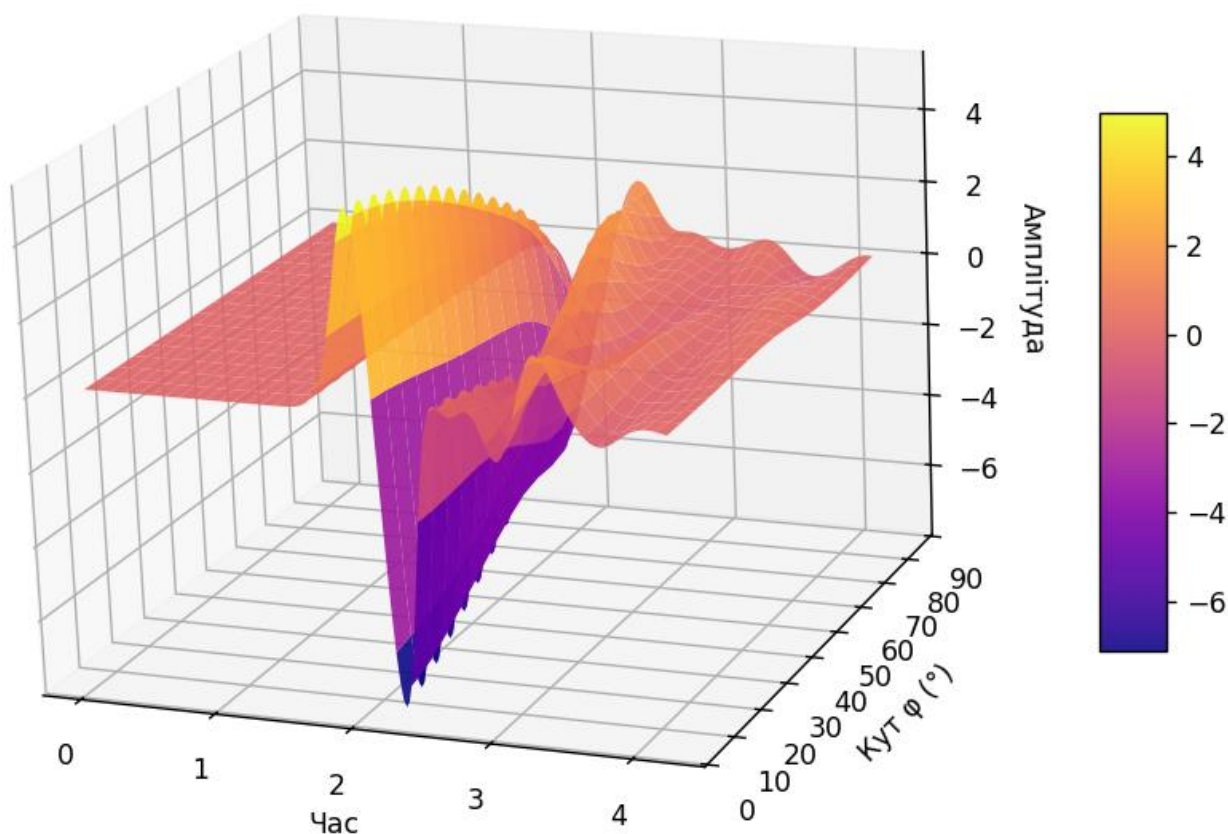


Рисунок 25 - Часові залежності амплітуди електричної компоненти електромагнітного поля на відстані 40 см від випромінювача та при кутах спостереження $\theta = 45^\circ$ та $\varphi = 5^\circ-85^\circ$.

Цікавий випадок де кут спостереження θ має невелике значення наприклад 5 градусів при зміні кута φ (рис. 26). Бачимо як сильну зміну післяімпульсних коливань, так і зміну часової залежності основного імпульса.

3D поверхня: залежність від φ при $\theta = 5$

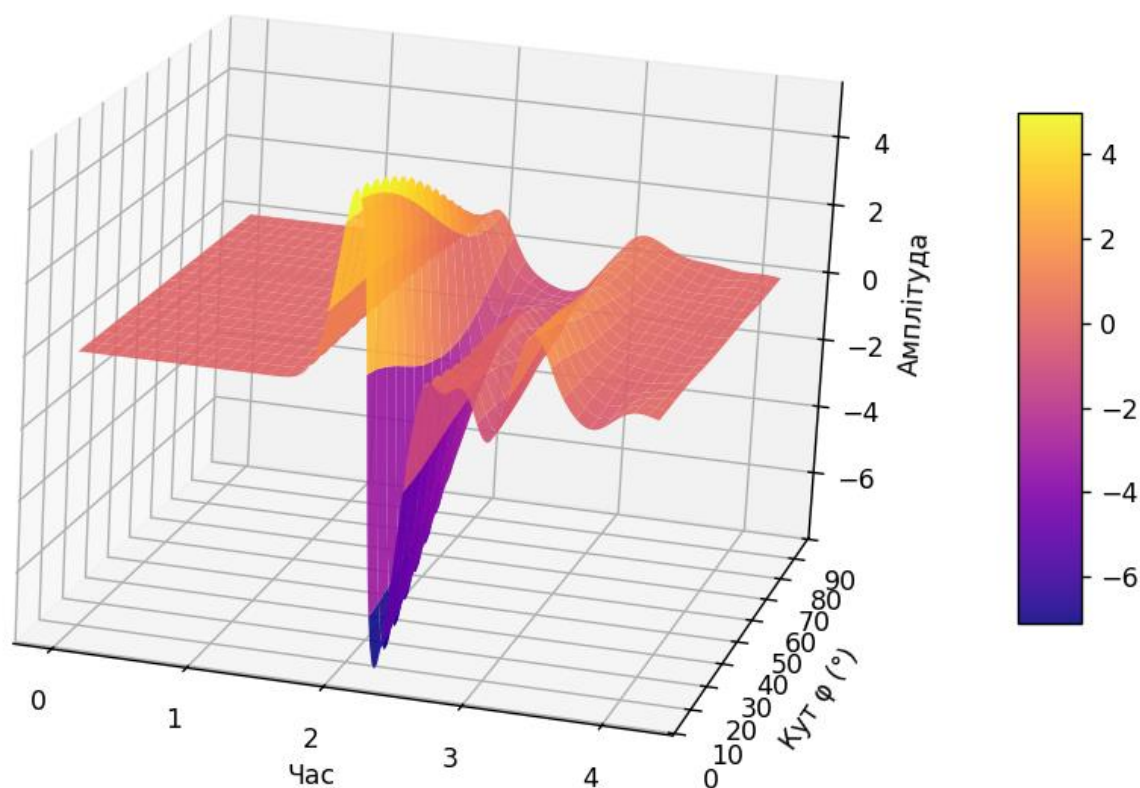


Рисунок 26 - Часові залежності амплітуди електричної компоненти електромагнітного поля на відстані 40 см від випромінювача та при кутах спостереження $\theta = 5^\circ$ та $\varphi = 5^\circ-85^\circ$.

На рис. 27 зображено можливе використання двох імпульсних випромінювачів в складі системи позиціонування. Тут смайликом позначено приймальну систему, яка реєструє електромагнітні імпульси, класифікує їх а має по визначеним кутам приходу сигналів та відомим положенням передавальних антен розраховувати своє положення в просторі. Як показали попередні дослідження [25], при використанні двох імпульсних антен для позиціонування як на площині так і в просторі, малі кути спостереження θ показують дуже погану точність. Щонайменше це пов'язано з тим, що при комбінації двох кутів

спостереження від обох антен обидва кута θ будуть мати мале значення, це буде вказувати на те, що спостерігач знаходиться дуже далеко від антенних систем.

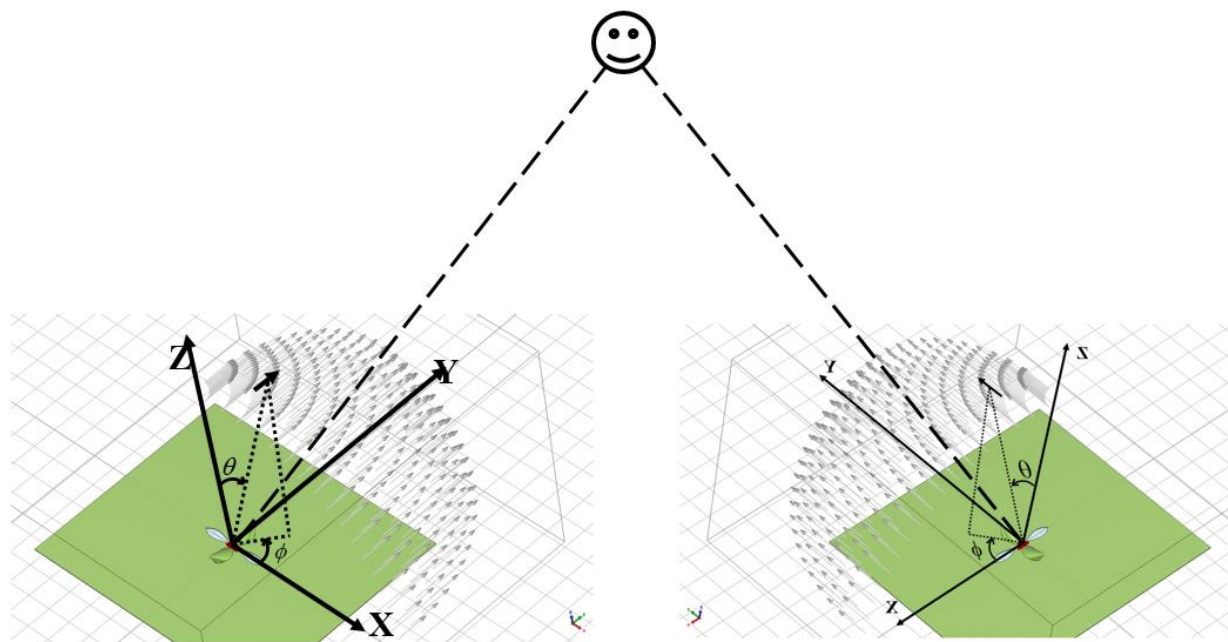


Рисунок 27 - Геометрія задачі позиціонування на надширокосмугових сигналах

Приймальна система, яка реєструє імпульси від двох антен і однозначно класифікувавши може визначити кути спостереження відносно кожної з антен. Для того, щоб орієнтуватись в площині, достатньо двох кутів - θ_1 від першої антени та θ_2 від другої антени. А для позиціонування в просторі (X, Y, Z) , необхідно 3 кути, θ_1 , ϕ_1 та θ_2 або ϕ_2 . Геометрія задачі продемонстрована нижче. Визначивши кути спостереження θ_1 , ϕ_1 будується промінь з початком від першої антени, а визначивши наприклад ϕ_2 з відомими координатами другого передавач будується площина. Шукана координата буде знаходитись в точці перетину променя та площини, як зображено на рис. 28.

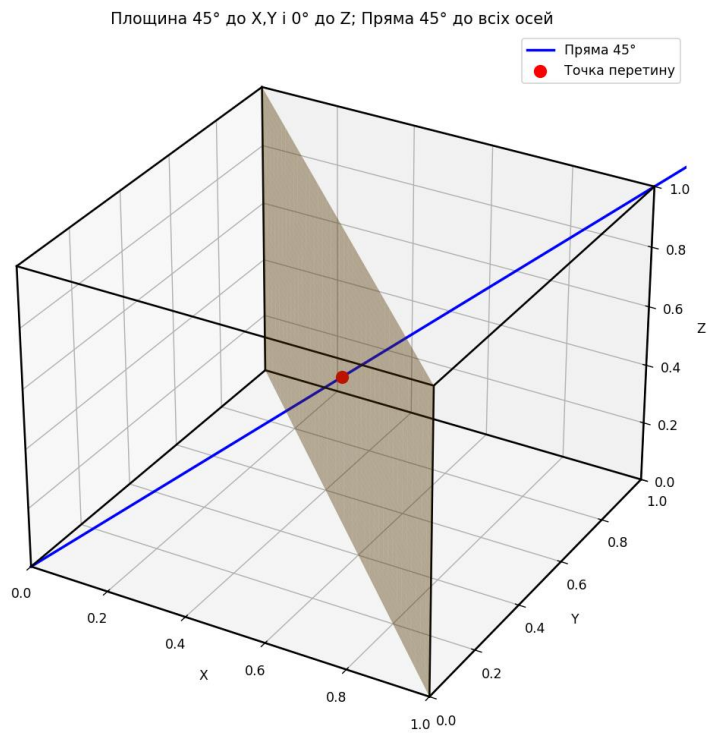


Рисунок 28 - Геометрія задачі позиціонування в просторі

Для відносного позиціонування достатньо знати відстань між передавальними антенами, для абсолютного позиціонування також необхідно знати точні координати передавачів. Проте вони можуть їх передавати за допомогою імпульсного кодування на приймач.

4. ВИСНОВКИ

У межах виконання дипломної роботи було проведено комплексне дослідження особливостей формування та поширення імпульсних електромагнітних полів надширокосмуговими випромінювачами, а також проаналізовано можливості їх застосування для задач просторового позиціонування. На основі детального огляду літератури узагальнено сучасні підходи до моделювання імпульсних антен та їх часових характеристик, що дозволило сформулювати вимоги до моделі надширокосмугового джерела та параметрів його збудження.

Для чисельних розрахунків був опанований електродинамічний симулятор CST Microwave Studio у часовому просторі методом FDTD. Особливу увагу приділено автоматизації побудови моделей та розміщення вимірювальних проб, що дало змогу значно прискорити процес формування великої вибірки симуляційних даних. У роботі реалізовані програмні засоби, які здійснюють автоматичне створення сітки точок спостереження, запуск моделювання та подальший імпорт часових залежностей.

Як надширокосмуговий випромінювач був обраний комбінований вібраторно-щілинний елемент типу Клевіна, для якого побудовано повноцінну електродинамічну модель. Проведене моделювання дало змогу визначити межу дальньої зони для імпульсного збудження, яка для використаного сигналу становить близько 30 см. Це дозволило коректно задати позиції проб електричного поля та забезпечити відповідність моделі умовам далекого поля.

Отримані часові залежності з усіх проб були впорядковані, інтерпольовані та подані у вигляді тривимірних графічних структур, що відображають еволюцію форми сигналу в просторі. Таке подання дозволило проаналізувати характер спотворень імпульсу при різних кутах спостереження та оцінити потенціал

вибраного випромінювача для задачі визначення координат приймальних пристроїв.

Проведені обчислювальні експерименти демонструють, що імпульсний випромінювач типу Клевіна формує часові сигнатури, чутливі до зміни просторового положення точки спостереження. Це відкриває можливість використання такого випромінювача в системах позиціонування на базі надширококутних сигналів, зокрема у поєднанні з методами машинного навчання. Представлена у роботі візуалізація просторових залежностей є свідченням того, що часові форми сигналів можуть містити достатній обсяг інформації для визначення напрямку або області можливого розташування приймача.

Отже, поставлені в роботі завдання були повністю виконані: проведено аналіз наукових джерел, створено модель надширококутного випромінювача, реалізовано автоматизовану систему збору даних, отримано та опрацьовано часові характеристики поля й на їх основі сформовано об'ємні графічні представлення. Отримані результати можуть бути використані як основа для подальшого розвитку систем просторового позиціонування, зокрема для побудови нейромережових алгоритмів класифікації та оцінки координат на основі імпульсних електромагнітних сигналів.

ЛІТЕРАТУРА

1. I. Oppermann, M. Hämäläinen, J. Iinatti, UWB: Theory and Applications. Wiley, 2004, doi:
https://www.academia.edu/18200382/UWB_Theory_and_Applications
2. S. Sung, H. Kim, J.-II Jung: Accurate Indoor Positioning for UWB-Based Personal Devices Using Deep Learning, doi:
https://www.researchgate.net/publication/368967116_Accurate_Indoor_Positioning_for_UWB-Based_Personal_Devices_Using_Deep_Learning
3. Lansitec: UWB Positioning Technology, doi:
<https://www.lansitec.com/uwb-positioning-technology/>
4. L. Bras, N. Borges Carvalho, P. Pinho, L. Kulas, K. Nyka: A Review of Antennas for Indoor Positioning Systems, Dec. 2012, doi:
<https://onlinelibrary.wiley.com/doi/10.1155/2012/953269>
5. B. Cadilhon, B. Cassany, P. Modin, J. Diot, V. Bertrand and L. Pecastaing: Ultra Wideband Antennas for High Pulsed Power Applications, Nov. 2010, doi:
<https://www.intechopen.com/chapters/17812>
6. К. А. Трубочанінова: Модель антени для випромінювання надширокосмугових сигналів, 2020, doi:
https://www.researchgate.net/publication/347961799_MODEL_ANTENI_DLA_VIPROMINUVANNA_NADSIROKOSMUGOVIIH_SIGNALIV/fulltext/6382f18fc2cb154d29304289/MODEL-ANTENI-DLA-VIPROMINUVANNA-NADSIROKOSMUGOVIIH-SIGNALIV.pdf
7. Cadence System Analysis: Phase Delay vs. Group Delay: What's the Difference? doi:
<https://resources.system-analysis.cadence.com/blog/phase-delay-vs-group-delay-whats-the-difference>

8. Totoku: What is phase stability? Explaining the basics, importance, and the difference with phase margin, 2025, doi:
<https://en.totoku.co.jp/special-contents/column/coaxial24/>
9. B. Scheers, M. Acheroy and A. Vander Vors: Chapter 4. Time domain characterisation of antennas by normalised impulse response, doi:
<https://www.sic.rma.ac.be/~scheers/Papers/chapter4.pdf>
10. Predoehl, Andrew M.: Time domain antenna pattern measurements, 1996, doi:
<https://vtechworks.lib.vt.edu/items/04eb3fad-083f-4b68-b9ac-d722ff94a576>
11. D. W. Ward, K. A Nelson: Finite Difference Time Domain (FDTD) Simulations of Electromagnetic Wave Propagation Using a Spreadsheet, Oct.2005, doi:
https://www.researchgate.net/publication/2170283_Finite_Difference_Time_Domain_FDTD_Simulations_of_Electromagnetic_Wave_Propagation_Using_a_Spreadsheet
12. M. Faeik Ruzaij Al-Okby, S. Junginger, T. Roddelkopf and K. Thurow: UWB-Based Real-Time Indoor Positioning Systems: A Comprehensive Review, 2024, doi:
<https://www.mdpi.com/2076-3417/14/23/11005>
13. С. Л. Бердник, «Формування електромагнітних полів комбінованими вібраторно-щілинними випромінюючими структурами в електродинамічних об'ємах з імпедансними границями», дис. д-ра фіз.-мат. наук, спец. 01.04.03, Харків. нац. ун-т ім. В. Н. Каразіна, Харків, Україна, 2021
14. Oleksandr Dumin, P. Fomin, Vadym Plakhtii, and N. Mikhail, "Ultrawideband Combined Monopole-Slot Radiator of Clavin Type," Sep. 2020, doi:
<https://doi.org/10.1109/diped49797.2020.9273399>.
15. M. Barnes, "Ultra-Wideband Magnetic Antenna," U.S. Patent 6,091,374.
16. H. G. Schantz, "UWB magnetic antennas," vol. 3, pp. 604–607, Mar. 2004, doi:
<https://doi.org/10.1109/aps.2003.1219920>.
17. P. Fomin, O. Dumin, V. Plakhtii, and N. Mikhail, "UWB Antenna Arrays with the Monopole-Slot Radiator of Clavin Type," 2021 IEEE 3rd Ukraine Conference on

- Electrical and Computer Engineering (UKRCON), pp. 258–261, Aug. 2021, doi: <https://doi.org/10.1109/ukrcon53503.2021.9575282>.
18. П.Г. Фомін, «Надширокосмугові антенні решітки на випромінювачах Клевіна», кваліфікаційна робота магістра, спец. 105 «Прикладна фізика та наноматеріали», освітньо-професійна програма «Радіофізика і електроніка», Харків. нац. ун-т ім. В. Н. Каразіна, Харків, Україна, 2023.
 19. Oleksandr Dumin, Vadym Plakhtii, Oleksandra Skvortsova, and Viktor Katrich, “Tilted Cone-Enhanced Ultrawideband Monopole-Slot Radiators in Antenna Arrays,” pp. 122–127, Sep. 2024, doi: <https://doi.org/10.1109/diped63529.2024.10706144>.
 20. “CST Studio Suite,” Dassault Systèmes, Jul. 25, 2023. <https://www.3ds.com/products/simulia/cst-studio-suite>
 21. C. A. Balanis, *Antenna Theory: Analysis and Design*, 4th ed. Hoboken, NJ, USA: Wiley, 2016.
 22. L.G. Sodin, “Characteristics of pulsed radiation from antennas (An electromagnetic missile)”, *Radiotekhnika i Elektronika*, vol. 36, no. 5, p. 1014–1022.
 23. V. Plakhtii, O. Dumin, L. Capineri, G. Pochanin, and D. Havrylenko, “3D UWB Positioning System on Data Processing by ANN,” 2025 13th International Workshop on Advanced Ground Penetrating Radar (IWAGPR), pp. 1–5, Jul. 2025, doi: <https://doi.org/10.1109/iwagpr65621.2025.11109019>.
 24. Є. В. Гришин, Система позиціонування в просторі на основі надширокосмугових хвиль, кваліфікаційна робота магістра, спец. 105 – Прикладна фізика та наноматеріали, Харківський національний університет імені В. Н. Каразіна, Харків, Україна, 2024, 43 с.
 25. В. А. Плахтій, «Надширокосмугові електромагнітні поля в задачах розпізнавання підповерхневих об’єктів штучними нейронними мережами», дисертація доктора філософії, Харків. нац. ун-т ім. В. Н. Каразіна, Харків, Україна.

ДОДАТОК 1

```
import math

# Параметри
radius = 400 # радіус у сферичній системі координат, мм
phi_step = 5 # крок по  $\phi$  (азимутальний кут), градусів
theta_step = 5 # крок по  $\theta$  (зенітний кут), градусів
z_level = 0 # рівень Z
filename = 'output_semi_sphere.txt' # ім'я файлу для запису
filename_angles = 'angles.txt' # ім'я файлу для кутів

# Конвертація градусів у радіани
def deg_to_rad(deg):
    return deg * math.pi / 180

# Дані для запису
data = []
phi_theta = []
probe_number = 1

# Генерація точок
for phi_deg in range(5, 86, phi_step): #  $\phi$  від  $0^\circ$  до  $90^\circ$  з кроком  $5^\circ$ 
    for theta_deg in range(5, 86, theta_step): #  $\theta$  від  $0^\circ$  до  $90^\circ$  з кроком  $5^\circ$ 

        # Перетворення кутів у радіани
        phi = deg_to_rad(phi_deg)
        theta = deg_to_rad(theta_deg)

        # Перетворення у декартові координати
        x = radius * math.sin(theta) * math.cos(phi)
        y = radius * math.sin(theta) * math.sin(phi)
        z = radius * math.cos(theta) + z_level

        # Додавання точки до списків
        data.append([probe_number, round(x, 3), round(y, 3), round(z, 3), 'X'])
        phi_theta.append([probe_number, phi_deg, theta_deg])

        # print(phi_deg, theta_deg)

        probe_number += 1

# Запис даних у файл координат
```

```

with open(filename, 'w') as file:
    # Запис заголовків
    file.write('#Number\tX\tY\tZ\tOrientation\n')
    file.write('#===== \n')

    # Запис кожного рядка даних
    for row in data:
        file.write('\t'.join(map(str, row)) + '\n')

print(f'Дані записані у файл {filename}')

# Запис даних у файл кутів
with open(filename_angles, 'w') as file:
    # Запис заголовків
    file.write('#Number\tPhi (deg)\tTheta (deg)\n')
    file.write('#===== \n')

    # Запис кожного рядка даних
    for row in phi_theta:
        file.write('\t'.join(map(str, row)) + '\n')

print(f'Кути записані у файл {filename_angles}')

```

ДОДАТОК 2

' export field

Sub Main ()

 SelectTreeItem ("1D Results\Probes\E-Field\Probe Signals\")

 With ASCIIExport

 .Reset

 .FileName ("D:\Zadorozhnya\Klevin\0_25GHZ.txt")

 .Execute

 End With

End Sub

ДОДАТОК 3

```
import re
import json

def parse_efield_file(file_path):
    """Парсить файл із часовими залежностями для різних координат."""
    data = {}
    current_key = None
    current_values = []

    try:
        with open(file_path, 'r', encoding='utf-8') as file:
            for line in file:
                # Шукаємо рядок, що містить координати
                match = re.search(r"E-Field \(((^)+)\)", line)
                if match:
                    # Зберігаємо попередній набір даних
                    if current_key and current_values:
                        data[current_key] = current_values
                        current_values = []

                    # Витягуємо ключ
                    current_key = match.group(1)

                # Пропускаємо рядки з тире
                elif "-----" in line:
                    continue

                # Обробка числових значень
                elif current_key:
                    parts = line.split()
                    if len(parts) == 2:
                        try:
                            time_value = float(parts[0])
                            field_value = float(parts[1])
                            current_values.append((time_value, field_value))
                        except ValueError:
                            pass # Пропускаємо неправильні значення

            # Зберігаємо останній набір даних
            if current_key and current_values:
                data[current_key] = current_values
```

```

except FileNotFoundError:
    print(f"Файл не знайдено: {file_path}")
except Exception as e:
    print(f"Виникла помилка: {e}")

return data

def save_to_json(data, output_file):
    """Зберігає дані у файл JSON."""
    try:
        with open(output_file, 'w', encoding='utf-8') as file:
            json.dump(data, file, ensure_ascii=False, indent=4)
        print(f"Дані збережено у файл: {output_file}")
    except Exception as e:
        print(f"Помилка під час збереження: {e}")

def main():
    # Список файлів для обробки
    files = [
        "0_25GHZ.txt" # Додайте імена файлів тут
    ]

    for file_path in files:
        print(f"Обробка файлу: {file_path}")
        file_data = parse_efield_file(file_path)

        # Генерація імені вихідного JSON-файлу
        output_file = file_path.rsplit('.', 1)[0] + '.json'

        # Збереження результату у JSON
        save_to_json(file_data, output_file)

if __name__ == "__main__":
    main()

```

ДОДАТОК 4.

```
import json

def interpolate(number, n, h, x, y):
    """
    Інтерполяція для часових залежностей.

    :param number: Індекс точки, від якої почати інтерполяцію.
    :param n: Кількість інтерпольованих точок.
    :param h: Крок інтерполяції.
    :param x: Список значень часу.
    :param y: Список значень E-Field.
    :return: Масив із інтерпольованими значеннями [XX, YY].
    """
    xx = [x[number] + i * h for i in range(1, n + 1)]
    yy = []

    for xx_value in xx:
        ni = 0
        while ni < len(x) - 1 and xx_value > x[ni + 1]:
            ni += 1
        # Лінійна інтерполяція
        y_interp = y[ni] + (y[ni + 1] - y[ni]) * (xx_value - x[ni]) / (x[ni + 1] - x[ni])
        yy.append(y_interp)

    return xx, yy

def process_json_files(file_paths, number, n, h):
    """
    Обробляє файли JSON із часовими залежностями та виконує інтерполяцію.

    :param file_paths: Список файлів JSON для обробки.
    :param number: Індекс початкової точки для інтерполяції.
    :param n: Кількість точок для інтерполяції.
    :param h: Крок інтерполяції.
    """
    for file_path in file_paths:
        try:
            # Завантаження даних із JSON-файлу
            with open(file_path, 'r', encoding='utf-8') as file:
                data = json.load(file)
```

```

interpolated_data = {}

# Обробка кожного ключа (координати x, y, z)
for key, values in data.items():
    x = [point[0] for point in values]
    y = [point[1] for point in values]

    # Перевірка індексу
    if number >= len(x) - 1:
        print(f"Увага: Індекс {number} перевищує доступну кількість точок
у ключі '{key}'. Пропускаємо.")
        continue

    # Інтерполяція
    xx, yy = interpolate(number, n, h, x, y)
    interpolated_data[key] = list(zip(xx, yy))

# Збереження оброблених даних у новий JSON-файл
output_file = file_path.replace(".json", "_interpolated.json")
with open(output_file, 'w', encoding='utf-8') as output:
    json.dump(interpolated_data, output, ensure_ascii=False, indent=4)

print(f"Файл '{output_file}' успішно збережено.")

except FileNotFoundError:
    print(f"Файл не знайдено: {file_path}")
except Exception as e:
    print(f"Помилка під час обробки файлу {file_path}: {e}")

def main():
    # Список файлів JSON для обробки
    files = [
        "0_25GHZ.json" # Додайте ваші файли сюди
    ]

    # Параметри інтерполяції
    number = 0 # Введіть індекс точки, від якої почати інтерполяцію (number: ")
    n = 210 # Введіть кількість точок для інтерполяції (N: ")
    h = 0.02

    # Обробка файлів
    process_json_files(files, number, n, h)

```

```
if __name__ == "__main__":  
    main()
```

ДОДАТОК 5

Парсер даних для перевизначення декартових координат в сферичні координати

```
import json
import re
```

```
def load_table(file_path):
    """
    Завантажує таблицю з текстового файлу у вигляді списку словників.
    """
    table = []
    try:
        with open(file_path, 'r', encoding='utf-8') as file:
            for line in file:
                # Пропускаємо коментарі та пусті рядки
                if line.startswith("#") or line.strip() == "" or "=" in line:
                    continue

                # Розбиваємо рядок на колонки
                parts = line.split()

                # Спробуємо перетворити частини на числа або залишимо текст
                row = []
                for x in parts:
                    try:
                        # Обробка числа з можливим від'ємним знаком
                        value = float(x)
                    except ValueError:
                        value = x # Якщо це не число, зберігаємо як текст
                row.append(value)

                table.append(row)
    except FileNotFoundError:
        print(f'Файл не знайдено: {file_path}')
    except Exception as e:
        print(f'Помилка під час завантаження таблиці: {e}')
    return table

def process_json_file(json_file, coord_table, angle_table, output_file):
    """
    Обробляє JSON-файл: змінює значення 'X' на кути Phi_Theta.
    """
```

```

try:
    # Завантажуємо JSON-файл
    with open(json_file, 'r', encoding='utf-8') as file:
        data = json.load(file)

    updated_data = { }

    for key, values in data.items():
        # Парсимо координати з ключа
        match = re.search(r"Y;\s*([\d\.s]+)", key)
        if not match:
            print(f'Не вдалося знайти координати у ключі: {key}')
            continue

        coords = [float(coord) for coord in match.group(1).split()]

        # Знаходимо номер (Number) координати у таблиці
        number = None
        for row in coord_table:
            if coords == row[1:4]:
                number = int(row[0]) # Порядковий номер
                break

        if number is None:
            print(f'Координати {coords} не знайдено у таблиці координат.')
            continue

        # Знаходимо кути Phi та Theta у таблиці кутів
        phi_theta = None
        for row in angle_table:
            if int(row[0]) == number:
                phi_theta = f"{int(row[1])} {int(row[2])}"
                break

        if phi_theta is None:
            print(f'Номер {number} не знайдено у таблиці кутів.')
            continue

        # Замінюємо значення 'X' у ключі на кути
        updated_key = phi_theta
        updated_data[updated_key] = values

    # Зберігаємо оновлений JSON
    with open(output_file, 'w', encoding='utf-8') as file:

```

```

    json.dump(updated_data, file, ensure_ascii=False, indent=4)

print(f'Файл '{output_file}' успішно створено.")

except FileNotFoundError:
    print(f'Файл не знайдено: {json_file}')
except Exception as e:
    print(f'Помилка під час обробки файлу {json_file}: {e}')

def main():
    # Файли
    coord_file = "output_semi_sphere.txt"
    angle_file = "angles.txt"
    json_files = [
        "0_25GHZ_interpolated.json"
    ]

    # Завантажуємо таблиці
    coord_table = load_table(coord_file)

    angle_table = load_table(angle_file)

    # Обробляємо кожен JSON-файл
    for json_file in json_files:
        output_file = json_file.replace(".json", "_updated.json")
        process_json_file(json_file, coord_table, angle_table, output_file)

if __name__ == "__main__":
    main()

```

ДОДАТОК 6

Побудова 3D картинок поля

```
import json
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# -----
# 1. Імпорт JSON
# -----
file_path = "0_25GHZ_interpolated_updated.json"

with open(file_path, "r") as f:
    data = json.load(f)

# -----
# 2. Парсинг даних
# -----
entries = []

for key, arr in data.items():
    theta, phi = map(float, key.split())
    t = np.array([p[0] for p in arr])
    val = np.array([p[1] for p in arr])
    entries.append((theta, phi, t, val))

# -----
# 3. Побудова 3D-поверхні для фіксованого φ
# -----
def plot_surface_theta(phi_fixed):

    # фільтрація
    subset = [(theta, t, val) for (theta, phi, t, val) in entries if phi == phi_fixed]

    # сортуємо по θ щоб поверхня була гладка
    subset.sort(key=lambda x: x[0])

    thetas = np.array([s[0] for s in subset])
    times = subset[0][1] # вважаємо, що час однаковий у всіх
    values = np.array([s[2] for s in subset]) # матриця: θ × час

    # створюємо координатні сітки
    THETA, TIME = np.meshgrid(thetas, times, indexing='ij')
```

```

fig = plt.figure(figsize=(11, 7))
ax = fig.add_subplot(111, projection='3d')

# поверхня
surf = ax.plot_surface(TIME, THETA, values,
                       cmap="viridis",
                       linewidth=0,
                       antialiased=True,
                       alpha=0.9)

ax.set_xlabel("Час")
ax.set_ylabel("Кут  $\theta$  ( $^\circ$ )")
ax.set_zlabel("Амплітуда")
ax.set_title(f'3D поверхня: залежність від  $\theta$  при  $\varphi = \{\text{phi\_fixed}\}$ ')

fig.colorbar(surf, shrink=0.5, aspect=8)
plt.show()

# -----
# 4. Побудова 3D-поверхні для фіксованого  $\theta$ 
# -----
def plot_surface_phi(theta_fixed):

    subset = [(phi, t, val) for (theta, phi, t, val) in entries if theta == theta_fixed]
    subset.sort(key=lambda x: x[0])

    phis = np.array([s[0] for s in subset])
    times = subset[0][1]
    values = np.array([s[2] for s in subset])

    PHI, TIME = np.meshgrid(phis, times, indexing='ij')

    fig = plt.figure(figsize=(11, 7))
    ax = fig.add_subplot(111, projection='3d')

    surf = ax.plot_surface(TIME, PHI, values,
                           cmap="plasma",
                           linewidth=0,
                           antialiased=True,
                           alpha=0.9)

    ax.set_xlabel("Час")
    ax.set_ylabel("Кут  $\varphi$  ( $^\circ$ )")

```

```
ax.set_zlabel("Амплітуда")  
ax.set_title(f"3D поверхня: залежність від  $\varphi$  при  $\theta = \{theta\_fixed\}$ ")
```

```
fig.colorbar(surf, shrink=0.5, aspect=8)  
plt.show()
```

```
# -----  
# 5. Виклики  
# -----
```

```
plot_surface_theta(phi_fixed=85)  
plot_surface_phi(theta_fixed=85)
```