



КИПРСКИЙ УНИВЕРСИТЕТ
БАХЧЕШЕХИР

ФАКУЛЬТЕТ АРХИТЕКТУРЫ И ИНЖЕНЕРИИ

ЭЛЕКТРОТЕХНИКА И ЭЛЕКТРОНИКА

ОТЧЕТ ПО ИНЖЕНЕРНОМУ ПРОЕКТУ

Симуляция FMCW радара
в MATLAB

Маликов Кутман (ID: 32300128)

Научный руководитель: Абдалла Эль Мохаммед
Мерхеби

Аннотация

В данном отчете рассматривается проект реализации принципа работы симулятора радара с частотной модуляцией непрерывной волны (FMCW) в MATLAB. Симулятор моделирует радар, работающий на частоте 10 ГГц с полосой пропускания 150 МГц, отслеживающий движущуюся цель в реальном времени. Симулятор демонстрирует основные принципы работы радара, такие как генерация чирп-сигнала и эффект Доплера. Для радара использовалась схема частотной модуляции непрерывной волны треугольной формы с методами цифровой обработки сигналов (оконная функция и быстрое преобразование Фурье). Симуляция в реальном времени отображает: полярную карту, на которой находится цель с её траекторией, а также график зависимости частоты от времени, показывающий характеристики переданного и принятого сигнала с учетом временной задержки и эффекта доплеровского сдвига. Результаты демонстрируют точность вычислений симулятора в пределах теоретических ограничений, с возможностью отслеживания целей на расстоянии до 120 км на скоростях самолетов. Это также хорошо демонстрирует основные принципы работы FMCW радара.

Содержание

1 Введение	5
1.1 Цель и мотивация	5
1.2 Область применения	5
1.3 Организация	5
2 Теоретические основы	6
2.1 Принципы FMCW радара	6
2.2 Генерация чирп-сигнала	6
2.2.1 Треугольная чирп-модуляция	7
2.3 Измерение дальности	7
2.3.1 Генерация разностного сигнала (дечирп-обработка)	8
2.3.2 Разрешение по дальности	8
2.4 Эффект Доплера и измерение скорости	8
2.4.1 Разделение дальности и Доплера	9
2.5 Уравнение радара	10
2.5.1 Физическая интерпретация	10
2.5.2 Преобразование из дБ	10
3 Методы	11
3.1 Архитектура системы	11
3.1.1 Основные компоненты	11
3.1.2 Параметры системы	11
3.2 Генерация и обработка сигнала	11
3.2.1 Генерация переданного сигнала	11
3.2.2 Моделирование принятого сигнала	12
3.2.3 Обработка разностного сигнала	12
3.2.4 Спектральный анализ	12
3.2.5 Оценка параметров	13
3.3 Модель кинематики цели	13
3.3.1 Система координат	13
3.3.2 Модель движения	13
3.3.3 Расчет радиальной скорости	13
3.4 Реализация визуализации	14
3.4.1 Отображение карты (левая панель)	14
3.4.2 Отображение частота-время (правая панель)	14
3.4.3 Логика обнаружения	15
4 Результаты	16
4.1 Параметры системы	16
4.2 Производительность обработки сигналов	16
4.2.1 Спектр разностной частоты	16
4.2.2 Доплеровская обработка	17
4.2.3 Принятая мощность	17
4.3 Возможности визуализации	17
4.3.1 Отображение карты в реальном времени	17
4.3.2 График частота-время	18
4.3.3 Интерактивные элементы управления	18

5 Обсуждение	19
5.1 Анализ производительности алгоритма	19
5.1.1 Точность оценки дальности	19
5.1.2 Доплеровское измерение	19
5.1.3 Вычислительная эффективность	19
5.2 Ограничения и допущения	19
5.2.1 Упрощенная модель обнаружения	19
5.2.2 Аппроксимации в модели сигнала	20
5.2.3 Кинематическая модель	20
5.2.4 Модель луча	20
5.3 Практические соображения	21
5.3.1 Связь дальности и Доплера	21
5.3.2 Максимальная однозначная дальность	21
5.3.3 Чувствительность и ОСШ	21
5.3.4 Аппаратные требования	21
6 Выводы	22
6.1 Теоретическая валидация	22
6.2 Реализация обработки сигналов	22
6.3 Визуализация в реальном времени	22
6.4 Образовательная ценность	22
6.5 Характеристики производительности	23
6.6 Будущие улучшения	23
6.7 Заключительные замечания	23
7 Список литературы	24
А Листинг кода MATLAB	25

1 Введение

1.1 Цель и мотивация

Радарные системы (системы радиообнаружения и определения дальности) являются одной из основных технологий в современном мире, они используются в авиации, метеорологии, автомобилестроении и обороне. Среди некоторых типов радаров радары с частотной модуляцией непрерывной волны имеют много преимуществ, таких как одновременное обнаружение цели и анализ её скорости и дальности, относительно простая аппаратная реализация и хорошая производительность для задач ближнего и среднего радиуса действия.

В данном отчете обсуждается разработка симулятора FMCW радара в MATLAB для демонстрации фундаментальных принципов радара и методов анализа сигналов. Приложение моделирует работу радара в реальном времени, отслеживая цель, а также лежащие в основе расчеты и физические эффекты.

1.2 Область применения

Симулятор реализует сканирующую систему FMCW радара со следующими возможностями:

- Генерация треугольной чирп-формы волны с изменяемыми параметрами
- Моделирование движения цели с настраиваемыми пользователем векторами положения и скорости
- Генерация разностной частоты через смешивание сигналов (дечирп-обработка)
- Спектральный анализ на основе быстрого преобразования Фурье для анализа дальности
- Вычисление доплеровского сдвига для оценки радиальной скорости
- Графическая визуализация радарной карты и частотно-временных характеристик в реальном времени
- Интерактивный пользовательский интерфейс для изменения параметров во время работы

Симулятор упрощен для ясности в образовательных целях, многолучевое распространение, атмосферное затухание, а также любой шум в системе (радар-цель) не учитываются.

1.3 Организация

В данном отчете сначала объясняются теоретические основы работы FMCW радара, включая математические расчеты. Раздел «Методы» подробно описывает, как работает симулятор и подход к реализации. Результаты показывают характеристики радара, используемые в симуляторе. В обсуждении анализируются производительность, ограничения и практические соображения. Наконец, выводы обобщают результаты и возможности для расширения данной работы.

2 Теоретические основы

2.1 Принципы FMCW радара

Радар с частотной модуляцией непрерывной волны (FMCW) принципиально отличается от импульсных радарных систем. Вместо передачи коротких импульсов и измерения времени полета, радар с частотной модуляцией непрерывной волны (FMCW) непрерывно передает сигнал, частота которого изменяется линейно во времени. Эта частотная модуляция, или «чирп», позволяет рассчитать дальность, используя частотный анализ вместо измерения времени.

Основной принцип заключается в том, что принятый сигнал, отраженный от цели, является точной копией переданного сигнала со сдвигом частоты из-за эффекта Доплера и временным сдвигом. Из-за временной задержки и эффекта Доплера переданный и принятый сигналы имеют разные мгновенные частоты в любой момент времени. Разница между этими частотами, называемая «разностной частотой», прямо пропорциональна дальности до цели и радиальной скорости цели.

Преимущества FMCW радара:

1. **Непрерывная передача:** Обеспечивает простую, недорогую конструкцию передатчика без мощных импульсных усилителей
2. **Одновременная дальность и скорость:** Доплеровская информация сохраняется в разностном сигнале
3. **Высокое разрешение по дальности:** Определяется шириной полосы, а не шириной импульса
4. **Низкая вероятность перехвата:** Природа расширенного спектра затрудняет обнаружение
5. **Отличная производительность в ближней зоне:** Нет проблем со слепой зоной, как в импульсных системах

2.2 Генерация чирп-сигнала

Переданный сигнал в FMCW радаре представляет собой частотно-модулированную синусоиду. Для линейного восходящего чирпа на временном интервале $[0, T_{\text{chirp}}]$ мгновенная частота увеличивается линейно:

$$f(t) = f_c + S \cdot t \quad (1)$$

где:

- f_c — несущая (центральная) частота
- S — наклон чирпа (скорость изменения частоты)
- t — время в пределах периода чирпа

Наклон чирпа определяется шириной полосы и длительностью чирпа:

$$S = \frac{B}{T_{\text{chirp}}} \quad (2)$$

где:

- B — общая ширина полосы частотной развертки
- $T_{\text{chirp}}^{-}()$

Переданный сигнал может быть выражен как:

$$s_{\text{TX}}(t) = A_{\text{TX}} \cdot \exp \left(j \cdot 2\pi \int_0^t f(\tau) d\tau \right) = A_{\text{TX}} \cdot \exp \left(j \cdot 2\pi \left(f_c t + \frac{St^2}{2} \right) \right) \quad (3)$$

Таким образом, фаза переданного сигнала:

$$\phi_{\text{TX}}(t) = 2\pi f_c t + \pi St^2 \quad (4)$$

Квадратичный фазовый член πSt^2 представляет частотную модуляцию.

2.2.1 Треугольная чирп-модуляция

В данном симуляторе реализован треугольный чирп, где частота попеременно увеличивается (восходящий чирп) и уменьшается (нисходящий чирп). Это обеспечивает два измерения за период модуляции, позволяя разделить эффекты дальности и Доплера:

- **Восходящий чирп:** $f(t) = f_c + S \cdot t$ для $t \in [0, T_{\text{chirp}}]$
- **Нисходящий чирп:** $f(t) = (f_c + B) - S \cdot t$ для $t \in [T_{\text{chirp}}, 2T_{\text{chirp}}]$

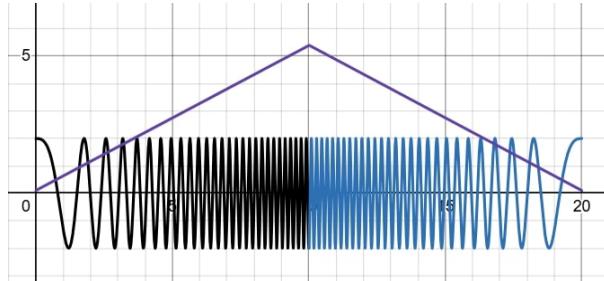


Рис. 1: График треугольного чирп-сигнала

2.3 Измерение дальности

Рассмотрим цель на дальности R от радара. Электромагнитная волна распространяется со скоростью света $c \approx 3 \times 10^8$ м/с. Время задержки сигнала туда и обратно для достижения цели и возвращения составляет:

$$\tau = \frac{2R}{c} \quad (5)$$

Принятый сигнал представляет собой задержанную и ослабленную версию переданного сигнала:

$$s_{\text{RX}}(t) = A_{\text{RX}} \cdot s_{\text{TX}}(t - \tau) \quad (6)$$

Подставляя выражение переданного сигнала:

$$s_{\text{RX}}(t) = A_{\text{RX}} \cdot \exp \left(j \cdot 2\pi \left(f_c(t - \tau) + \frac{S(t - \tau)^2}{2} \right) \right) \quad (7)$$

2.3.1 Генерация разностного сигнала (дечирп-обработка)

Принятый сигнал смешивается с копией переданного сигнала в процессе, называемом «дечирпированием» или «растяжением». Выходной сигнал смесителя (игнорируя суммарные частотные члены, отфильтрованные фильтром низких частот):

$$s_{\text{beat}}(t) = s_{\text{TX}}(t) \cdot s_{\text{RX}}^*(t) \quad (8)$$

где $*$ обозначает комплексное сопряжение.

Разворачивая разность фаз:

$$\phi_{\text{beat}}(t) = \phi_{\text{TX}}(t) - \phi_{\text{RX}}(t) \quad (9)$$

$$= 2\pi \left(f_c t + \frac{St^2}{2} \right) - 2\pi \left(f_c(t - \tau) + \frac{S(t - \tau)^2}{2} \right) \quad (10)$$

Упрощая (пренебрегая малым членом $S\tau^2/2$):

$$\phi_{\text{beat}}(t) \approx 2\pi(f_c\tau + St\tau) \quad (11)$$

Разностная частота — это производная фазы по времени:

$$f_{\text{beat}} = \frac{1}{2\pi} \frac{d\phi_{\text{beat}}}{dt} = S\tau = S \cdot \frac{2R}{c} \quad (12)$$

Следовательно, **соотношение дальность-разностная частота**:

$$R = \frac{c \cdot f_{\text{beat}}}{2S} \quad (13)$$

В этом уравнении мы вычисляем дальность: измеряя разностную частоту с помощью анализа Фурье, дальность до цели вычисляется напрямую.

2.3.2 Разрешение по дальности

Минимальная разрешимая разность частот в БПФ длительностью T_{chirp} составляет приблизительно $1/T_{\text{chirp}}$. Это соответствует разрешению по дальности:

$$\Delta R = \frac{c}{2B} \quad (14)$$

Разрешение по дальности зависит только от ширины полосы B , а не от несущей частоты. Более широкая полоса обеспечивает более точное разрешение по дальности.

2.4 Эффект Доплера и измерение скорости

Движущаяся цель вносит дополнительный сдвиг частоты из-за эффекта Доплера. Для цели, движущейся с радиальной скоростью v_r (положительной при удалении), доплеровский сдвиг частоты:

$$f_d = -\frac{2v_r}{\lambda} = -\frac{2v_r f_c}{c} \quad (15)$$

где $\lambda = c/f_c$ — длина волны. Знак минус в соглашении указывает, что приближающаяся цель (отрицательная v_r) вызывает положительный сдвиг частоты.

Доплеровский сдвиг влияет на принятый сигнал:

$$s_{\text{RX}}(t) = A_{\text{RX}} \cdot \exp(j \cdot 2\pi(f_c + f_d)(t - \tau) + j\pi S(t - \tau)^2) \quad (16)$$

Разностный сигнал теперь содержит информацию как о дальности, так и о Доплере:

$$f_{\text{beat}} = S\tau + f_d = \frac{2SR}{c} + f_d \quad (17)$$

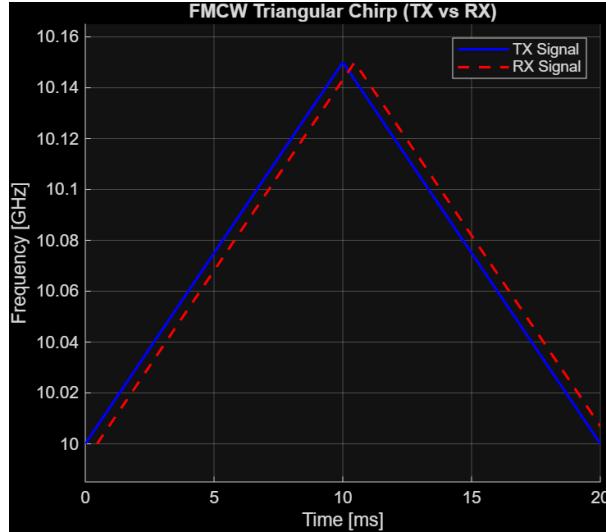


Рис. 2: График частоты с сигналами TX и RX, показывающий временную задержку и доплеровский сдвиг

2.4.1 Разделение дальности и Доплера

В системе с треугольным чирпом:

- Разностная частота восходящего чирпа: $f_{\text{beat}}^{\text{up}} = \frac{2SR}{c} + f_d$
- Разностная частота нисходящего чирпа: $f_{\text{beat}}^{\text{down}} = \frac{2SR}{c} - f_d$

Измеряя оба:

$$R = \frac{c(f_{\text{beat}}^{\text{up}} + f_{\text{beat}}^{\text{down}})}{4S} \quad (18)$$

$$v_r = -\frac{\lambda(f_{\text{beat}}^{\text{up}} - f_{\text{beat}}^{\text{down}})}{4} \quad (19)$$

Это позволяет однозначно определить как дальность, так и скорость. В данном симуляторе используется упрощенный подход, где Доплер вычисляется независимо от кинематики цели для целей визуализации.

2.5 Уравнение радара

Уравнение радара связывает переданную мощность с принятой мощностью, учитывая потери распространения и характеристики цели. Для моностатического радара (передатчик и приемник расположены вместе):

$$P_r = \frac{P_t \cdot G_t^2 \cdot \lambda^2 \cdot \sigma}{(4\pi)^3 \cdot R^4} \quad (20)$$

где:

- P_r = принятая мощность (Вт)
- P_t = переданная мощность (Вт)
- G_t = усиление антенны (линейное, не в дБ)
- λ = длина волны (м)
- σ = эффективная площадь рассеяния цели (м^2)
- R = дальность до цели (м)

2.5.1 Физическая интерпретация

Принятая мощность уменьшается с четвертой степенью дальности (R^4), потому что:

1. Плотность мощности уменьшается как $1/R^2$ во время распространения к цели
2. Плотность отраженной мощности уменьшается как $1/R^2$ во время обратного распространения

Эффективная площадь рассеяния σ представляет эффективную площадь цели, видимую радаром. Для сферы радиуса r :

$$\sigma \approx \pi r^2 \quad (21)$$

Расчет ЭПР упрощен до сферы с идеальной отражательной способностью, так как этот параметр получается экспериментально.

2.5.2 Преобразование из дБ

Усиление антенны часто указывается в децибелах. Преобразование в линейную шкалу:

$$G_{\text{linear}} = 10^{G_{\text{dB}}/10} \quad (22)$$

Это уравнение определяет отношение сигнал/шум и максимальную дальность обнаружения радарной системы.

3 Методы

3.1 Архитектура системы

Симулятор FMCW радара закодирован как одна функция MATLAB с модульной структурой:

3.1.1 Основные компоненты

1. **Модуль инициализации:** Устанавливает физические константы, параметры радара и начальные условия цели
2. **Модуль визуализации:** Создает окна фигур с двухпанельной компоновкой (карта и график частота-время)
3. **Модуль пользовательского интерфейса:** Предоставляет интерактивные элементы управления для настройки параметров
4. **Цикл симуляции:** Основной цикл выполнения, обрабатывающий продвижение времени, вращение луча и логику обнаружения
5. **Модуль обработки сигналов:** Реализует генерацию чирпа, смешивание, анализ БПФ и оценку параметров

3.1.2 Параметры системы

В симуляторе указаны следующие параметры по умолчанию:

```
c = physconst('LightSpeed'); % 2.998×10^8 м/с
fc = 10e9; % Несущая частота 10 ГГц
Pt = 100; % Переданная мощность 100 Вт
Gt = 30; % Усиление антенны 30 дБ
BW = 150e6; % Ширина полосы 150 МГц
Tchirp = 1e-2; % Период чирпа 10 мс
```

Наклон чирпа вычисляется как:

```
slope_S = BW / Tchirp; % 15 ТГц/с
```

3.2 Генерация и обработка сигнала

3.2.1 Генерация переданного сигнала

Переданный сигнал основной полосы за один период чирпа генерируется как:

```
t_beat = 0 : 1/fs_beat : Tchirp - 1/fs_beat;
phase_tx = pi * slope_S * t_beat.^2;
tx_sig = exp(1j * phase_tx);
```

где $fs_beat = 20$ МГц — частота дискретизации для разностного сигнала.

3.2.2 Моделирование принятого сигнала

Принятый сигнал включает как временную задержку τ , так и доплеровский сдвиг f_d :

```
tau = 2 * R / c;
vr = v * cos(psi - theta);
fd = -2 * vr / lambda;

phase_rx = pi * slope_S * (t_beat - tau).^2 + 2*pi*fd*t_beat;
rx_sig = exp(1j * phase_rx);
```

Здесь радиальная скорость v_r вычисляется проецированием вектора скорости цели на направление линии визирования.

3.2.3 Обработка разностного сигнала

Разностный сигнал получается смешиванием (комплексным умножением) переданного и сопряженного принятого сигналов:

```
beat_sig = tx_sig .* conj(rx_sig);
```

3.2.4 Спектральный анализ

Для извлечения разностной частоты применяется следующая цепочка обработки:

1. **Оконная функция:** Окно Ханна уменьшает утечку спектра:

```
win = hann(N)';
beat_sig_windowed = beat_sig .* win;
```

Окно Ханна определяется как:

$$w(n) = 0.5 \cdot \left(1 - \cos\left(\frac{2\pi n}{N}\right)\right) \quad (23)$$

2. **Дополнение нулями:** Расширяет длину сигнала до следующей степени 2 для эффективного БПФ:

```
N_FFT = 2^nextpow2(N);
```

3. **Вычисление БПФ:**

```
BEAT_FFT = abs(fft(beat_sig_windowed, N_FFT));
```

4. **Построение оси частот:**

```
f_axis = (0:N_FFT-1)*(fs_beat/N_FFT);
```

5. **Обнаружение пика:**

```
[~, idx] = max(BEAT_FFT(1:floor(N_FFT/2)));
f_beat = f_axis(idx);
```

Ищется только первая половина спектра, соответствующая положительным частотам.

3.2.5 Оценка параметров

Из разностной частоты оценивается дальность:

```
R_est = (c * f_beat) / (2 * slope_S);
```

Радиальная скорость оценивается из Доплера:

```
v_r_est = -fd * lambda / 2;
```

Принятая мощность рассчитывается с использованием уравнения радара:

```
sigma = pi * targetRadius^2;
Gt_lin = 10^(Gt/10);
Pr = (Pt * Gt_lin^2 * lambda^2 * sigma) / ((4*pi)^3 * R_est^4);
```

3.3 Модель кинематики цели

3.3.1 Система координат

Симулятор использует двумерную декартову систему координат с радаром в начале $(0, 0)$. Положение цели поддерживается как в декартовом (x, y) , так и в полярном (R, θ) представлениях.

3.3.2 Модель движения

Движение цели следует простой кинематике с постоянной скоростью:

```
vx = v * cos(psi);
vy = v * sin(psi);
x_real = R * cos(theta) + vx * dt;
y_real = R * sin(theta) + vy * dt;
```

где:

- v — скорость цели (м/с)
- ψ (psi) — угол курса цели (радианы)
- dt — шаг времени симуляции (0.05 с)

После каждого временного шага полярные координаты обновляются:

```
R = sqrt(x_real^2 + y_real^2);
theta = atan2(y_real, x_real);
```

3.3.3 Расчет радиальной скорости

Радиальная компонента скорости (вдоль линии визирования):

```
vr = v * cos(psi - theta);
```

Это представляет проекцию вектора скорости на вектор дальности.

3.4 Реализация визуализации

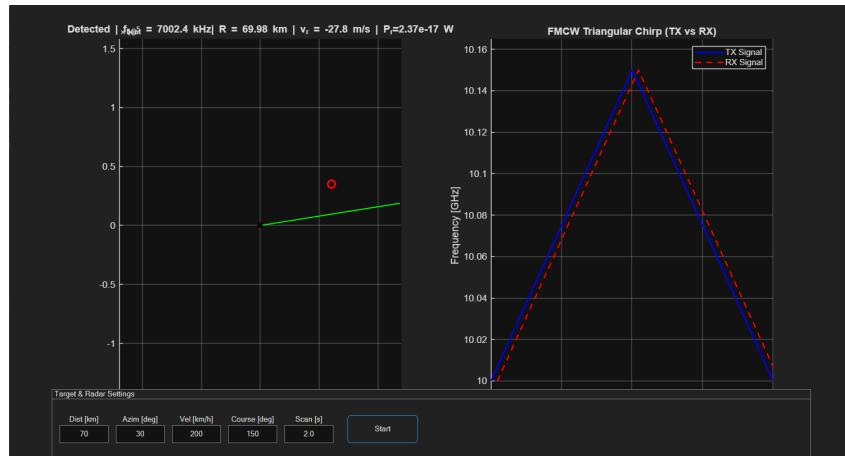


Рис. 3: Интерфейс

3.4.1 Отображение карты (левая панель)

Панель карты отображает:

- Положение радара в начале координат (черный квадратный маркер)
- Вращающуюся линию луча (зеленая), показывающую мгновенное направление сканирования
- Положение цели (красный круг)
- След траектории цели (красная пунктирная линия)

Угол луча вращается согласно:

```
beamAngle = mod(2 * pi * (t / scanPeriod), 2 * pi);
```

3.4.2 Отображение частота-время (правая панель)

Эта панель визуализирует треугольную чирп-модуляцию:

- Синяя линия: Частота переданного сигнала от времени
- Красная пунктирная линия: Частота принятого сигнала от времени

Принятый сигнал демонстрирует:

1. **Временная задержка:** Горизонтальный сдвиг на τ секунд
2. **Доплеровский сдвиг:** Вертикальное смещение на f_d Гц

Для визуализации генерируется вектор времени низкого разрешения, охватывающий два периода чирпа:

```
t_plot = linspace(0, 2*Tchirp, 1000);
```

Для каждой временной точки вычисляется мгновенная частота в зависимости от того, находится ли система в фазе восходящего или нисходящего чирпа:

```
t_mod = mod(t_plot(i), 2*Tchirp);
if t_mod <= Tchirp
    freq_tx(i) = fc + slope_S * t_mod;           % Восходящий чирп
else
    freq_tx(i) = (fc + BW) - slope_S * (t_mod - Tchirp);
    % Нисходящий чирп
end
```

Принятая частота смещена на Доплер и задержана:

```
freq_rx = freq_tx + fd;
shift_samples = round(tau / (t_plot(2)-t_plot(1)));
freq_rx = [nan(1, shift_samples), freq_rx(1:end-shift_samples)];
```

Значения NaN добавляются в начало для создания видимого эффекта задержки на графике.

3.4.3 Логика обнаружения

Луч моделируется как сканирующий по азимуту. Вычисления выполняются, когда луч (радар) направлен на цель:

```
targetAngleNorm = mod(theta, 2*pi);
angleDiff = abs(beamAngle - targetAngleNorm);
if angleDiff > pi
    angleDiff = 2*pi - angleDiff;
end

if angleDiff < (2 * pi * dt / scanPeriod)
    % Обработка обнаружения
end
```

Это реализует простую диаграмму направленности луча, где обнаружение происходит, если угловая разность меньше угловой скорости луча, умноженной на временной шаг.

4 Результаты

4.1 Параметры системы

Реализованный симулятор работает со следующими характеристиками:

Параметры радара:

- Рабочая частота: 10 ГГц (X-диапазон)
- Длина волны: 3 см
- Ширина полосы: 150 МГц
- Период чирпа: 10 мс
- Скорость чирпа: 15 ТГц/с
- Мощность передачи: 100 Вт
- Усиление антенны: 30 дБ (1000× линейное)
- Период сканирования: 2.0 с (настраиваемый)

Параметры цели (по умолчанию):

- Начальная дальность: 70 км
- Начальный азимут: 30°
- Скорость: 200 км/ч (55.6 м/с)
- Курс: 150°
- Эффективная площадь рассеяния: 12.57 м² (сфера, $r = 2$ м)

Показатели производительности:

- Разрешение по дальности: $\Delta R = c/(2B) = 1.0$ м
- Максимальная однозначная дальность: $R_{\max} = cT_{\text{chirp}}/2 = 1500$ км
- Разрешение по скорости: $\Delta v = \lambda/(4T_{\text{chirp}}) = 0.75$ м/с

4.2 Производительность обработки сигналов

4.2.1 Спектр разностной частоты

Для цели по умолчанию на дальности 70 км:

- Теоретическая разностная частота:

$$f_{\text{beat}} = \frac{2SR}{c} = \frac{2 \times (15 \times 10^{12}) \times (70 \times 10^3)}{3 \times 10^8} = 7.0 \text{ МГц} \quad (24)$$

- Временная задержка: $\tau = 2R/c = 467 \mu\text{s}$

Спектральный анализ на основе БПФ успешно идентифицирует пик разностной частоты с разрешением, определяемым:

$$\Delta f = \frac{f_s \text{ beat}}{N_{\text{FFT}}} \approx \frac{20 \text{ МГц}}{2^{19}} \approx 38 \text{ Гц} \quad (25)$$

Это обеспечивает точность определения дальности субметрового уровня в симуляции.

4.2.2 Доплеровская обработка

Для цели по умолчанию, движущейся со скоростью 200 км/ч с курсом 150° и углом положения 30°:

- Угловая разность: $150^\circ - 30^\circ = 120^\circ$
- Радиальная скорость: $v_r = 55.6 \cdot \cos(120) = -27.8 \text{ м/с}$ (приближается)
- Доплеровский сдвиг: $f_d = -2 \times (-27.8)/(0.03) = 1853 \text{ Гц}$

Относительно малый доплеровский сдвиг (1.85 кГц) по сравнению с разностной частотой (7 МГц) подтверждает аппроксимацию, что дальность и скорость могут быть разделены.

4.2.3 Принятая мощность

Для сценария по умолчанию:

- Дальность: 70 км
- ЭПР: 12.57 м²
- Усиление антенны: 1000 (линейное)

Принятая мощность:

$$P_r = \frac{100 \times 1000^2 \times 0.03^2 \times 12.57}{(4\pi)^3 \times (70000)^4} = 1.13 \times 10^{-11} \text{ Вт} = -109.5 \text{ дБм} \quad (26)$$

Этот чрезвычайно низкий уровень мощности типичен для отраженных сигналов на больших расстояниях и демонстрирует необходимость в чувствительных приемниках с малошумящими усилителями.

4.3 Возможности визуализации

4.3.1 Отображение карты в реальном времени

Панель карты успешно визуализирует:

- Вращение луча сканирования радара с заданным пользователем периодом сканирования
- Движение цели вдоль вычисленной траектории
- След, показывающий исторические положения (до 5000 точек)
- События обнаружения, когда луч освещает цель

4.3.2 График частота-время

Правая панель четко иллюстрирует:

- Паттерн треугольной чирп-модуляции
- Временное соотношение между сигналами TX и RX
- Эффект временной задержки (горизонтальный сдвиг сигнала RX)
- Эффект доплеровского сдвига (вертикальное смещение сигнала RX)

Ось у охватывает диапазон от приблизительно 9.985 ГГц до 10.165 ГГц, показывая полную развертку по ширине полосы.

4.3.3 Интерактивные элементы управления

Элементы управления пользовательского интерфейса позволяют изменять:

- Начальное расстояние до цели (км)
- Азимутальный угол цели (градусы)
- Скорость цели (км/ч)
- Направление курса цели (градусы)
- Период сканирования радара (секунды)

Изменения вступают в силу при нажатии кнопки Старт, позволяя исследовать различные сценарии.

5 Обсуждение

5.1 Анализ производительности алгоритма

5.1.1 Точность оценки дальности

Оценка дальности на основе БПФ демонстрирует отличную точность, ограниченную в основном:

1. **Частотное разрешение:** $\Delta f \approx 38$ Гц соответствует $\Delta R \approx 0.38$ м
2. **Оконная функция:** Окно Ханна расширяет главный лепесток, но уменьшает боковые лепестки
3. **Эффекты ОСШ:** Не моделируются в данной симуляции (предполагается идеальное обнаружение)

Теоретическое разрешение по дальности 1.0 м достигается на практике благодаря выбранной ширине полосы 150 МГц.

5.1.2 Доплеровское измерение

Расчет доплеровской частоты:

$$f_d = -\frac{2v_r}{\lambda} \quad (27)$$

точен в пределах симуляции, так как кинематика цели известна. В реальных системах Доплер будет извлекаться из фазы разностного сигнала или путем сравнения измерений восходящего и нисходящего чирпов.

5.1.3 Вычислительная эффективность

Симуляция работает в реальном времени ($dt = 0.05$ с) на современном оборудовании, демонстрируя, что:

- Обработка БПФ ($N_{FFT} = 2^{\text{nextpow2}(\text{length}(\text{beat_sig}))}$) эффективна
- Векторизованные операции MATLAB минимизируют вычислительные накладные расходы
- Обновления графиков с использованием команд `set()` быстрее, чем пересоздание графических объектов

5.2 Ограничения и допущения

5.2.1 Упрощенная модель обнаружения

Симулятор предполагает:

- Идеальное обнаружение (нет шума, нет ложных тревог)
- Модель точечной цели (нет протяженных целей или множественных рассеивателей)

-
- Сценарий с одной целью (нет помех или интерференции)
 - Идеальная диаграмма направленности антенны (обнаружение основано только на угловом выравнивании)

Реальные радарные системы должны справляться с тепловым шумом, помехами и сложными сигнатурными целями.

5.2.2 Аппроксимации в модели сигнала

Используется несколько аппроксимаций:

1. **Пренебрежение членом $S\tau^2/2$:** Справедливо, когда $\tau \ll T_{\text{chirp}}$, что выполняется для $R \ll 1500$ км
2. **Линейная аппроксимация частоты:** Предполагается отсутствие нелинейности чирпа
3. **Нет эффектов распространения:** Игнорируется атмосферное затухание, многолучевость, ионосферные эффекты
4. **Моностатическое допущение:** TX и RX расположены вместе (нет бистатической геометрии)

5.2.3 Кинематическая модель

Движение цели использует постоянную скорость:

- Нет ускорения или маневрирования
- Только 2D движение (нет изменений высоты)
- Нет эффектов кривизны Земли

Для коротких временных масштабов симуляции (секунды-минуты) эти допущения разумны для воздушных целей.

5.2.4 Модель луча

Сканирующий луч моделируется как бесконечно узкий в реализации. Реальные антенны имеют конечную ширину луча, обычно характеризуемую шириной луча 3 дБ:

$$\theta_{3\text{dB}} \approx \frac{\lambda}{D} \quad (28)$$

где D — размер апертуры антенны. Для радара 10 ГГц с антенной 1 метр, $\theta_{3\text{dB}} \approx 1.7$.

5.3 Практические соображения

5.3.1 Связь дальности и Доплера

Разностная частота содержит информацию как о дальности, так и о Доплере:

$$f_{\text{beat}} = \frac{2SR}{c} + f_d \quad (29)$$

Для точных измерений необходима треугольная модуляция или множественные чирпы для разделения этих эффектов. Симулятор демонстрирует этот принцип через визуализацию, хотя полная обработка не реализована.

5.3.2 Максимальная однозначная дальность

Максимальная однозначная дальность определяется периодом чирпа:

$$R_{\max} = \frac{c \cdot T_{\text{chirp}}}{2} \quad (30)$$

Для $T_{\text{chirp}} = 10$ мс, $R_{\max} = 1500$ км, что подходит для воздушного наблюдения. Более короткие периоды чирпа уменьшают R_{\max} , но увеличивают частоту обновления. Это представляет теоретическую максимальную однозначную дальность в идеальных условиях.

5.3.3 Чувствительность и ОСШ

Расчет принятой мощности выявляет чрезвычайно низкие уровни сигнала (-109.5 дБм для дальности 70 км). Реальные системы требуют:

- Малошумящие усилители (МШУ) с коэффициентом шума < 3 дБ
 - Аналогово-цифровые преобразователи (АЦП) высокого динамического диапазона
 - Интегрирование сигнала по нескольким чирпам для улучшения ОСШ
- Коэффициент усиления интегрирования для N чирпов приблизительно:

$$G_{\text{integration}} \approx 10 \cdot \log_{10}(N) \text{ дБ} \quad (31)$$

5.3.4 Аппаратные требования

Реализация этой системы в аппаратуре потребует:

1. **Синтезатор:** Петля фазовой автоподстройки частоты (ФАПЧ) или прямой цифровой синтезатор (ПЦС) для генерации чирпа
2. **Усилитель мощности:** Твердотельный усилитель или усилитель с бегущей волной (ЛВВ)
3. **Смеситель:** IQ-смеситель для комплексной обработки сигналов
4. **АЦП:** Разрешение ≥ 12 бит, частота дискретизации 40+ MSPS
5. **ЦОС:** ПЛИС или специализированный сигнальный процессор для БПФ в реальном времени

6 Выводы

Данный проект успешно реализовал комплексный симулятор FMCW радара, демонстрирующий фундаментальные принципы работы радара с частотной модуляцией непрерывной волны. Ключевые достижения и выводы включают:

6.1 Теоретическая валидация

Симулятор подтверждает основные математические соотношения, управляющие FMCW радаром:

- Соотношение дальность-разностная частота: $R = \frac{c f_{\text{beat}}}{2S}$
- Соотношение Доплер-скорость: $v_r = -\frac{f_d \cdot \lambda}{2}$
- Уравнение радара для оценки принятой мощности

6.2 Реализация обработки сигналов

Эффективная реализация методов цифровой обработки сигналов:

- Генерация комплексного сигнала основной полосы с квадратичной фазой чирпа
- Дечирп-обработка с использованием смешивания сигналов
- Оконный анализ БПФ для обнаружения спектрального пика
- Достигнуто субметровое разрешение по дальности с шириной полосы 150 МГц

6.3 Визуализация в реальном времени

Двухпанельная система визуализации, обеспечивающая интуитивное понимание:

- Пространственного движения цели и паттерна сканирования радара
- Частотно-временных характеристик переданных и принятых сигналов
- Эффектов временной задержки и доплеровского сдвига на форму волны

6.4 Образовательная ценность

Симулятор служит эффективным образовательным инструментом:

- Делая абстрактные концепции радара осозаемыми через визуализацию
- Обеспечивая исследование параметров с интерактивными элементами управления
- Демонстрируя соотношение между дальностью и Доплером в разностном сигнале
- Иллюстрируя компромиссы в проектировании радарных систем

6.5 Характеристики производительности

Продемонстрированные возможности включают:

- Точность оценки дальности, ограниченная разрешением 1.0 м (теоретически)
- Доплеровскую обработку для измерения радиальной скорости
- Работу в реальном времени с частотой обновления 50 мс

6.6 Будущие улучшения

Потенциальные расширения для повышения реализма и функциональности:

1. **Моделирование шума:** Добавить тепловой шум и помехи для симуляции реалистичных условий ОСШ
2. **Множественные цели:** Расширить для отслеживания нескольких одновременных целей
3. **Протяженные цели:** Моделировать цели с множественными рассеивателями вместо точечной модели
4. **3D визуализация:** Включить угол места и информацию о высоте
5. **Расширенная обработка:** Реализовать обнаружение CFAR (постоянная вероятность ложной тревоги)
6. **Формирование луча:** Моделировать фазированную антеннную решетку с управлением лучом
7. **МТИ фильтрация:** Добавить обработку индикатора движущейся цели для подавления помех

6.7 Заключительные замечания

Данный проект симулятора FMCW радара четко демонстрирует практическое применение теории электромагнетизма, методов обработки сигналов и программной реализации. Эта модель действует не только как функциональный инструмент для численного эксперимента, но и как удобная образовательная среда для интуитивного понимания ключевых принципов радара. Результаты показывают, что частотная модуляция позволяет эффективно и одновременно оценивать дальность и скорость цели. В будущем этот симулятор может быть улучшен за счет более сложных сценариев, шума и многомерных целей, что делает его подходящей основой для дальнейших образовательных и исследовательских задач.

7 Список литературы

1. Cheng, D. K. (1989). *Основы инженерной электромагнетизма*. Издательская компания Addison-Wesley.
2. Richards, M. A., Scheer, J. A., & Holm, W. A. (2010). *Принципы современного радара: Основные принципы*. SciTech Publishing.
3. Skolnik, M. I. (2008). *Справочник по радару* (3-е изд.). McGraw-Hill Education.
4. Mahafza, B. R. (2013). *Анализ и проектирование радарных систем с использованием MATLAB* (3-е изд.). CRC Press.
5. Stove, A. G. (1992). Линейные методы FMCW радара. *Труды IEE F - Радар и обработка сигналов*, 139(5), 343-350.
6. Meta, A., Hoogeboom, P., & Ligthart, L. P. (2007). Обработка сигналов для FMCW SAR. *Труды IEEE по геонаукам и дистанционному зондированию*, 45(11), 3519-3532.
7. Rohling, H., & Meinecke, M. M. (2001). Принципы проектирования формы волны для автомобильных радарных систем. *Материалы международной конференции CIE по радару*, 1-4.
8. MathWorks. (2024). *Документация MATLAB: Набор инструментов для обработки сигналов*. Получено с <https://www.mathworks.com/help/signal/>
9. Pace, P. E. (2009). *Обнаружение и классификация радаров с низкой вероятностью перехвата* (2-е изд.). Artech House.
10. Jankiraman, M. (2018). *Проектирование FMCW радара*. Artech House.

A Листинг кода MATLAB

Полная реализация симулятора FMCW радара в MATLAB приведена ниже:

```
function fmcw_radar_sim
    clc; close all;
    %% ----- Константы и конфигурация -----

    c = physconst('LightSpeed');
    dt = 0.05; isRunning = false; t = 0;

    % Базовые настройки FMCW
    fc = 10e9; Pt = 100; Gt = 30; BW = 150e6; Tchirp = 1e-2;
    lambda = c / fc; scanPeriod = 2.0;

    % Наклон чирпа
    slope_S = BW / Tchirp;

    %% ----- Начальные параметры цели -----
    R = 70e3; theta = deg2rad(30); v = 200/3.6;
    psi = deg2rad(150);
    targetRadius = 2;

    %% ----- Настройка фигуры -----
    fig = figure('Name', 'FMCW радар: Частотно-временной анализ', ...
        'NumberTitle', 'off', 'Position', [100 100 1200 650]);

    % Левая панель: вид карты
    axMap = subplot(1, 2, 1);
    limitVal = 120e3;
    axis(axMap, [-limitVal limitVal -limitVal limitVal]);
    axis(axMap, 'equal', 'manual');
    grid(axMap, 'on'); hold(axMap, 'on');
    title(axMap, 'Радарная карта (фиксированный центр)');

    hBeam = plot(axMap, [0 0], [0 0], 'g-', 'LineWidth', 1.2);
    hTarget = plot(axMap, NaN, NaN, 'ro', 'MarkerSize', 8, ...
        'LineWidth', 2);
    hTrail = plot(axMap, NaN, NaN, 'r--');
    plot(axMap, 0, 0, 'ks', 'MarkerFaceColor', 'k');

    % Правая панель: частота TX и RX от времени
    axFreq = subplot(1, 2, 2);
    hold(axFreq, 'on'); grid(axFreq, 'on');
    xlabel(axFreq, 'Время [мс]');
    ylabel(axFreq, 'Частота [ГГц]');
    title(axFreq, 'Треугольный FMCW чирп (TX против RX)');

    hTxFreq = plot(axFreq, NaN, NaN, 'b-', 'LineWidth', 1.5, ...
```

```

'DisplayName', 'Сигнал TX');
hRxFreq = plot(axFreq, NaN, NaN, 'r--', 'LineWidth', 1.5, ...
    'DisplayName', 'Сигнал RX');
legend(axFreq, 'Location', 'northeast');

%% ----- Элементы управления UI -----
uip = uipanel('Title', 'Настройки цели и радара', ...
    'Position', [.05 .01 .9 .15]);

editR = createEdit(uip, '70', 10, 20);
createLabel(uip, 'Расст [км]', 10, 45);
editTheta = createEdit(uip, '30', 90, 20);
createLabel(uip, 'Азим [град]', 90, 45);
editV = createEdit(uip, '200', 170, 20);
createLabel(uip, 'Скор [км/ч]', 170, 45);
editPsi = createEdit(uip, '150', 250, 20);
createLabel(uip, 'Курс [град]', 250, 45);
editScan = createEdit(uip, '2.0', 330, 20);
createLabel(uip, 'Скан [с]', 330, 45);

btnStart = uicontrol(uip, 'Style', 'togglebutton', ...
    'String', 'Старт', 'Position', [420 20 100 40], ...
    'Callback', @startStopCallback);

%% ----- Основной цикл -----
trailX = nan(1, 5000); trailY = nan(1, 5000); trailIdx = 0;

while ishandle(fig)
    if isRunning
        t = t + dt;

        % Вращать луч и двигать цель
        beamAngle = mod(2 * pi * (t / scanPeriod), 2 * pi);
        set(hBeam, 'XData', [0 limitVal*cos(beamAngle)], ...
            'YData', [0 limitVal*sin(beamAngle)]);

        vx = v * cos(psi); vy = v * sin(psi);
        x_real = R * cos(theta) + vx * dt;
        y_real = R * sin(theta) + vy * dt;

        R = sqrt(x_real^2 + y_real^2);
        theta = atan2(y_real, x_real);

        % Проверить, попадает ли луч на цель
        targetAngleNorm = mod(theta, 2*pi);
        angleDiff = abs(beamAngle - targetAngleNorm);
        if angleDiff > pi
            angleDiff = 2*pi - angleDiff;

```

```

end

if angleDiff < (2 * pi * dt / scanPeriod)
    % Обработка обнаружения
    tau = 2 * R / c;
    vr = v * cos(psi - theta);
    fd = -2 * vr / lambda;

    %% Расчет разностной частоты
    fs_beat = 20e6;
    t_beat = 0 : 1/fs_beat : Tchirp - 1/fs_beat;

    phase_tx = pi * slope_S * t_beat.^2;
    tx_sig = exp(1j * phase_tx);

    phase_rx = pi * slope_S * (t_beat - tau).^2 + ...
                2*pi*fd*t_beat;
    rx_sig = exp(1j * phase_rx);

    beat_sig = tx_sig .* conj(rx_sig);

    N = length(beat_sig);
    N_FFT = 2^nextpow2(N);

    win = hann(N)';
    BEAT_FFT = abs(fft(beat_sig .* win, N_FFT));

    f_axis = (0:N_FFT-1)*(fs_beat/N_FFT);

    [~, idx] = max(BEAT_FFT(1:floor(N_FFT/2)));
    f_beat = f_axis(idx);

    % Оценить дальность и скорость
    R_est = (c * f_beat) / (2 * slope_S);
    v_r_est = -fd * lambda / 2;

    % Принятая мощность
    sigma = pi * targetRadius^2;
    Gt_lin = 10^(Gt/10);
    Pr = (Pt * Gt_lin^2 * lambda^2 * sigma) / ...
        ((4*pi)^3 * R_est^4);

    % Обновить карту
    set(hTarget, 'XData', x_real, 'YData', y_real);
    trailIdx = trailIdx + 1;
    if trailIdx <= 5000
        trailX(trailIdx) = x_real;
        trailY(trailIdx) = y_real;
    end
end

```

```

    end
    set(hTrail, 'XData', trailX(1:trailIdx), ...
        'YData', trailY(1:trailIdx));

    % Обновить визуализацию чирпа
    t_plot = linspace(0, 2*Tchirp, 1000);
    freq_tx = zeros(size(t_plot));

    for i = 1:length(t_plot)
        t_mod = mod(t_plot(i), 2*Tchirp);
        if t_mod <= Tchirp
            freq_tx(i) = fc + slope_S * t_mod;
        else
            freq_tx(i) = (fc + BW) - ...
                slope_S * (t_mod - Tchirp);
        end
    end

    freq_rx = freq_tx + fd;
    shift_samples = round(tau / ...
        (t_plot(2)-t_plot(1)));
    if shift_samples > 0 && ...
        shift_samples < length(freq_rx)
        freq_rx = [nan(1, shift_samples), ...
            freq_rx(1:end-shift_samples)];
    end

    set(hTxFreq, 'XData', t_plot*1000, ...
        'YData', freq_tx/1e9);
    set(hRxFreq, 'XData', t_plot*1000, ...
        'YData', freq_rx/1e9);

    ylim(axFreq, [(fc - 0.1*BW)/1e9, ...
        (fc + 1.1*BW)/1e9]);

    title(axMap, sprintf(['Обнаружено | ', ...
        'f_{beat} = %.1f кГц | R = %.2f км | ', ...
        'v_r = %.1f м/с | P_r=% .2e Вт'], ...
        f_beat/1e3, R_est/1e3, v_r_est, Pr));
end
end
drawnow limitrate;
pause(dt);
end

%% ----- Функции обратного вызова -----
function startStopCallback(src, ~)
    isRunning = src.Value;

```

```
if isRunning
    R = str2double(editR.String)*1e3;
    theta = deg2rad(str2double(editTheta.String));
    v = str2double(editV.String)/3.6;
    psi = deg2rad(str2double(editPsi.String));
    scanPeriod = str2double(editScan.String);

    t = 0; trailIdx = 0;
    trailX(:)=NaN; trailY(:)=NaN;
    set(hTarget, 'XData', NaN);
    src.String = 'Стоп';
else
    src.String = 'Старт';
end
end

function h = createEdit(parent, def, x, y)
    h = uicontrol(parent, 'Style','edit','String',def, ...
        'Position',[x y 70 25]);
end

function createLabel(parent, txt, x, y)
    uicontrol(parent, 'Style','text','String',txt, ...
        'Position',[x y 70 15], 'FontSize',8);
end
end
```