

Технологии беспроводной связи

2022/23 учебный год

Инженерный тур

Общая информация

Профиль напрямую связан с потребностью современного мира в надежной и производительной связи в условиях постоянного роста числа подключаемых устройств и объемов передаваемого трафика. Ключевыми перспективными областями развития технологий связи являются космическая сфера, промышленность, подводная и мобильная робототехника. Задачи практического тура профиля перекликаются с актуальными задачами систем связи от работы с различными форматами данных и организации помехоустойчивого кодирования для передачи информации в условиях шумов, до разработки адаптивных систем слежения и протоколов связи. Команды работают со слежением за подвижным объектом и источником сигнала, кодированием и декодированием сигналов, изучают характер шумов, а также создают собственные протоколы передачи данных с учетом определенных требований к надежности, характеру помех в канале, скорости передачи, взаимосвязи данных в многоканальных системах.

Легенда задачи

Навык создания собственных протоколов — базовый в технологиях связи, огромном растущем рынке в мире с увеличивающимся спросом на специалистов. Дальнейшее развитие беспроводных технологий связи — это не только технологии связи для спутников или подводных аппаратов, но и для домашних устройств, систем сенсоров, сельскохозяйственных дронов и других объектов Интернета вещей — все они требуют создание протоколов связи. В настоящее время количество различных протоколов связи, количество систем, где их важно проектировать непрерывно растет, но понимание базовых принципов и способность конструировать такие каналы падает, в связи с ростом сложности протоколов. Поэтому важно работать с одной стороны с максимально современной постановкой задачи, а с другой с максимальной прозрачностью и возможностью ее увидеть и «пощупать».

Каждая команда будет работать со стендами программно-аппаратного комплекса «Беспроводные технологии связи». Стенд «Узконаправленные низкоэнергетические каналы связи» позволяет моделировать реальную ситуацию для систем связи нового типа между автономными роевыми объектами. Задачи на стенде посвящены установлению слежения движущихся объектов друг за другом и адаптивной дешифрации в реальном времени. Команды создают свои протоколы передачи данных под конкретные условия шумов, требования к надежности, характеру помех в канале, скорости передачи. Создание собственных систем кодирования и защиты информации, работа с помехозащищенными кодами — основа задач на образовательном стенде «Оптомеханическая визуализация кодирования сигналов». Цель команд — получить протокол передачи данных совершеннее, чем протокол команд соперников. Побеждает команда, решившая наилучшим образом задачи по точной передаче сообщений в ежегодно изменяющихся условиях.

Участники практического тура создают:

- программный код, позволяющий реализовать слежение за подвижным объектом и источником сигнала,
- собственный протокол передачи данных — с созданием программного кода по кодированию и декодированию сигналов — с учетом определенных требований к надежности, характеру помех в канале, скорости передачи, взаимосвязи данных в многоканальных системах.
- собственный код, и дешифратор для него для точной передачи сообщений определенного формата.

Требования к команде и компетенциям участников

Количество участников в команде: 4–5.

Компетенции, которыми должны обладать члены команды: (Роли, которые должны быть представлены в команде)

1. Программист — программирование адаптивной системы слежения, работа с протоколами связи.
2. Программист-исследователь — анализ характеристик шума в каналах, выбор схемы помехоустойчивого кодирования, помехоустойчивое кодирование для оптического канала.
3. Аналитик-исследователь — работа с данными и основами корреляционного анализа, работа с механикой стенда, расшифровка кодов и характеристик стенда.
4. Программист — умение обработки сигналов, работа с данными и владение основами корреляционного анализа.
5. Капитан — сбор и анализ решений подзадач от всех специалистов, описанных выше, обсуждение и коррекция полученных результатов для более эффективного решения общей задачи, организация командной работы. Капитан должен обладать знаниями по всем основным темам, уметь разбираться в коде.

Совмещение ролей возможно, не рекомендуется совмещать роль капитана, с ролью программиста, так как грамотные действия капитана иногда являются критичными и решающими, а находясь одновременно в позиции программиста, очень сложно отвлечься от текущей реализации и принять необходимое решение.

Оборудование и программное обеспечение

На практическом туре команды профиля «Технологии беспроводной связи» работают с экспериментальными стендами «Узконаправленные низкоэнергетические каналы связи» (УНКС) и «Оптомеханическая визуализация кодирования сигналов» (ОВКС), входящими в образовательный комплекс «Беспроводные технологии связи», разработанный компанией ИнСитиЛаб при финансовой поддержке Фонда содействия инновациям. Стенды созданы в логике моделирования реальных инженерных задач по созданию протоколов связи. При создании стендов разработчиками и экспертами выступали специалисты ИСЗФ СО РАН с реальным опытом работы проектирования систем слежения — современных радаров. Общий вид комплекса, представлен на рисунке [VI.2.1](#). Подробное описание комплекса приведено в Прило-

жении 1.



Рис. VI.2.1. Общий вид комплекса «Беспроводные технологии связи»

Каждая команда тренировочно работала со своим комплексом. Проверка решений команд производилась на судейских контрольных стендах (одна задача — один стенд), для обеспечения равенства условий всех команд.

Работа идет в программной среде, разработанной для комплекса «Беспроводные технологии связи», для работы могут быть использованы только библиотеки, предустановленные в среде.

Для анализа данных на внешних ноутбуках участники могут использовать то ПО, которое считают необходимым. Команда может использовать для работы личные ноутбуки, при их отсутствии команда может запросить у организаторов один ноутбук.

Наименование	Количество	Описание
Стенд «Узконаправленные низкоэнергетические каналы связи» (УНКС) в судейской версии	3	Стенды для сдачи задач
Стенд «Узконаправленные низкоэнергетические каналы связи» (УНКС) в судейской версии	1	Дополнительный стенд, обеспечивающий возможность тестирования своих решений на судейском варианте стенда
Стенд «Оптомеханическая визуализация кодирования сигналов» (ОВКС)	1	Стенд для сдачи задач

Наименование	Количество	Описание
Образовательный комплекс «Беспроводные технологии связи» в стандартной версии	10	Тренировочные комплексы для работы команд

Система оценивания

Комплексы оснащены системой аппаратной проверки, что обеспечивает объективность измерений. Все задачи обладают автоматизированной системой проверки, все физические системы калибруются перед соревнованиями. Алгоритмы ведут себя одинаково на различных стендах. Все это обеспечивает предельную прозрачность судейства.

Итоговый балл за практический тур считается как сумма баллов за лучшую попытку по каждой задаче. Команды получали от 0 до 60 баллов: команда, набравшая наибольшее число баллов среди других команд, становилась командой-победителем. Все результаты команд нормировались по формуле

$$100 \times \frac{x}{MAX},$$

где x — число баллов, набранных командой, MAX — число баллов, максимально возможное за практический тур $MAX = 60$ баллов.

Описание задач

Программно-аппаратная платформа стенда ОВКС позволяет работать с физической модуляцией сигнала и анализом данных, полученных после модуляции. В задаче на данном стенде в этом году участникам было предложено 2 типа кодирования и соответственно представлены данные сигналов, промодулированных 2 типами кодирующих шестеренок. И перед участниками стояла задача декодировать сообщения на шестеренках, используя различные методы анализа данных. Кроме этого, в этом году участникам предлагалось разработать свой код для передачи двоичного сообщения и создать шестерню, данные с которой передаются в заранее написанный ими декодирующий.

Программно-аппаратная платформа стенда УНКС комплекса «Беспроводные технологии связи» позволяет работать с адаптивными алгоритмами и точностью передачи сигнала. В задачах рассматривались вопросы слежения за подвижным источником сигнала с учетом его диаграммы направленности и вопросы стабильной передачи данных в канале. Потери данных в задачах этого года происходили как из физического заужения апертуры приемника, так и из наличия препятствий между приемником и передатчиком на некоторых участках траектории передатчика. Задачи составлялись из разного соотношения этих факторов, а также из наличия дополнительного шума в данных и отличий скоростных характеристик траекторий передатчиков. Задачи посвящены адаптивной дешифрации в реальном времени. Последнее приводит к созданию протоколов передачи данных под конкретные условия шумов, требования к надежности, характеру помех в канале, скорости передачи. Этот тип задач является критически важным, когда мы говорим о дальнейшем развитии не только технологий связи для спутников или подводных аппаратов, но и для домашних устройств, систем сенсоров, сельскохозяйственных дронов.

Задачи командного практического тура были составлены таким образом, чтобы они, с одной стороны, были достаточно независимыми, а с другой образовывали взаимосвязанную цепочку. Участникам давалось 5 задач, выполняемых на 2 стендах — 2 задачи на стендах ОВКС, 3 задачи на стендах УНКС. Для всех команд задачи были сформулированы одинаково. В задаче на стенде ОВКС командам был дан один и тот же набор закрытых шестеренок для того, чтобы обеспечивать равенство условий. Обратная задача ОВКС принималась на одном и том же стенде при одинаковых параметрах. В задачах на стенде УНКС для равенства условий каждая из 3 задач проверялась на строго закрепленном стенде. Задачи могли сдаваться в любой последовательности. Количество одновременно решаемых задач определяло слаженность командной работы.

Задача VI.2.6.1. Работа на стенде ОВКС. Восстановление закодированного сообщения

В рамках этой и следующей задачи вы работаете со стендом «Оптомеханическая визуализация кодирования сигналов». Стенд моделирует кодирование сигнала и воспроизводит физические особенности его передачи. Стенд представляет собой оптомеханический формирователь сигнала на основе системы шестеренок, находящихся в зацеплении и вращаемых шаговым двигателем. На шестеренки наносится рисунок, пропускающий при определенных углах поворота шестеренки инфракрасное излучение от светодиода к фотодиоду (принцип оптического энкодера).

На шестернях закодированы сообщения двумя разными кодами. Для каждого из двух кодов вы получите шестерню с образцом кода (далее — образцовая), три индивидуальных шестерни (открытые, обозначены тремя номерами в порядке возрастания, уникальными для каждой команды) и общий для всех команд набор из десяти файлов, содержащих сигналы с шестерней, рисунок которых не выдается (далее — закрытые).

Описание кодов и данных закрытых шестерней помещены в папку ОВКС на рабочем столе тренировочного стенда. Для каждой шестерни снято около трех оборотов, с началом из случайного положения. Оптопары, на которых снят сигнал, немного различаются, поэтому соответствие записанного на шестернях сигнала снимаемому оптопарой может быть различным в данных с разных шестерней.

Описание кодов приведено в приложении ??.

Формат записи данных в файл

4 столбца данных, которые разделены пробелом. В первом столбце отсчет времени в Tick — номер временного тика (каждые 2 миллисекунды). Следующие три столбца содержат уровень сигнала в единицах АЦП с оптопары энкодера, который подключен в соответствующий номеру разъем — PD1, PD2, PD3.

В файлах сигналов с закрытых шестерней используется первый энкодер.

Задание

Используя данные сигналов, для каждой шестерни (3 открытых и 10 закрытых для каждого из двух кодов) нужно определить исходную символьную последователь-

ность (сообщение) для одного оборота.

Стартовое условие — это маркер начала очередного оборота шестерни. Сообщение закодировано между двумя стартовыми условиями.

При этом в коде E может присутствовать пустое пространство, которое сливается со стартовым условием, но к нему не относится.

Проверка и зачет осуществляется согласно порядку и расписания приема задач. На судейском стенде принимаются последовательности разгаданных шестерней, подробнее смотрите в приложении ??.

Результат работы

26 символьных последовательностей, проверяемых независимо друг от друга.

Критерий проверки

Соответствие по расстоянию Левенштейна:

- сообщение без ошибок — 1 первичный балл;
- 1 ошибка — 0,5 первичных баллов;
- 2 ошибки — 0,25 первичных баллов;
- 3 ошибки — 0,1 первичного балла;
- большее количество ошибок — 0 баллов.

По каждой шестерне берется максимальный результат из всех попыток.

Оценка

Максимальная сумма 10 рейтинговых баллов, рейтинговый балл пропорционально первичным так, что полный балл соответствует 26 точно расшифрованным шестерням (т. е. 26 первичных баллов).

Решение

В обоих кодах первыми действиями будет выделение сигнала, соответствующего одному обороту, для чего необходимо выделить стартовое условие (в обоих случаях это длительный высокий уровень сигнала). Далее в пределах оборота выделяются боды (сегменты сигнала), это достигается вычислением первой производной по форме сигнала и выделением резких всплесков значений.

При решении кода E необходимо выделить уровни внутри бодов и сопоставить их по плотности с логическими уровнями, после чего последовательно вычислить символы согласно протоколу. Ниже приведен график (рисунок VI.2.2) образцовой шестерни кода E с построенным графиком первой производной.

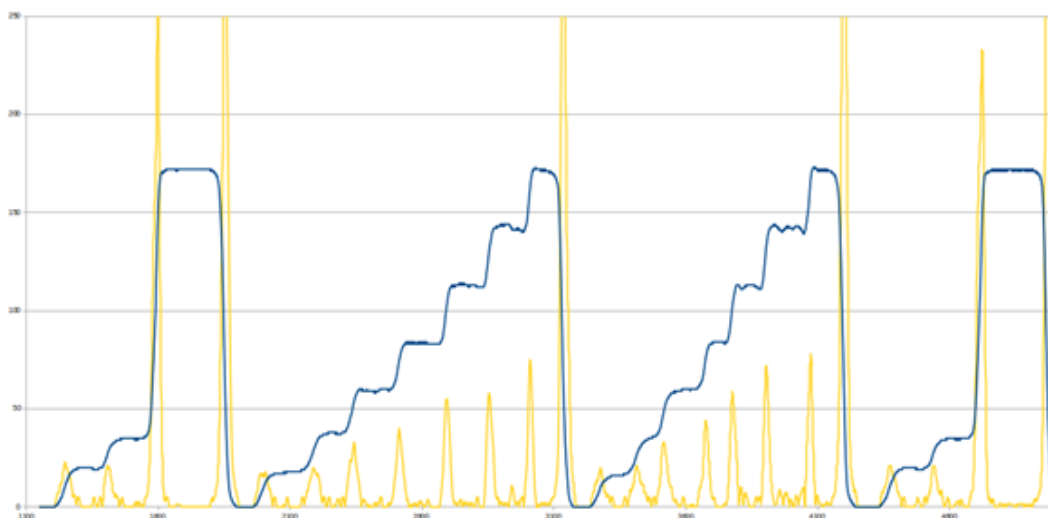


Рис. VI.2.2. График данных с образцовой шестерни кода E с графиком производной

На основании шестеренок, данных для примера, и на основании описаний их устройства очевидно, что границы бодов примерно совпадают со значимыми пиками производной. Их можно найти алгоритмически, выделив резкие всплески. Для определения уровня сигнала бода можно взять значение в его центре; это не безошибочная эвристика, но достаточная для начальной гипотезы. Далее требуется эмпирическая классификация физических параметров бода (высота сигнала) в логические (уровень заливки), после чего их можно декодировать.

При решении кода F (рисунок VI.2.3) кроме выделения границ бодов необходимо определить разницу между левым и правым уровнем бода, но эту величину можно вычислить косвенно по значению первой производной в середине очередного бода

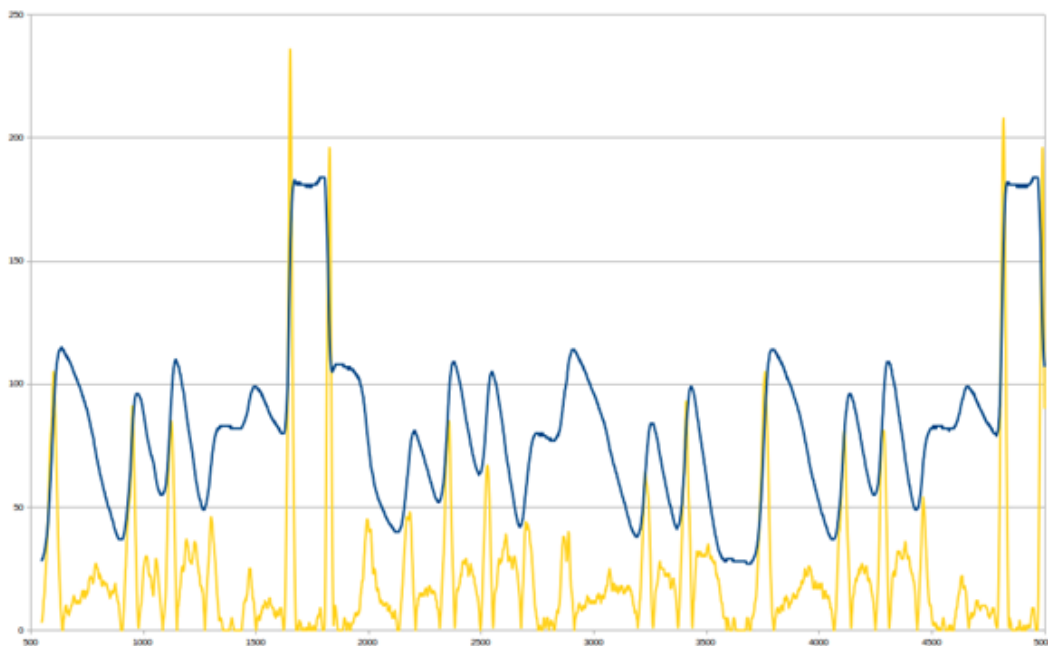


Рис. VI.2.3. График данных с образцовой шестерни кода F с графиком производной

Оба кода можно решать вручную, по графикам, но для более эффективного решения рекомендуется использовать программные средства (например, Python 3 с

библиотеками `numpy` и `scipy`).

Задача VI.2.6.2. Работа на стенде ОВКС. Передача сообщения с помощью шестерни

Вам необходимо выполнить передачу сообщения с помощью нанесения рисунка на шестерню. Сообщение представляет собой последовательность из 70 произвольных двоичных бит. Последовательность необходимо закодировать рисунком на шестерне, сигнал с которой будет передан в написанную вами программу-декодировщик. Восстановленное сообщение сравнивается с исходным, и ваш результат определяется числом корректно переданных бит.

Формат программы-декодировщика

Декодировщик пишется на C, C++, Java или Python 3 в одном файле. Входные данные программа читает из файла `gears.dat`, а выходное сообщение — в файл `output.dat`. Чтение и запись иных файлов недопустима.

Выходное сообщение выводится как текстовая последовательность нулей и единиц (например, 0101010110).

Формат входных данных декодировщика

4 столбца данных, которые разделены пробелом. В первом столбце отсчет времени в `Tick` — номер временного тика (каждые 2 миллисекунды). Следующие три столбца содержат уровень сигнала в единицах АЦП с оптопары энкодера, который подключен в соответствующий номеру разъем — PD1, PD2, PD3.

При зачетном измерении используется первый энкодер.

Задание

Напишите программу-декодер, а затем, используя имеющиеся средства, создайте на шестерне такой рисунок, который в сочетании с декодером позволит передать сообщение (или его часть).

Порядок работы

Передайте оператору с заявкой на задачу программу-декодер. В ответ вы получите файл с сообщением для передачи. Далее вам необходимо самостоятельно создать на шестерне необходимый рисунок.

Сдача попытки производится передачей шестерни оператору. После этого шестерня считывается на судейском стенде с фиксированными параметрами: время измерения 20 секунд, период вращения 6 секунд, амплитуда энкодера 40 единиц. На полученном сигнале запускается ваш декодер. Выходной файл декодера сравнивается с исходным. Вы можете неограниченно сдавать разные шестерни на одном декодере, смена декодера осуществляется в рамках новой попытки и сопровождается генерацией нового сообщения.

Результат работы

Программа-декодер и шестерня с рисунком.

Критерий проверки

Соответствие исходному сообщению по расстоянию Левенштейна. Каждая ошибка снимает 0,5 балла.

Оценка

Максимальная сумма 9 баллов.

Пояснение

Участникам были предоставлены чистые шестерни (без рисунка), черные маркеры, черная изолента, прозрачная пленка формата А4 для лазерного принтера и неограниченный доступ к сетевому лазерному принтеру на площадке проведения.

Решение

Для решения задачи необходимо написать программу-декодер, который сможет обработать данные, снятые со сделанной шестеренки, и вывести код для одного оборота. Принцип реализации этого декодера аналогичен программному решению задачи VI.2.6.1. Чтобы программа поняла, в каком месте начинается оборот, необходимо задать стартовое условие для шестерни – например, отличающийся от остальных по ширине полностью прозрачный бод.

Как пример авторского решения, представленные 70 бит можно разделить на шестерне как 70 бод по 5 градусов ширины, прозрачные (при единице) либо 100% плотности (при нуле). Из этого следует, что на стартовое условие остается $360 - (70 \cdot 5) = 10^\circ$.

Чтобы сделать шестерню, необходимо либо использовать черную изоленту для создания низкого уровня сигнала (что соответствует нулю), либо написать программу, которая генерирует изображение шестерни, которое распечатывается на предложенной прозрачной пленке и прикрепляется к чистой шестерне. Второй вариант более желателен, также рекомендуется использовать для генерации векторный формат изображения, т. к. он позволяет более точно задавать границы бодов.

С предложенным выше кодом в декодере необходимо во входных данных с шестерни найти места стартового условия (места с наибольшим количеством максимальных значений), и между ними считать код – максимальные значения при 1, минимальные при 0.

Задача VI.2.6.3. Работа на стенде УНКС. Передача данных по каналу связи с шумом и преградами

Канал связи образован подвижной системой «радар-спутник». «Спутник» перемещается вдоль рельсы (ось X) и имеет инфракрасный (ИК) передатчик. «Радар»

способен в режиме реального времени определять угловое положение «Спутника» относительно центрального направления визирования и передавать эти измеренные данные для обработки. «Спутник» движется по круговой орбите вокруг «Планеты», также движущейся по круговой орбите, и перемещение «Спутника» на стенде отражает проекцию орбиты на горизонтальную ось. На пути канала связи существуют физические препятствия в виде ограничения угла видимости ИК-приемника «Радара» (шора) и преград в поле передачи, перекрывающих ИК-передатчик «Спутника». Шора представляет собой пластину, которая перекрывает часть ИК-приемника, сужая его угол обзора примерно до 15 градусов.

Цель работы: максимально точная передача данных по ИК-каналу, обеспеченная максимально точным слежением «Радаром» за «Спутником» и помехозащищенным кодированием сигнала. Необходимо написать программу, реализующую алгоритм сопровождения «Спутника» (входными данными для программы является горизонтальное смещение «Спутника» относительно оси радара ΔX ; для управления радаром используются команды, позволяющие регулировать скорость и направление вращения камеры), а также написать программы, реализующие помехозащищенное кодирование. Подробное описание программ и команд, используемых в них, находится в файле «0_readme.txt» (путь к файлу /home/user/UserExamples/).

Задача

Отследите «Спутник» и передайте данные по каналу связи, в котором присутствует блочный шум. Напишите собственную программу слежения и запустите ее. Напишите собственные программы кодирования `encoder` и декодирования `decoder` на одном из 4 языков, которые позволят защитить данные в канале с помехами. За основу можно взять стандартные программы (путь к файлам /home/user/UserExamples/), которые обеспечивают прямую передачу данных по каналу без помех.

Результат работы

Программы слежения (трекер), кодировщик и декодировщик.

Критерий проверки

Соответствие исходному сообщению.

Оценка

Считается по лучшей попытке. Оценивается точность передачи (программным путем).

Балл за задачу начисляется по формуле

$$y = 6 \cdot \frac{x - k}{100 - k},$$

где k — значение точности передачи при запуске на демонстрационных программах, а x — процент, показывающий точность переданного файла при запуске на программах команды.

6 баллов — максимальное количество баллов при 100% точности передачи контрольного файла по каналу связи в условиях шумов.

0 баллов — точность передачи контрольного файла по каналу связи в условиях шумов, с использованием демонстрационных программ.

Ограничения

Время работы программ не должно превышать 5 минут. После наступления времени передача прерывается. Результат в случае прерывания вы получаете лишь за переданную часть файла с множителем 0,9.

Решение

Для решения задачи необходимо написать три программы:

- кодировщик (преобразует входные данные в передаваемые по неустойчивому каналу);
- декодировщик (принимает данные из канала и формирует выходной файл);
- трекер (выполняет управление радаром для слежения за спутником).

Каждую из трех программ можно писать на одном из четырех языков: C, C++, Java, Python 3. Заметим, что языки программы могут не совпадать между собой (например, допускается загружать трекер на Python, а кодировщик — на Java).

При написании программы можно использовать любые модули, входящие в стандартную библиотеку языка (установка дополнительных модулей не предусмотрена).

Принцип написания и взаимодействия программ со стендом един на всех языках (вплоть до одинаковых и не содержит радикальных отличий), поэтому далее будет обобщенное описание работы с каждой из программ. Конкретные детали и сигнатуры функций есть в предоставленных примерах программ.

Кодировщик

Необходимо реализовать процедуру вида `converter(FILE filein, FILE fileout)`, преобразующую содержимое входного файла и записывающую передаваемые данные в выходной.

Параметры:

- `filein` — входной файл, открытый на чтение в бинарном режиме («rb»).
- `fileout` — выходной файл, открытый на запись в бинарном режиме («wb»).

Пример программы `encoder` на Python 3.6.

```
1 #!/usr/bin/python3
2 def encoder(filein,fileout):
3     blocksize=9
4     while True:
5         s=filein.read(blocksize)
6         if not s:
7             break
8         fileout.write(s)
```

Декодировщик

Необходимо реализовать процедуру вида `converter(FILE filein, FILE fileout, FILE tracklog)`, преобразующую содержимое входного файла и записывающую передаваемые данные в выходной.

Параметры:

- `filein` — входной файл, открытый на чтение в бинарном режиме («rb»).
- `fileout` — выходной файл, открытый на запись в бинарном режиме («wb»).
- `tracklog` — файл журнала трекера, открытый на чтение в бинарном режиме («rb»).

Пример программы `decoder` на Python 3.6.

```
1  #!/usr/bin/python3
2  def decoder(filein, fileout, filelog):
3      blocksize = 9
4      s = filein.read(blocksize)
5      if not s:
6          return
7      while True:
8          fileout.write(s)
9          s = filein.read(blocksize)
10         if not s:
11             break
```

Трекер

Необходимо реализовать процедуру вида `tracker(FILE tracklog)`, выполняющую управление радаром на протяжении всего процесса передачи данных.

Параметры:

- `tracklog` — файл журнала трекера, открытый на запись в бинарном режиме («wb»).

Для взаимодействия с радаром используется модуль `client2server`, предоставляющий одноименный класс `client2server`. У класса есть несколько функций:

- `Status moveStop()` — остановить повороты радара, возвращает `Status`;
- `Status moveLeft(int n)` — поворачивать радар влево со скоростью n (в условных единицах), возвращает `Status`;
- `Status moveRight(int n)` — поворачивать радар вправо со скоростью n (в условных единицах), возвращает `Status`;
- `Status getStatus()` — получить статус состояния радара, возвращает `Status`.

`Status` — 128-разрядное беззнаковое целое со следующей структурой:

0–11 биты: DX (знаковое) — смещение спутника на камере радара

12–15 биты: состояние радара:

- 0 — неопределенное;
- 1 — отключен;
- 2 — включен;
- 3 — перемещается;

16–19 биты: положение радара:

- 0 — неопределенное;

- 1 — поворачивается влево;
- 2 — поворачивается вправо;
- 3 — в крайнем левом положении;
- 4 — в крайнем правом положении;

20–36 биты: точное время на радаре в миллисекундах с начала работы (беззнаковое).

Движение радара после команды выполняется непрерывно, т. е. для его остановки необходимо явно послать соответствующую команду.

Пример программы `tracker` на Python 3.6.

```
1 from client2server import client2server
2 class tracker:
3     def run(tracklog):
4         c2s = client2server()
5         i = 0
6         while i == 0:
7             status = c2s.getStatus()
8             dx = int(status) & 0x0fff
9             if dx > 2048:
10                dx = dx - 4096
11                # tracklog.write(f«{dx}«.encode())
12            if abs(dx) < 500:
13                if abs(dx) < 10:
14                    c2s.moveStop()
15                else:
16                    if dx > 0:
17                        c2s.moveLeft(4)
18                    else:
19                        c2s.moveRight(4)
```

Особенности работы с программами на стенде

Ни один из перечисленных файлов не нужно открывать и закрывать самостоятельно (это делается автоматически вне пользовательского кода).

Все файлы открыты в двоичном формате, то есть чтение и запись осуществляется не текстом, а набором байт. Особенно важно учитывать это при реализации на Python.

Программы работают изолированно друг от друга, поэтому не имеет смысла создавать и взаимодействовать с файлами, кроме тех, которые передаются в параметрах функций. Стандартные потоки вывода и ошибок (`stdout` и `stderr`) недоступны для прямого чтения (т. к. выполняются в изолированной среде). При необходимости журналировать ошибки — делать это необходимо через имеющиеся файлы.

При загрузке программы компилируются, но в случае с Python лишь производится синтаксическая проверка. Ошибки времени выполнения не регистрируются напрямую, но их можно заметить по поведению системы во время передачи. Если она завершилась досрочно — проблема в кодировщике или декодировщике. Если радар не двигается — возможно, в трекаре ошибка времени выполнения.

Задача VI.2.6.4. УНКС, не прямое управление «Спутником»

Канал связи образован подвижной системой «радар-спутник». «Спутник» перемещается вдоль рельсы (ось X) и имеет инфракрасный (ИК) передатчик. «Радар» способен в режиме реального времени определять угловое положение «Спутника» относительно центрального направления визирования и передавать эти измеренные данные для обработки. Угол обзора «Радара» ограничен шорой, в рамках этой задачи в «Радар» предзагружен скрипт, осуществляющий слежение по фиксированной траектории.

Для перемещения по траектории «Радар» опирается на «звезды» — пластину с четырьмя маркерами за осью «Спутника». Каждый из маркеров соответствует определенной абсолютной координате, приблизительно соотносящейся с координатной осью «Спутника» (от -40 см до 40 см). «Спутник» считывает маркеры с помощью камеры, определяет их смещение относительно оси визирования, пересчитывает их в фактические координаты и пересылает в скрипт слежения через статусное число.

Цель работы: максимально точная передача данных по ИК-каналу, обеспеченная максимально точным слежением «Радаром» за «Спутником». Необходимо подобрать для «Спутника» траекторию, максимально соответствующую траектории вращения «Радара». Его программа слежения зафиксирована и выводит в журнал данные отклонения положения «Спутника» от оси визирования (Dx). Передача данных осуществляется без дополнительного кодирования. Подробное описание конфигурации траектории находится в файле «1_trajectory.txt» (путь к файлу /home/user/UserExamples/), пример в файле «config_spu.dat» (в той же папке).

Задача

Опишите траекторию «Спутника» так, чтобы обеспечить максимально возможную точность передачи.

Результат работы

Траектория «Спутника».

Критерий проверки

Объем полученного сообщения.

Пояснения

Обратите внимание, спутник начинает передачу и движение по траектории не сразу! Сначала «Спутник» производит калибровку, проезжая до краев, после возвращается в середину и после этого начинает передачу с движением по траектории. Вы можете отследить начало передачи, направив камеру на передатчик под маркером спутника.

Радар начинает слежение сразу после запуска.

Шумы в канале отсутствуют.

Значение $D\chi$ за пределами 1000 по модулю сигнализирует потерю спутника из области видимости камеры.

При сдаче решения вы можете задать любое начальное положение радара и спутника.

Оценка

Считается по лучшей попытке. Оценивается точность передачи (программным путем).

Балл за задачу начисляется по формуле

$$y = 15 \cdot \frac{x - k}{100 - k},$$

где k — значение точности передачи при запуске на демонстрационных программах, а x — процент, показывающий точность переданного файла при запуске на программах команды.

15 баллов — максимальное количество баллов при 100% точности передачи контрольного файла по каналу связи в условиях шумов.

0 баллов — точность передачи контрольного файла по каналу связи в условиях шумов, с использованием демонстрационных программ.

Решение

Для решения задачи необходимо описать настройки траектории «Спутника» так, чтобы во время движения «Спутник» своим положением максимально соответствовал оси визирования «Радара». В задачу встроен трекер, который выдает в файл лога $D\chi$ — отклонение «Спутника» от оси визирования «Радара». В основе движения «Спутника» лежит кеплеровская орбита, и при нулевом фокусе ее проекция по оси x соответствует сумме двух синусоид. Необходимо восстановить траекторию по $D\chi$, подбирая для этого значения (коэффициенты) орбите «Спутника».

Задача VI.2.6.5. Работа со стендом УНКС. Слежение за «Спутником» с ретрансляцией

Канал связи образован подвижной системой «радар-спутник». «Спутник» передвигается вдоль рельсы (ось X) и имеет инфракрасный (ИК) передатчик. «Радар» способен в режиме реального времени определять угловое положение «Спутника» относительно центрального направления визирования и передавать эти измеренные данные для обработки. В поле канала связи существует физическое препятствие, а также физический ретранслятор в виде зеркала, которое частично покрывает область за препятствием. Угол обзора «Радара» ограничен шорой, радар настроен для передачи абсолютной координаты по «звездам».

Цель работы: максимально точная передача данных по ИК-каналу, обеспеченная максимально точным слежением «Радаром» за передатчиком «Спутника». Необходимо написать программу, реализующую алгоритм сопровождения спутника с учетом его нахождения за препятствием. Подробное описание программ и команд, используемых в них, находится в файле «0_readme.txt» (путь к файлу /home/user/UserExamples/).

Задача

Отследите «Спутник» и передайте данные по каналу связи, в котором нет шума. Напишите собственную программу слежения, которая учитывает наличие ретранслятора и возможность его применения. Используйте передаваемую в статусном числе абсолютную координату радара.

Пояснения

Абсолютная координата радара передается в диапазоне от 0 до 2000 (значения за пределом сигнализируют потерю «звезд») в миллиметрах. Координата 1000 соответствует середине оси спутника.

Шумы в канале отсутствуют.

Оценка

Считается по лучшей попытке. Оценивается точность передачи (программным путем).

Балл за задачу начисляется по формуле

$$y = 20 \cdot \frac{x - k}{100 - k},$$

где k — значение точности передачи при запуске на демонстрационных программах, а x — процент, показывающий точность переданного файла при запуске на программах команды.

20 баллов — максимальное количество баллов при 100% точности передачи контрольного файла по каналу связи в условиях шумов.

0 баллов — точность передачи контрольного файла по каналу связи в условиях шумов, с использованием демонстрационных программ.

Решение

Для решения задачи необходимо написать программу-трекер, которая выполняет управление «Радаром» для слежения за «Спутником», в соответствии с написанием трекера для задачи VI.2.6.3. При этом в решении предусматривается как попытка решения обычным слежением, что даст только баллы минимального порога, так и возможность учесть задачу ретрансляции. С учетом ретрансляции необходимо визуально исследовать конфигурацию стенда, в частности используя полигон, определить приближенное соответствие DX (отклонение «Спутника» от оси визирования «Радара») смещению абсолютной координаты «Радара», вычислить приближенную координату «Спутника» и при входе его в слепую зону смещаться к ретранслятору (одному из зеркал) — такая функция добавляется к трекеру.

Материалы для подготовки

Курсы от разработчиков профиля:

-
- Курс «Технологии беспроводной связи» познакомит с базовыми понятиями беспроводной связи и поможет погрузиться в тематику профиля. Каждая лекция содержит задания для размышления, самопроверки и дополнительные материалы <https://onti.polyus-nt.ru/course/view.php?id=9>.
 - Курс «Технологии беспроводной связи+» возник как дополнение и развитие к профилю с таким же названием и как онлайн ресурс для всех, кто участвует в этом профиле и занимается или планирует заниматься на базе лаборатории по беспроводным технологиям связи <https://onti.polyus-nt.ru/course/view.php?id=12>.
 - Курс «Тематические разборы задач профиля ТБС» <https://onti.polyus-nt.ru/course/view.php?id=3>. В курсе собраны все задачи второго тура профиля за 2016/2017-2020/2021 гг по темам.
 - Видеоразборы задач 2021/2022 года приведены в курсе «Разбор задач второго этапа профиля ТБС НТО 21/22» <https://onti.polyus-nt.ru/course/view.php?id=14>.

Сборники заданий профиля

- Сборник задач профиля «Технологии беспроводной связи» 2016/2017 <https://drive.google.com/file/d/10Hg3YBuH7MrIUqkP1U7orayTk6EVnJtg/view>.
- Сборник задач 2017/2018 <https://ntcontest.ru/docs/7> - Материалы заданий ТБС.pdf.
- Сборник задач 2018/2019 <https://drive.google.com/file/d/1QomB7Ahr1CbL55RSbbIPluchONfD4uGP/view>.
- Сборник задач 2019/2020 <https://drive.google.com/file/d/14WN80Kmh8nJCaBB8KEPlxuRaoP-eNFgm/view>.
- Сборник задач 2020/2021 <https://drive.google.com/file/d/1M9wmIjXm1MjH0riLz-ns1NpbixWu8dCc/view?usp=sharing>.
- Сборник задач 2021/2022 <https://ntcontest.ru/docs/tbs-assignments.pdf>.

Статьи и книги

- Статья «Помехоустойчивое кодирование с использованием различных кодов» <https://habr.com/ru/post/111336/>.
- Статья «Коды Рида-Соломона. Часть 1 — теория простым языком» <https://habr.com/ru/company/yadro/blog/336286/>.
- Статья «Коды Рида-Соломона. Часть 2 — арифметика полей Галуа» <https://habrahabr.ru/company/yadro/blog/341506/>.
- Видео «Коды Хэмминга — Григорий Кабатянский». Что привело Хэмминга к созданию его знаменитых кодов? <https://www.youtube.com/watch?v=DycYyYBwb9E>.
- Линейная аппроксимация <https://prog-cpp.ru/mnk/>, при обработке экспериментальных данных часто возникает необходимость аппроксимировать их линейной функцией.
- Статья «Методы сжатия данных» <https://habr.com/ru/post/251295/>.
- Обзор методов сжатия данных <http://www.compression.ru/arctest/descript/methods.htm>.
- Статья «Нежное введение в автокорреляцию и частичную автокорреляцию» <https://www.machinelearningmastery.ru/gentle-introduction-autocorrel>

[ation-partial-autocorrelation/](#).

- Видео «Основы ЦОС:Корреляционная функция» <https://www.youtube.com/watch?v=SR2ettkhmio>.
- Программирование на Python (<https://stepik.org/course/67/promo>).
- Программирование на Python для решения олимпиадных задач (<https://stepik.org/course/66634/promo>).
- Python: основы и применение (<https://stepik.org/course/512/promo>).
- Программирование на языке C++ для решения олимпиадных задач <https://stepik.org/course/66646/promo>.
- Курс «Введение в машинное обучение» <https://openedu.ru/course/hse/INTRML/>.
- «Код: тайный язык информатики» Чарльза Петцольда <https://www.livelib.ru/book/1000005181-kod-tajnyj-yazyk-informatiki-charlz-pettsold>.

Приложения

Приложение. Описание комплекса «Беспроводные технологии связи»

Образовательный комплекс «Беспроводные технологии связи» создан на стыке научно-инженерных задач и технологических барьеров рынков НТИ — AeroNet и SpaceNet в области передачи сигнала по зашумленным каналам связи, проектирования систем связи для Интернета вещей и между автономными роевыми объектами: спутниками или подводными аппаратами. Комплекс предназначен для обучения принципам беспроводной передачи данных; исследования природы шума; кодирования и декодирования сигнала; управления системами с обратной связью и преодоления когнитивных барьеров, препятствующих развитию беспроводных систем связи.

Комплекс является программно-аппаратным и включает стенды «Оптомеханическая визуализация кодирования сигналов», «Узконаправленные низкоэнергетические каналы связи».

Стенд «Оптомеханическая визуализация кодирования сигналов» (ОВКС) (рисунок 1) — тренажер для визуализации и обучения работе с аналоговыми и цифровыми сигналами, низкоуровневыми протоколами, современными системами кодирования. Стенд представляет собой автоматизированную платформу, осуществляющую оптомеханическую амплитудную и частотную модуляцию интенсивности ИК-сигнала с аппаратным выводом данных. Электрическая модуляция осуществляется управлением интенсивностью излучения оптических излучателей. Каналы передачи данных обладают слабыми собственными помехами, и значительными программно-наведенными. Механическая модуляция осуществляется за счет вариативности установочных элементов, создающими контролируемые помехи для передачи сигналов.

Платформа стенда. Основание, обеспечивающее аппаратное управление стендом. Усиленная верхняя поверхность платформы позволяет фиксировать на ней блоки оптических энкодеров и блоки передач. Точность позиционирования определяется равномерной сеткой отверстий, позволяющей дискретно изменять положение блоков. Возможность изменять положение блоков позволяет менять конфигурацию системы, что расширяет многообразие задач и вариантов их решения.

Блоки-энкодеров (рисунок 2 справа). Блоки представляют собой оптомеханические формирователи сигнала, осуществляющие оптомеханическую амплитудную и частотную модуляцию интенсивности ИК-сигнала. Электрическая модуляция осуществляется управлением интенсивностью излучения оптических излучателей. Для механической модуляции сигнала используются установочные элементы в виде шестерней с прорезями или рисунками различной формы, изменяющие интенсивность ИК — сигнала при вращении.



Рис. 1. Стенд ОВКС

Наборы установочных элементов. Наборы кодирующих шестерней — наборы шестеренок, имеющие разный диаметр и разные прорезы или рисунки, соответствующие различным кодовым последовательностям. Набор передаточных шестеренок — позволяет изменять конфигурацию системы и ее параметры при установке дополнительных передач на блоки передачи.

Блок основной передачи (рисунок 2 слева) — блок, со встроенным шаговым двигателем и системой крепления, позволяющей устанавливать на него передаточную шестерню.

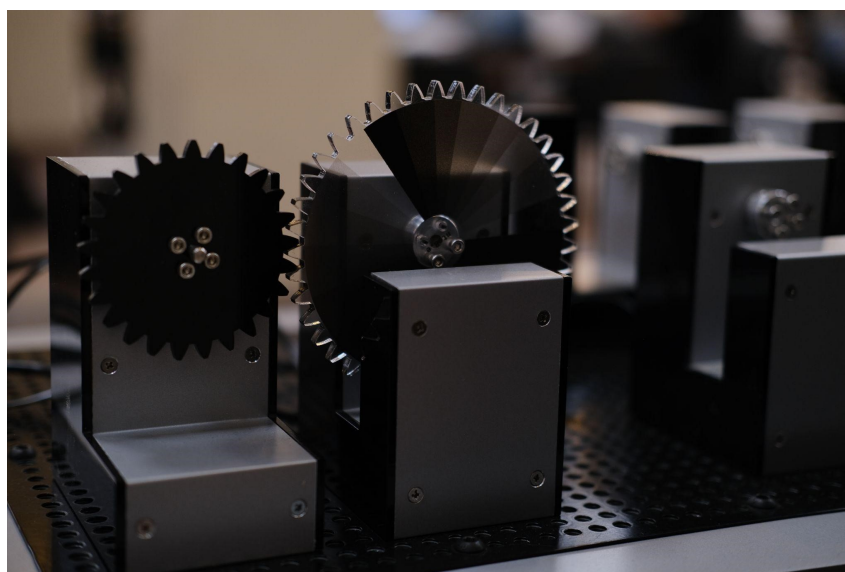


Рис. 2. Блок основной передачи (слева), блок оптического энкодера (справа) с установочным элементом с нанесенным УФ-рисунком

Блоки передач — блоки, позволяющие устанавливать на стенд различные виды

передач, чтобы передавать и менять момент движения. Блоки оснащены системой крепления, позволяющей использовать базовый набор передач, а также изготавливать передачи с использованием 3D моделирования и прототипирования.

Стенд «Узконаправленные низкоэнергетические каналы связи» (УНКС) — тренажер, моделирующий кинематику автономных устройств с системой позиционирования приемника и передатчика, а также моделирующий процесс связи между автономными устройствами по зашумленным каналам связи. Стенд представляет собой автоматизированную платформу, в которой реализована система программно-аппаратного относительного перемещения инфракрасных оптоэлектронных приемника и передатчика, а также реализована возможность вращения приемника вокруг вертикальной оси, положение которого задается шаговым двигателем и регулируется программно. При этом наличие системы видеонаблюдения за передатчиком позволяет осуществлять моделирование процессов оптического слежения за движущимся объектом и наведением на него антенны, что применимо в наземных, подводных и космических комплексах. С точки зрения освоения программирования, стенд позволяет разрабатывать алгоритмы слежения за передатчиком с использованием машинного зрения (OpenCV) и восстановления траекторной информации, осваивать ряд методов и приемов работы с распознаванием видеоизображений и работу с PID регулятором. ИК-каналы передачи данных в устройстве обладают слабыми собственными помехами, и значительными программно-наведенными. Стенд позволяет изучать способы передачи данных по зашумленным каналам, осваивать методы кодирования/декодирования сообщений для каналов связи с различными типами шумов. УНКС со «звездами», преградами и зеркалами представлен на рисунке 3, на рисунке 4 представлено крепление шоры, вместе с установленной в него шорой.

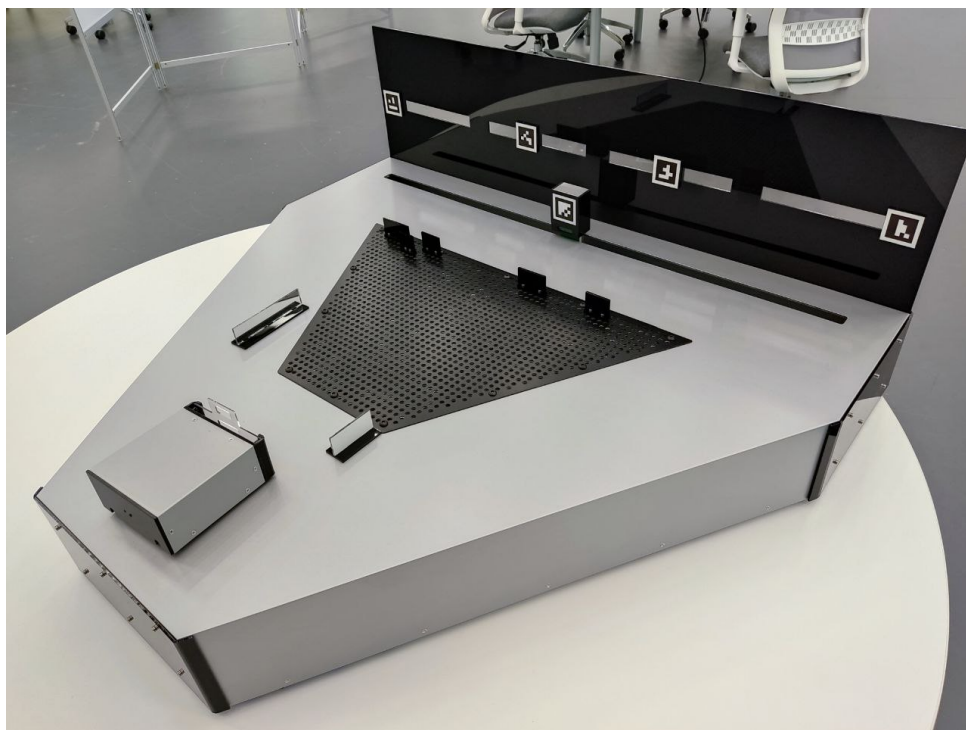


Рис. 3. УНКС со «звездами», преградами и зеркалами

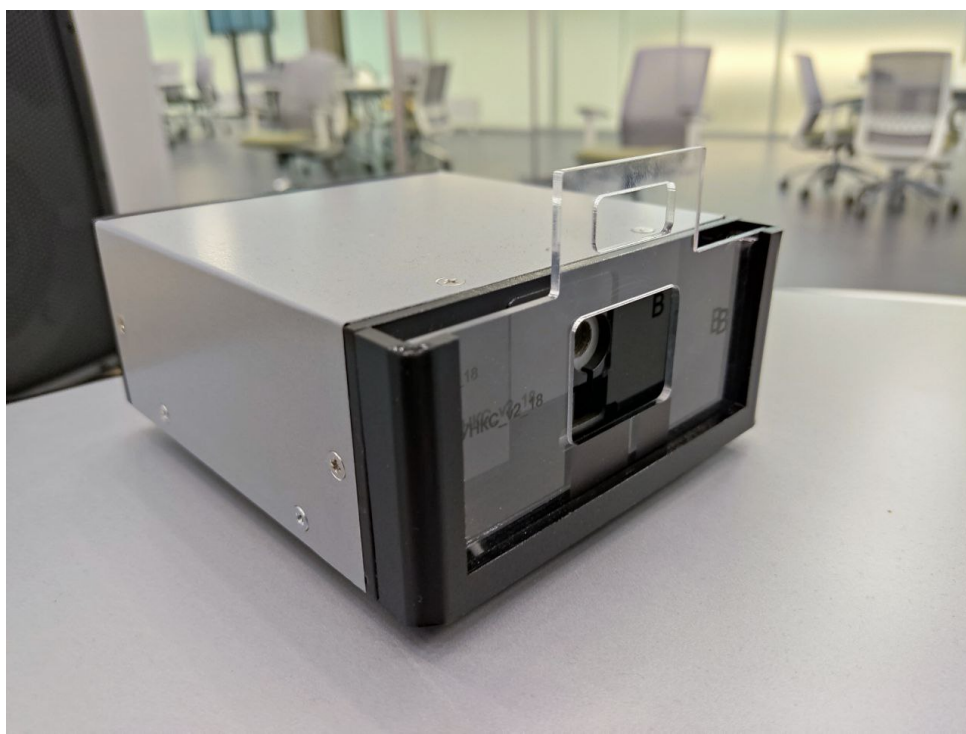


Рис. 4. Шора на «Радаре» УНКС

Приложение. Регламент сдачи решений для зачетных попыток

Расписание слотов для зачетных попыток.

№ слота	Дата	Время
1	07.03.22	15:30 – 16:45
2		16:45 – 18:00
3	08.03.22	13:00 – 14:15
4		14:15 – 15:30
5		15:30 – 16:45
6		16:45 – 18:00
7	09.03.22	09:30 – 10:45
8		10:45 – 12:00
9		12:00 – 13:15
10		14:00 – 15:15
11		15:15 – 16:30
12		16:30 – 17:45
13	10.03.22	09:30 – 10:45
14		10:45 – 12:00
15		12:00 – 13:15

В один слот можно заявить по одной попытке на каждую задачу. Подача заявки на слот осуществляется в течение первых 10 минут после открытия слота.

Для подачи заявки необходимо написать в чат инженерной задачи сообщение с хэштегом «заявка» и указанием номера задачи и команды по принципу:

`#заявка з4 к2 (Задача, сдает команда 2)`.

Передача решения и проверка попыток производится в порядке подачи заявок, результат попытки предоставляется команде в этот же или следующий слот.

Прием решений начинается сразу по мере обработки заявок. При подаче заявки вы уже должны быть готовы сдать решение, в т. ч. параллельно на несколько задач (несколькими флешками). Если команда задерживает передачу решения, судья или оператор имеет право сдвинуть попытку в конец очереди. При повторной задержке или отсутствии попытка отменяется.

На тестовых командных стендах представлены универсальные версии задач УНКС и ОВКС для отработки задач [VI.2.6.1](#), [VI.2.6.2](#), [VI.2.6.3](#), [VI.2.6.4](#) (частично).

Для отработки задач [VI.2.6.4](#) и [VI.2.6.5](#) представлен стенд-полигон, доступ к которому осуществляется во время работы по профилю в порядке живой очереди.

Порядок приема задач на стенде УНКС

Для приема задачи УНКС предоставьте оператору флешку, в корне которой лежат следующие файлы.

Для задачи [VI.2.6.3](#):

- `tracker.*` — программа слежения (эта и следующие программы предоставляются на одном из 4 языков: C, C++, Java, Python 3);

-
- `encoder.*` — программа-кодировщик;
 - `decoder.*` — программа-декодировщик.

Для задачи VI.2.6.4:

- `config_spu.dat` — файл траектории.

Для задачи VI.2.6.5:

- `tracker.*` — программа слежения.

При иных названиях и некорректном размещении оператор оставляет за собой право отложить проверку для исправления размещения и сместить вашу заявку в конец очереди.

Далее оператор загружает программы в интерфейс и выполняет компиляцию. В случае ошибки компиляции по вине команды попытка отклоняется.

В задаче VI.2.6.4 перед началом слежения вы можете сами задать начальное положение радара и спутника.

После выполнения попытки оператор по запросу может загрузить обратно на флешку файл `tracker.log` (журнал слежения). Обратная загрузка иных файлов не предусмотрена.

Ограничения

Исходный файл во всех задачах 256 000 байт.

Время работы системы не более 5 минут во всех задачах. После наступления времени оператор нажимает стоп и передача обрывается. Результат вы получаете лишь за переданную часть файла с множителем 0,9.

За 5 минут успевает передаться объем примерно 1,8 Мбайт.

Если декодированный файл превышает исходный в 1,5 раза, то оценка 0 (файлы не сравниваются).

Порядок приема прямой задачи ОВКС

В течение слота предоставьте оператору на флешке файл текстовом формате, в котором через пробел выписано до 26 пар «номер шестерни — сообщение». Пары приводятся в любом порядке, но среди них не должно быть повторов шестерней (в случае повтора проверяется только первое упоминание шестерни). Например (сообщения произвольные):

```
f01 ABCDABCDABCDABCDABCD
e05 BBBVABBBVAAAABVAA
e13 BAVABAVABAVA
```

Название файла должно иметь следующий вид:

«ОВКС_номеркоманды_номерслота.txt».

Например «ОВКС_4_2.txt».

В случае некорректного формата таблицы оператор оставляет за собой право отложить прием попытки для исправления.

Далее оператор выполняет сравнение и отражает его результат в рейтинговой таблице.

Порядок приема обратной задачи ОБКС

В начале слота предоставьте оператору на флешке файл `gear_decode` с расширениями `ru`, `c`, `cpp`, `java`. Оператор компилирует программу, в случае ошибок компиляции и невозможности их оперативного исправления оператор оставляет за собой право отложить прием декодера.

Далее оператор сгенерирует и загрузит на флешку файл, содержащий сообщение, которое необходимо закодировать.

Создайте рисунок шестерни, который в сочетании с декодером сформирует искомое сообщение, и передайте его оператору в порядке живой очереди. Вы вправе сдавать шестерню многократно, в том числе в следующие слоты. Замена декодера производится в рамках новой попытки и влечет изменение сообщения (оператор вновь генерирует файл и передает его вам).

Оператор снимет сигнал и запустит на файле сигнала ваш декодер. В случае ошибок выполнения и невозможности их оперативного исправления попытка считается завершенной.

Приложение. Описание кодов к задаче VI.2.6.1.

Код E

В этом коде данные закодированы уровнем сигнала, а вернее, переходом уровня между сегментами. Сегменты имеют переменную ширину в 17 ± 5 градусов и уровень, который зависит от плотности закраски рисунка (чем темнее, тем выше плотность и тем ниже уровень).

Соответствие плотности и уровня представлено ниже.

Уровень	0	1	2	3	4	5	6	7
Плотность	0%	14%	28%	42%	56%	70%	84%	100%

Символ сообщения кодируется **переходом** между уровнями по циклическому принципу и зависит от текущей буквы. При этом переход между сегментами происходит всегда (не существует подряд идущих сегментов одного цвета).

Алфавит — ABCDEFG.

Так, переход на 1 уровень вверх означает, что в сообщение записывается та же самая буква, переход на 2 уровня вверх — следующая по алфавиту (или А, если до этого была G), переход на 1 уровень вниз — предыдущая буква. Первый переход (из стартового условия) производится из буквы А, при этом она в сообщение не заносится.

Пример таблицы для буквы В

Какой уровень был	Какой уровень стал							
	0	1	2	3	4	5	6	7
0	-	B	C	D	E	F	G	A
1	A	-	B	C	D	E	F	G
2	G	A	-	B	C	D	E	F
3	F	G	A	-	B	C	D	E
4	E	F	G	A	-	B	C	D
5	D	E	F	G	A	-	B	C
6	C	D	E	F	G	A	-	B
7	B	C	D	E	F	G	A	-

Стартовое условие — 30 градусов нулевой плотности.

Код F

В этом коде сегменты, формирующие данные, представлены градиентными переходами плотности фиксированной ширины (20 градусов). Градиентные переходы осуществляются между произвольными уровнями, значащей является величина перехода, которая и определяет закодированный символ:

Символ	A	B	C	D	E
Переход	5%	20%	35%	50%	65%

Алфавит — ABCDE.

Стартовое условие — 20 градусов прозрачного.

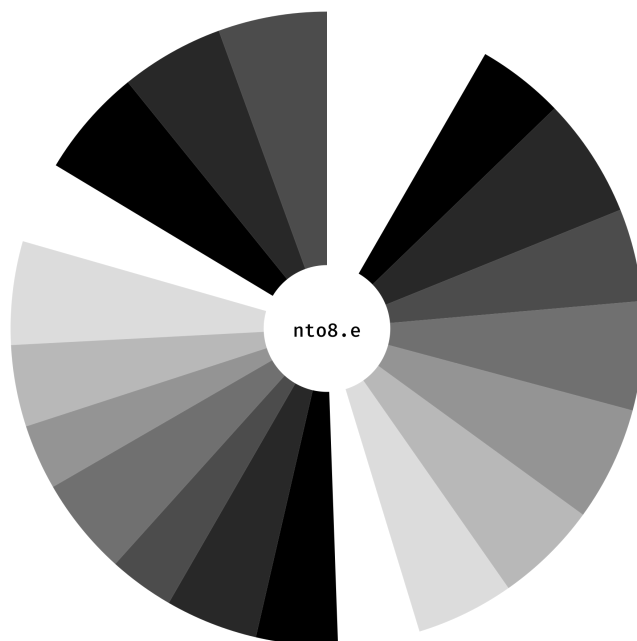


Рис. 5. Образцовая шестерня кода E, сообщение GFEDCBAGFEDCBAGFEDC

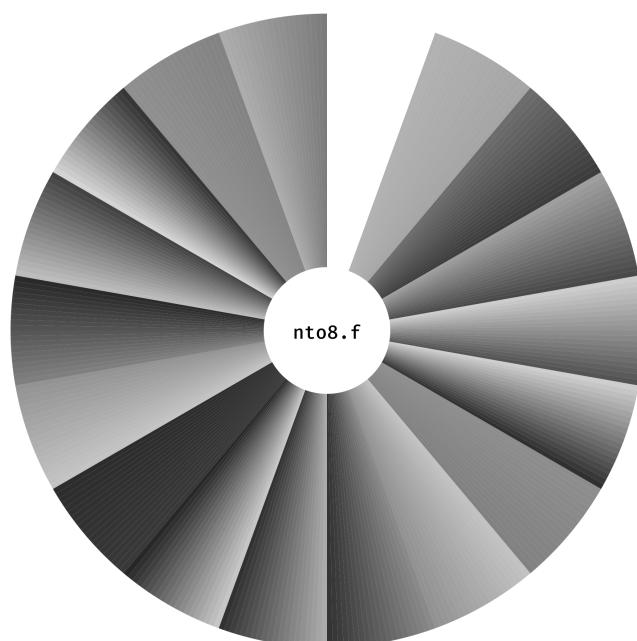


Рис. 6. Образцовая шестерня кода F, сообщение ABCDEABCDEABCDEAB