

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Кафедра «Прикладная математика»

Направление 01.04.02 – «Прикладная математика и информатика»

Профиль – «Математика и информатика в науке и образовании»

К ЗАЩИТЕ ДОПУСКАЮ

Заведующий кафедрой

_____ А.Л. Григорьева

«___» _____ 2026 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Разработка робастного цифрового двойника
изолированной гибридной энергетической системы

Студент группы 4МИМ-1

подпись, дата

Е.О. Шелопугина

Научный руководитель,
канд.техн. наук, доцент

подпись, дата

В.Д. Бердоносков

Нормоконтролёр

подпись, дата

О.В. Козлова

Рецензент

подпись, дата

А.Н. Анисимов

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Кафедра «Прикладная математика»

УТВЕРЖДАЮ

Зав. кафедрой

_____ А.Л. Григорьева

«____» _____ 2026 г.

З А Д А Н И Е **на магистерскую диссертацию**

Выдано студенту Шелопугиной Еве Олеговне

Тема магистерской диссертации Разработка робастного цифрового двойника
изолированной гибридной энергетической системы.

утверждена приказом по университету № 0317-ЛСС-ГИА-ФКТ от 11.12.2024

Срок сдачи студентом законченной работы _____

Исходные данные к работе научная и учебная литература.

Перечень подлежащих разработке вопросов: _____

а) изучение предметной области;

б) изучение и анализ оборудования для изолированной гибридной энергетической системы;

в) написание математической модели информационно-математического обеспечения;

г) разработка информационно-математического обеспечения;

д) тестирование и отладка информационно-математического обеспечения.

Перечень графического материала: *не предусмотрен.*

Задание принял к исполнению _____ « ____ » _____ 202_ г.
(подпись)

Руководитель, _____ *Бердоносков Виктор Дмитриевич*
(подпись) (Ф.И.О.)
должность, ученая степень *канд. техн. наук, доцент* « ____ » _____ 2026 г.

Аннотация

Разработка робастного цифрового двойника изолированной гибридной энергетической системы

Пояснительная записка 133 с., 9 рис., 8 табл., 43 источника, 1 приложение

В работе разработан робастный цифровой двойник изолированной гибридной энергосистемы на примере горнолыжной базы «Холдоми» в Хабаровском крае. Подобрано основное оборудование с учётом климатических условий региона и юридической чистоты закупки. Построена математическая модель, включающая ETL-процесс для метеоданных, расчёт солнечного потока, суточный энергопрофиль и баланс мощностей. Создано веб-приложение на Python (Streamlit) для имитационного моделирования и визуализации результатов, которое позволяет оценивать долю покрытия нагрузки, дефицит и потребление из внешней сети при различных сценариях.

Abstract

Development of a robust digital twin of an isolated hybrid energy system

Explanatory note 133 p., 9 fig., 8 tabl., 43 sources, 1 enclosure

This paper presents a robust digital twin of an isolated hybrid energy system, developed for the «Kholdomi» ski resort in Khabarovsk Krai. The key equipment is selected based on regional climate conditions and legal procurement aspects. A mathematical model is constructed, covering ETL data handling for meteorological inputs, solar flux calculation, daily load profiling, and power balance. A Python web application (Streamlit) is implemented for simulation and visualization, enabling assessment of load coverage, energy deficit, and grid draw under different scenarios.

Содержание

Введение.....	7
1 Цифровой двойник (ЦД).....	9
1.1 Определения, связанные с цифровым двойником.....	9
1.2 Роль ЦД в моделировании и анализе солнечной энергии	13
1.3 Принципы построения ЦД для поступающей солнечной энергии	14
2 Выбор оборудования для составления ИГЭС	23
2.1 Подбор оборудования для солнечной части ИГЭС.....	23
2.2 Подбор оборудования для аккумуляторной части ИГЭС.....	29
2.3 Подбор оборудования для инверторной части ИГЭС.....	34
2.4 Подбор оборудования для контроллерной части ИГЭС.....	40
3 Интеграция данных	48
3.1 Сбор данных о географических координатах, времени и метеорологических параметрах.....	49
3.2 Процесс интеграции данных в цифровой двойник.....	51
4 Математическая модель ЦД и методы моделирования	53
4.1 Математическая модель процесса получения, преобразования и хранения данных облачности и температуры (ETL-модель).....	55
4.2 Математическая модель расчёта солнечного потока	60
4.3 Математическая модель суточного энергетического профиля и расчёта параметров гибридной изолированной энергосистемы	66
4.4 Математическая модель согласования фотоэлектрического поля, инверторов и аккумуляторного банка.....	74
4.5 Модель энергетического баланса ИГЭС	79
5 Анализ результатов и интерпретация	84

5.1 Интерпретация данных Цифрового Двойника в ИГЭС	84
5.2 Веб-приложение для ЦД ИГЭС и его испытание.....	87
5.3 Оценка достижения поставленных целей и задач	91
Заключение	93
Список использованных источников	95
ПРИЛОЖЕНИЕ А Программный комплекс цифрового двойника изолированной гибридной энергетической системы в формате веб-приложения.....	101

Введение

Цифра уже везде. Промышленность, энергетика, транспорт, услуги, критическая инфраструктура – всё это сейчас переупаковывается цифровой трансформацией. И одно из самых интересных направлений здесь – цифровые двойники. Виртуальные копии реальных объектов и процессов, которые дают возможность анализировать состояние, прогнозировать поведение и принимать решения, опираясь на данные.

В энергетике двойники особенно нужны. Требования к надёжности, эффективности, устойчивости растут постоянно, системы усложняются, а доля солнца и ветра – с их непостоянной выработкой – всё выше. Без интеллектуальных инструментов мониторинга, моделирования и оптимизации режимов уже никак.

Труднее всего с изолированными системами. Те, что работают сами по себе, без подключения к централизованной сети. Для них критически важна бесперебойность при ограниченных ресурсах, переменной нагрузке и высокой чувствительности к климату. Добавьте сюда неравномерное потребление, необходимость как-то рационально распределять энергию между источниками и накопителями, держать всё стабильным в условиях неопределённости [1].

Гибридные изолированные системы – когда вместе дизель, солнце, ветер, накопители – дают больше автономности. Но управлять этим хозяйством не просто: погода, сезонные изменения нагрузки, техническое состояние оборудования, параметры батарей. Тут как раз и нужны робастные подходы – чтобы цифровая модель и алгоритмы не сыпались при помехах и неопределённостях.

Создать робастный цифровой двойник для такой системы – задача на стыке науки и практики. Он должен адекватно описывать работу энергосистемы в реальном времени, прогнозировать её состояние и помогать принимать решения по управлению энергопотоками. Тогда анализ становится точнее, рисков нерационального использования ресурсов меньше, а система надёжно работает в разных эксплуатационных режимах.

Отдельный интерес – горнолыжная база. Нагрузка там жёстко сезонная: то пик, то затишье. Всё зависит от интенсивности работы инфраструктуры, климата, числа посетителей. Автономная или полуавтономная гибридная система может лечь в основу энергоснабжения, а цифровой двойник – стать инструментом её анализа, прогнозирования и оптимизации. Всё это отлично ложится в тренды распределённой энергетики, повышения энергоэффективности и цифровизации удалённых объектов.

Так что разработка такого робастного двойника – задача не надуманная. Она важна и для теории, и для реальной жизни.

1 Цифровой двойник (ЦД)

1.1 Определения, связанные с цифровым двойником

Прежде чем браться за создание робастного цифрового двойника, стоит прояснить несколько базовых вещей. Термины, которыми будет оперировать данная работа, что даст больше ясности. Это важно потому, что понятие «цифровой двойник» в литературе и практике нередко используется широко, однако в инженерной и нормативной среде оно имеет вполне определенное содержание. Для настоящего подраздела за основу логично взять действующий российский стандарт ГОСТ Р 57700.37-2021, поскольку именно он устанавливает общие положения разработки и применения цифровых двойников изделий и остается базовым документом в данной области; при этом в 2025 году были введены и более узкие стандарты, относящиеся к отдельным видам промышленного оборудования, но они не заменяют общее определение, а развивают его для конкретных применений [23, 38].

Согласно ГОСТ Р 57700.37-2021, цифровой двойник изделия – это система, состоящая из цифровой модели изделия и двусторонних информационных связей с изделием (при наличии изделия) и/или его составными частями. Иными словами, цифровой двойник не сводится только к математической модели или визуальной 3D-копии: его сущность заключается именно в наличии обратной связи между реальным объектом и его цифровым представлением. Такая связь позволяет передавать в модель данные о состоянии объекта, а также, при необходимости, использовать результаты расчетов и анализа для корректировки режимов работы реального объекта. Именно двусторонний обмен информацией отличает цифровой двойник от более простых имитационных моделей, которые существуют отдельно от физической системы.

Стандарт также поясняет, что цифровой двойник разрабатывается и применяется на всех стадиях жизненного цикла изделия, а его наполнение и функциональность зависят от конкретной стадии – разработки, производства или эксплуатации [23]. В тексте стандарта отдельно введены связанные понятия:

цифровая модель изделия, цифровые (виртуальные) испытания, цифровой (виртуальный) испытательный стенд и цифровой (виртуальный) испытательный полигон. Кроме того, стандарт задает цель создания цифрового двойника: выполнение технических требований к изделию, снижение сроков и стоимости разработки, повышение технологичности, надежности и эффективности эксплуатации. Отсюда следует, что цифровой двойник в инженерной практике нужен не только для «наблюдения» за объектом, но и для поддержки решений на всем протяжении его существования – от проектирования до эксплуатации и модернизации.

Для полноты понимания полезно кратко обозначить, из каких положений состоит сам стандарт. В ГОСТ Р 57700.37-2021 закреплены общие определения, принципы построения цифровой модели и цифрового двойника, назначение цифровых (виртуальных) испытаний, а также общие требования к разработке цифровых двойников. Отдельно подчеркивается, что при создании цифрового двойника рекомендуется использовать программно-технологическую платформу цифровых двойников, а обновление цифрового двойника должно выполняться за счет двусторонних информационных связей с изделием. Это означает, что цифровой двойник является динамической системой: по мере накопления данных он уточняется, а не остается статичной расчетной схемой. В контексте настоящей работы это особенно важно, поскольку для энергетической системы на горнолыжной базе необходимо учитывать сезонность, погодную изменчивость и переменную нагрузку, то есть факторы, которые постоянно меняют режим работы объекта [23, 42].

Следующий ключевой термин – гибридная энергетическая система. В современной энергетической литературе под *hybrid energy system* понимают систему, включающую несколько технологий генерации, хранения и/или преобразования энергии, которые интегрированы физически или через единую систему управления для повышения эффективности, надежности, гибкости и экологических показателей по сравнению с отдельным использованием этих технологий. В более прикладном смысле гибридные системы могут объединять несколько

источников энергии в одной точке присоединения и использовать общую архитектуру управления. Для настоящей работы это определение следует адаптировать к предметной области: гибридная энергетическая система рассматривается как система, в которой совместно используются солнечные панели как локальный возобновляемый источник и внешняя центральная энергосистема как дополнительный или резервный источник питания. Такая конфигурация позволяет покрывать часть нагрузки за счет фотоэлектрической генерации, а недостаток энергии компенсировать из внешней сети [2, 4, 15, 43].

Поскольку в работе используется именно изолированная система, важно отдельно пояснить, что под этим понимается, если объект имеет возможность получать электроэнергию из центральной энергосистемы. В энергетике и микросетях режим «изолированный» означает работу системы при отключении от главной сети, когда локальные генераторы, накопители и нагрузки функционируют автономно и баланс мощности обеспечивается внутри самой системы. Из этого следует, что наличие линии связи с центральной энергосистемой само по себе не отменяет изолированность. В рамках данной работы систему можно считать изолированной, если ее нормальная и расчетная устойчивость обеспечиваются локальными источниками и управлением, а внешний источник выступает как вспомогательный канал, используемый по необходимости, например в аварийных, пиковых или дефицитных режимах. То есть внешняя сеть не является обязательным условием существования системы, а лишь расширяет диапазон ее эксплуатации; именно поэтому объект остается изолированным с точки зрения своей основной энергетической архитектуры [5, 14].

Такой подход особенно хорошо согласуется с задачами цифрового двойника. Для изолированной гибридной системы цифровой двойник должен описывать не только производство энергии солнечными панелями, но и особенности перераспределения потоков между локальной генерацией, внешней сетью и нагрузкой объекта. На практике это значит, что модель должна учитывать погодные условия, сезонные изменения солнечной инсоляции, график потребления, ограничения по мощности присоединения и возможные переходы между

режимами работы. Иначе говоря, цифровой двойник в данном случае является инструментом не только моделирования, но и выбора режима энергоснабжения в зависимости от текущих условий эксплуатации. Для горнолыжной базы это особенно актуально, поскольку нагрузка носит выражено сезонный характер, а надежность питания в пиковые периоды имеет критическое значение [8, 16].

Наконец, при разработке робастного цифрового двойника необходимо определить и понятие робастности. В теории управления робастная система – это система, которая остается устойчивой и сохраняет требуемые показатели качества при расхождении между реальным объектом и расчетной моделью, а также при наличии неопределенностей параметров, возмущений и неучтенной динамики [8]. В более общем инженерном смысле робастность означает способность системы сохранять заданные свойства при воздействии возмущений и изменении условий работы. Для цифрового двойника это означает, что он должен быть достаточно устойчивым к неполноте исходных данных, изменению погодных условий, сезонным колебаниям нагрузки, деградации оборудования и другим неопределенностям, неизбежным для реального энергетического объекта. Таким образом, робастность здесь следует понимать не как «идеальную точность» модели, а как ее способность оставаться полезной и адекватной в широком диапазоне режимов работы системы [39, 40, 41].

В результате можно сформулировать рабочую терминологическую основу для дальнейшего исследования. В данной работе цифровой двойник понимается как динамическая система, связанная с физическим объектом двусторонним обменом данными; гибридная энергетическая система – как объединение солнечной генерации и внешнего сетевого источника; изолированная система – как система, способная автономно обеспечивать энергоснабжение и сохранять независимую архитектуру управления даже при наличии резервной связи с центральной энергосистемой; робастность – как способность цифрового двойника и алгоритмов управления сохранять приемлемое качество работы при неопределенности и возмущениях. Именно на этой понятийной базе далее можно пе-

реходить к описанию структуры и принципов построения цифрового двойника изолированной гибридной энергетической системы для горнолыжной базы.

1.2 Роль ЦД в моделировании и анализе солнечной энергии

Цифровой двойник сегодня плотно вошёл в самые разные области – и энергетика не исключение. По сути, это виртуальная модель физического объекта или процесса, которая повторяет его поведение в реальном мире, живёт с ним параллельно. И когда речь заходит о солнце, о солнечной генерации, двойник становится инструментом, без которого уже сложно представить нормальный анализ и моделирование.

Солнечная энергия – это, пожалуй, самый массовый возобновляемый источник. Но чтобы выжимать из неё максимум, требуется детально всё просчитывать и анализировать. Вот тут и выходит на сцену цифровой двойник – как средство прогнозирования работы солнечных систем и повышения их эффективности. Он представляет собой виртуальную копию, которая довольно точно воспроизводит и физическую структуру, и работу реальной установки. Внутри модели зашиты географическое положение, погодные условия, характеристики самих панелей, электрические компоненты – всё, что имеет значение. Дальше с этим можно играть: запускать сценарные анализы, искать оптимальные параметры. Скажем, прикинуть, как на производительность системы повлияют разные углы наклона панелей, их ориентация, интенсивность излучения [22, 24, 42, 43]. Это уже прямая дорога к стратегиям управления, нацеленным на максимальную выработку.

Главный плюс – в возможности виртуальных испытаний. Не надо тратить реальные ресурсы, гонять железо или ждать нужной погоды. Можно прогонять разные сценарии и стратегии без риска для живой системы, а заодно экономить время и деньги. Плюс цифровой двойник позволяет заглядывать вперёд – предсказывать, как система поведёт себя, опираясь на текущие условия и накопленный опыт.

Выходит, что в моделировании и анализе солнечной энергии двойник – штука ключевая. Он даёт точную картину работы системы, позволяет проводить сценарные расчёты, виртуальные тесты и прогнозирование. Всё это упирается в эффективное управление солнечной генерацией, снижение затрат и рост энергоэффективности.

1.3 Принципы построения ЦД для поступающей солнечной энергии

Цифровой двойник будет разработан для горнолыжной базы «Холдоми» (рисунок 1.1) как для территориально и функционально специфического объекта, энергопотребление которого определяется не только режимом эксплуатации инфраструктуры, но и выраженной сезонной неравномерностью туристической нагрузки, климатическими колебаниями и особенностями удаленного расположения. Такое моделирование позволит описать объект не в статике, а как сложную адаптивную систему, в которой взаимосвязанные энергетические потоки, погодные воздействия и эксплуатационные режимы формируют единую динамическую среду принятия решений.

Чтобы цифровой двойник нормально работал с поступающей солнечной энергией, нужно держать в голове несколько вещей. Без них модель будет оторвана от жизни. Первое – география и климат. Солнечное излучение, уровень освещённости, скачки температуры, всё, что влияет на панели. Это база. Второе – не упустить ни одной детали самой установки. Панели, инверторы, аккумуляторы, управляющая электроника. Моделируешь всё до винтика – тогда и видишь, как система поведёт себя в разных условиях, и можешь выжать из неё максимум [21, 37].

Дальше – сухие цифры. Технические характеристики каждого элемента. КПД панелей, коэффициенты тепловых потерь, параметры инверторов и батарей. Всё это зашивается в модель. Без реальных цифр двойник останется красивой картинкой. А так – он действительно позволяет точно считать и оптимизировать производительность.

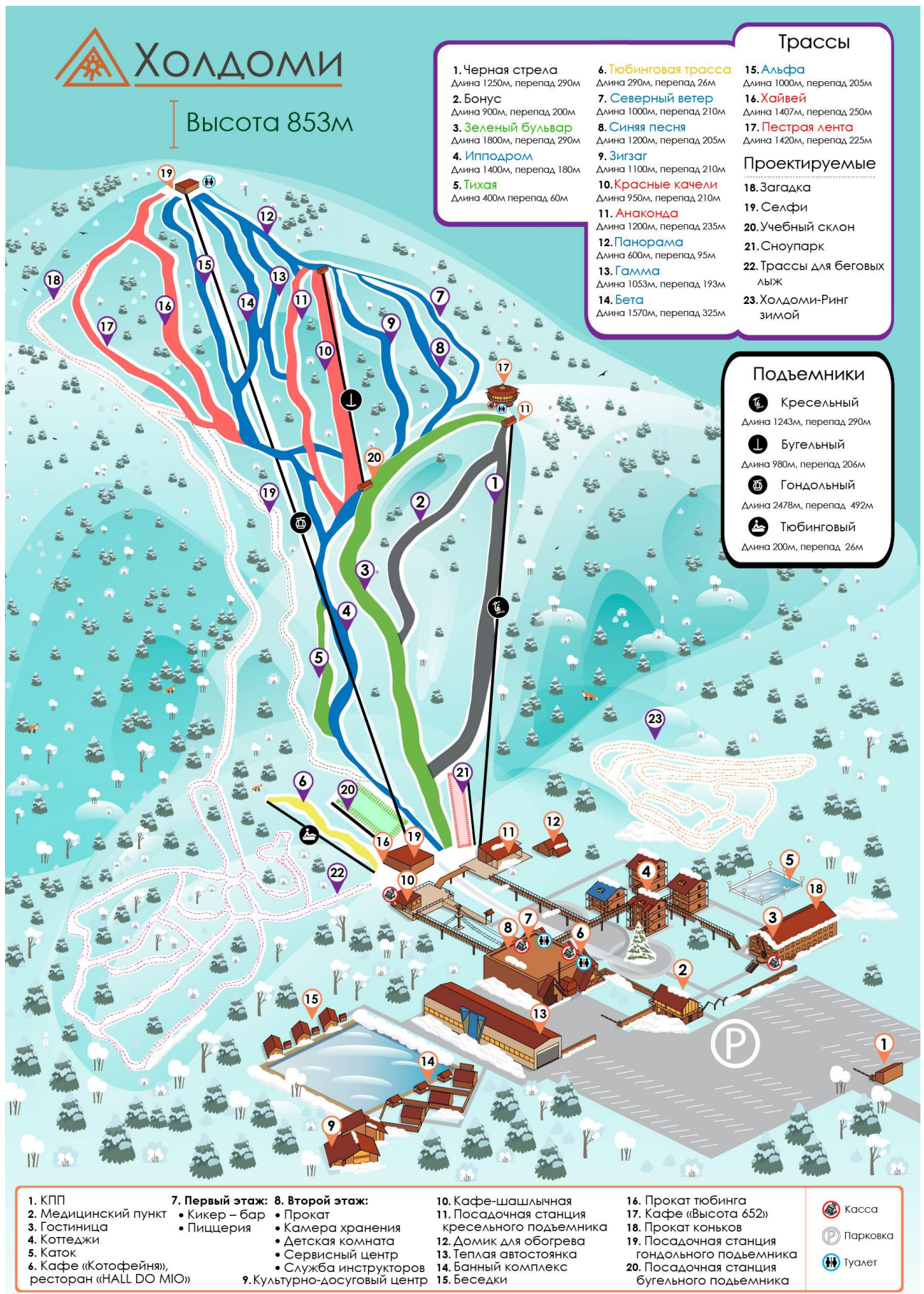


Рисунок 1.1 – Карта горнолыжного курорта «Холдоми»

Двойник хорош ещё и тем, что на нём можно перебирать варианты расстановки панелей, конфигурации всей системы. Искать лучший угол наклона, ориентацию, подбирать ёмкость батарей. Или, скажем, прикинуть, насколько сильно тень от соседнего здания или дерева режет выработку.

И про погоду. Модель умеет заглядывать вперёд, опираясь на текущие условия и прогноз. Оператор видит, чего ждать, и принимает осмысленные решения: что подкрутить, как перенастроить систему под надвигающиеся облака или ясное небо.

Коротко говоря, все эти принципы работают на одну цель – сделать максимально точную виртуальную копию солнечной установки. Чтобы анализировать, оптимизировать, управлять. С прицелом на то, что солнца в нашем энергетическом будущем будет много.

Теперь о конкретном объекте. Горнолыжная база «Холдоми» – это окрестности Комсомольска-на-Амуре, Хабаровский край. Энергии она потребляет прилично. Суммарно выходит порядка 3292,2 кВт·ч в месяц, среднесуточному – около 110 кВт·ч. Подъёмники, системы искусственного оснежения, свет на трассах, служебные здания, отопление, вентиляция – всё это тянет. А сети далеко, либо их пропускной способности не хватает. Поэтому без изолированной гибридной системы с фотоэлектрической генерацией, аккумуляторами и резервным подключением к центральной сети надёжного энергообеспечения не получить.

В этой работе задача – разработать цифровой двойник такой ИГЭС, заточенный под имитационное моделирование энергетических процессов. Принципиальное отличие от того, что делалось раньше (в частности, в бакалаврской ВКР, где создали двойник входных энергетических потоков): физической реализации системы на объекте нет. Модель чисто виртуальная.

Для испытаний взяли самое крупное здание комплекса – Базу. Там и пункт проката, и офисы, и кафе, и ресторан. Интересно оно не только размером и удобной плоской кровлей. Главное в другом: заложенная в двойник модель изначально привязана к конкретной геометрии. Она раскладывает панели

именно на этой крыше, со всеми её реальными ограничениями, а не на какой-то идеальной площадке. Это осознанное сужение возможностей. Но оно и приближает цифровую копию к настоящему, живому объекту. Цифровой двойник (ЦД) входных энергетических потоков изолированной гибридной энергетической системы (ИГЭС) ранее содержал три канала: имитационный (теоретический), аппаратный (физический) и вспомогательный (рисунок 1.2).

В теоретическом канале производилась имитация потока солнечной энергии в режиме «чистое небо» (ClearSky), предполагая отсутствие облачности, но учитывая альбедо.

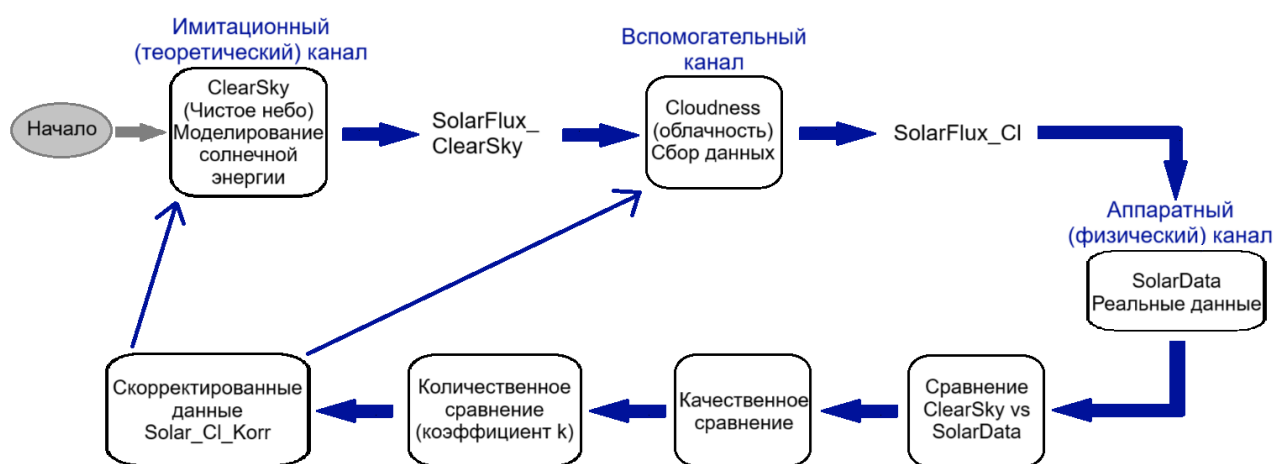


Рисунок 1.2 – Каналы ЦД ИГЭС из ВКР бакалавриата

Во вторичный канал ранее поступали данные об уровне облачности (Cloudness), которые затем использовались для корректировки солнечных потоков в режиме «чистое небо». Это позволяло получить поток солнечной энергии (SolarFlux_CI), наиболее точно соответствующий реальному.

В физическом канале фиксировались данные солнечного потока с реальной установки (SolarData). По мере поступления данных в каждый из каналов производилось сравнение ClearSky и SolarData для выявления качественных и количественных различий. Качественное сравнение позволяло оценить соответствие данных облачности, полученных с внешнего источника (ранее с сайта VentuSky), данным реальной установки. Количественные различия усреднялись

для формирования коэффициента коррекции. Этот коэффициент затем использовался для корректировки данных SolarFlux_CI, формируя SolarFlux_CI_Korr.

На рисунке 1.3 представлена структурная схема ЦД ИГЭС и элементы внешнего взаимодействия первой итерации ВКР бакалавриата.

Трёхканальная архитектура первой итерации позволяла не только моделировать, но и верифицировать расчёты по фактическим данным, а также адаптировать модель к конкретному месту установки. Однако её ключевым ограничением являлась жёсткая привязка к наличию физически смонтированной и действующей энергосистемы – без аппаратного канала верификация и калибровка становились невозможными, что делало модель неприменимой на этапе предпроектного анализа [31, 32, 33].

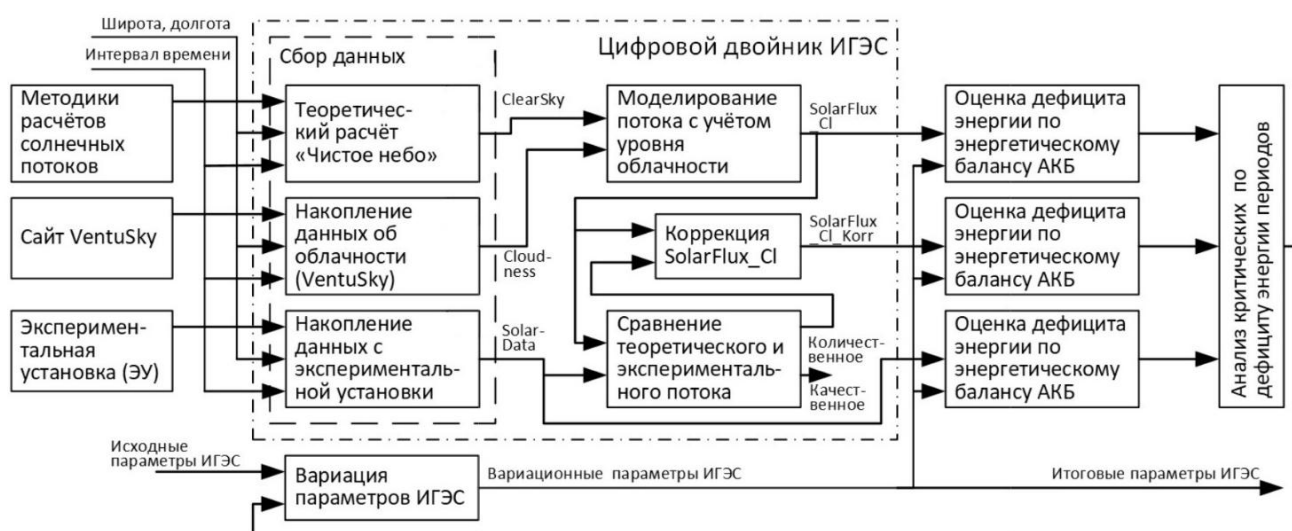


Рисунок 1.3 – Структурная схема ЦД ИГЭС первой итерации

В данной же работе реализация аппаратного канала невозможна по объективным причинам: реальная гибридная энергосистема на горнолыжной базе «Холдоми» на текущий момент не смонтирована и не эксплуатируется, следовательно, отсутствуют экспериментальные данные, которые могли бы быть использованы для верификации или калибровки. Вспомогательный канал, предназначенный для коррекции модели на основе обратной связи от реальной системы, претерпел изменения и в данной работе отвечает за взаимодействие с

внешним источником [20, 35]. Поэтому в составе цифрового двойника настоящей работы реализуется только вспомогательный и имитационный каналы (рисунок 1.4).

Модифицированный цифровой двойник построен как последовательность вычислительных каналов, каждый из которых отвечает за свой этап превращения исходных данных в итоговый почасовой баланс. Выглядит этот конвейер так:

1. Метеорологический блок (ETL, вспомогательный канал). Извлечение, нормализация и сохранение временных рядов облачности и температуры. На выходе – очищенная и проверенная на дубли таблица *screenshots* с локальными и UTC-метками.

2. Астрономический и атмосферный блок (часть имитационного канала). Чисто теоретический блок: по юлианской дате, долготе и широте вычисляются зенитный угол, деклинация и внеатмосферный поток. Результат – голая солнечная геометрия без влияния погоды. Далее астрономия встречается с метеоданными, а внеатмосферный поток умножается на оптическую толщину атмосферы, корректируется температурой ячейки и, главное, ослабляется облачностью. На выходе – искомый *solar_flux* (мгновенный солнечный поток у поверхности).

3. Конфигурационный канал (часть имитационного канала). На основе профиля нагрузки, параметров модулей и геометрии выбранной крыши решается задача компоновки: плотная раскладка панелей, подбор длины строк, числа инверторов и аккумуляторных ветвей. На выходе – полностью согласованная конфигурация *system_config* (PV-поле + инверторы + АКБ).

4. Балансовый канал (часть имитационного канала). Динамическое ядро двойника. Час за часом сопоставляет *solar_flux* и нагрузку, направляет избыток в батарею или, наоборот, забирает из неё недостающее, а при глубоком разряде подключает внешнюю сеть. Итог – суточный профиль покрытия, заряда, разряда, дефицита и энергии, взятой из сети.

При этом для повышения адекватности модели вместо усреднённых климатических параметров используются реальные максимально приближенные к

реальным метеорологическим данным, получаемые из открытого программного интерфейса Open-Meteo [13, 36].

Open-Meteo представляет собой агрегатор численных моделей прогноза и реанализа (ECMWF/IFS, GFS, ICON и др.), предоставляющий исторические и прогнозные данные по запросу в формате API.

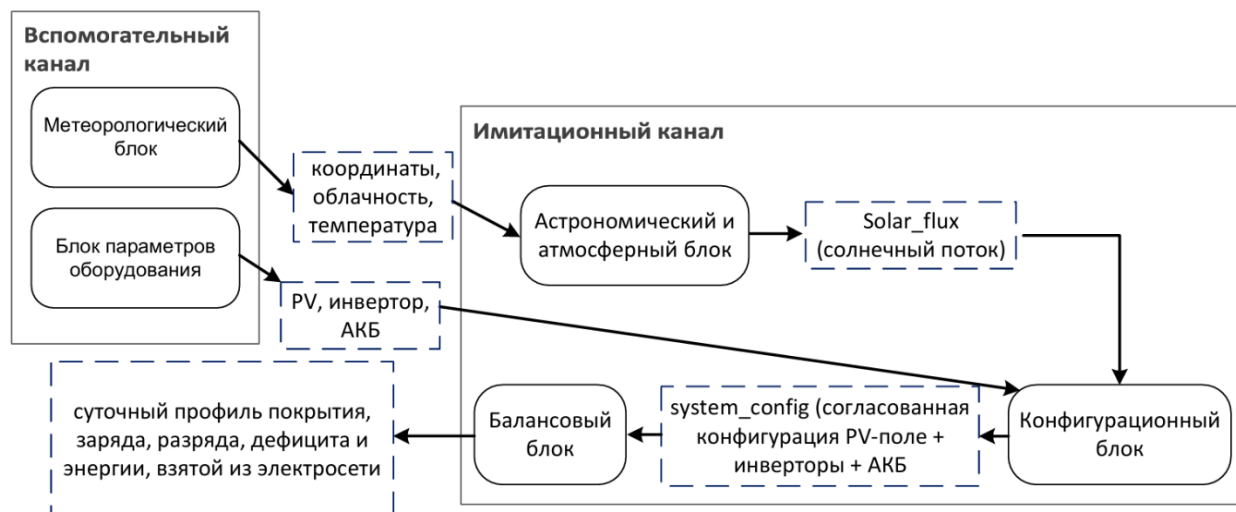


Рисунок 1.4 – Каналы ЦД ИГЭС

Для моделирования солнечной электростанции много переменных не надо – хватило двух ключевых. Первое, на что смотришь, – облачность. Самая обычная, общая облачность (total cloud cover). Это доля неба, закрытая облаками, в процентах или десятых долях. Почему она важна? Да потому что от неё напрямую зависит, сколько прямой и рассеянной радиации дойдёт до панелей. Плотная облачность в пару баллов – и выработка резко просаживается. Тонкая дымка – картина иная. Без этого параметра модель слепая.

Второй параметр – температура воздуха на высоте двух метров. Стандартная метеорологическая высота, не какая-то экзотика. Она нужна не сама по себе, а чтобы потом рассчитать температуру самого фотоэлектрического модуля. Тут в ход идёт модель NOCT – Nominal Operating Cell Temperature. Алгоритм старый, проверенный: берёт температуру воздуха, добавляет нагрев от солнечного излучения и получает, насколько греется ячейка. А от температуры ячейки уже пляшет КПД панели – чем выше нагрев, тем ниже эффективность. Не учёшь этого – и цифры выработки поплывут, особенно в солнечные, но

жаркие дни. Так что температура воздуха – входной билет к нормальному учёту температурной зависимости.

Данные для всего этого добра беру из Open-Meteo. Пространственное разрешение там не везде одинаковое – зависит от модели. Для исторических данных, из реанализов, сетка обычно порядка 9×9 км. Это примерно 0,1° по широте и долготе. Сервер, когда получает запрос на конкретную точку, не просто берёт значение из ближайшего узла – он динамически интерполирует. Скорее всего, билинейной интерполяцией или методом ближайшего соседа – под капот не заглядывал, но по документации похоже. Так мы получаем число, которое представляет среднее для всей ячейки размером девять на девять километров.

Можно, конечно, испугаться: объект-то точечный, горнолыжная база, а тут усреднение по огромному квадрату. Но на деле для наших задач этого разрешения – за глаза. Облачность и температура воздуха внутри девятикилометровой ячейки не скажут настолько, чтобы сильно исказить месячные и годовые энергобалансы. Микроклиматические нюансы на склоне, допустим, локальный туман или холодный карман в низине, они, безусловно, есть, но при расчётах, где мы оперируем киловатт-часами в месяц и год, такие тонкости погоды не делают. Погрешность будет, но не критичная. Поэтому смело опираемся на эти две переменные – облачность и температуру, – и стрим на них цифровую модель.

Выбор Open-Meteo в качестве источника входных данных обоснован следующими факторами:

- свободный доступ и отсутствие лицензионных ограничений для академических исследований;
- возможность программного получения временных рядов с заданным шагом (например, почасовым);
- достаточная точность для задач предпроектного анализа и имитационного моделирования;
- применимость для удалённых регионов, включая Дальний Восток.

Таким образом, разрабатываемый в настоящей работе цифровой двойник представляет собой автономную имитационную модель, входные параметры которой (облачность, температура) формируются на основе реальных метеоданных из внешнего API, а выходные (выработка электроэнергии солнечными панелями, уровень заряда аккумуляторов, доля потребления, покрываемая за счёт солнечной генерации) вычисляются с использованием физических моделей фотоэлектрического преобразования, накопления энергии и работы гибридного инвертора.

Вместе с тем архитектура цифрового двойника спроектирована с учётом возможности последующей модернизации. При условии развёртывания реальной ИГЭС на горнолыжной базе «Холдоми» предлагаемая имитационная модель может быть расширена до полноценного цифрового двойника путём добавления аппаратного канала (подключения к контроллерам и датчикам системы) и вспомогательного канала (алгоритмов калибровки и адаптации). Это позволит перейти от пассивного имитационного моделирования к активному управлению и прогнозированию состояния энергосистемы в реальном времени, что является приоритетной целью для дальнейших исследований.

2 Выбор оборудования для составления ИГЭС

2.1 Подбор оборудования для солнечной части ИГЭС

При подборе оборудования для солнечной части изолированной гибридной энергетической системы (ИГЭС) особенно важно ориентироваться не только на технические параметры панелей, но и на юридическую сторону закупки. Для дорогостоящего и длительно эксплуатируемого оборудования целесообразно работать с официально зарегистрированными в Российской Федерации компаниями и уполномоченными дилерами, поскольку именно они выступают понятным контрагентом, к которому можно предъявлять требования по качеству, ремонту, замене или возврату товара. В соответствии со ст. 470 ГК РФ гарантия качества действует в пределах гарантийного срока, установленного договором, а по ст. 18 Закона РФ «О защите прав потребителей» требования при обнаружении недостатков могут предъявляться продавцу, изготовителю, уполномоченной организации или импортеру. Это особенно важно для удаленных объектов, где простой оборудования ведет к заметным эксплуатационным потерям [30, 34].

Для горнолыжной базы «Холдоми» данное обстоятельство имеет принципиальное значение: объект расположен в регионе со сложными климатическими условиями, а значит, в случае отказа оборудования или несоответствия паспортным характеристикам должен быть понятный и доступный канал обращения по гарантии и сервису. Именно поэтому в дальнейшем в качестве потенциальных поставщиков рассматриваются прежде всего компании, имеющие официальное присутствие в РФ, понятные реквизиты, сервисную поддержку и возможность поставки оборудования в Хабаровский край [38].

Ниже приведен полный перечень компаний и представительств, которые были найдены в открытых источниках (таблица 2.1). Часть из них имеет прямое присутствие в Хабаровске или Хабаровском крае, часть работает через доставку

по России, а часть представляет собой федеральные бренды или дистрибьюторов, у которых можно заказать оборудование для региона.

Таблица 2.1 – Перечень компаний и представительств

№	Название	Адрес
1	Nevel Group («Хевел»)	Головной офис в Москве; производство – Новочебоксарск; продажи и поставки идут через дилерскую сеть.
2	Русолар	Владивосток, ул. Хабаровская, 22, оф. 29; в открытых источниках также встречается связанная компания «Русолар ДВ» во Владивостоке.
3	SPARES.RU («Спэйрс»)	Долгопрудный, Лихачёвский проезд, 8, офис 102; федеральная доставка по России.
4	Aurinko® («Аурилко»)	Екатеринбург, ул. Амундсена, 107, оф. 707; на сайте есть отдельный раздел «Каталог Aurinko в Хабаровске».
5	Sunways Group	Зеленоград, ул. Конструктора Гуськова, 3, стр. 6; также указан адрес для корреспонденции в Зеленограде.
6	Limit Energy	Москва, есть доставка в Хабаровский край
7	GWS-Energy	Москва, г. Троицк, ул. Центральная, 5, офис 40.
8	Компания «Одно Солнце» (one-sun.ru)	Зеленоград, ул. Конструктора Гуськова, 3, стр. 6.
9	ECO50	Москва: м. Котельники, Люберцы, ул. Юбилейная, 26; также есть офис в Краснодаре.
10	Realsolar	Санкт-Петербург, Октябрьская набережная, д. 104, корп. 1П, офис 307.
11	Солнечные технологии Дальнего Востока	Хабаровск, ул. Ленинградская, 28 ст. 4, лит. Ф, офис 216.
12	Альтернатива ДВ	В открытых источниках встречается хабаровский адрес, но компания в ряде реестров отмечена как ликвидированная; как поставщик для закупки требует дополнительной проверки.
13	Технолайн	Хабаровск, просп. 60-летия Октября, 158, офис 337; в текущих открытых источниках интернет-магазин e-solarpower привязан к ИП из Владивостока, поэтому реквизиты перед закупкой нужно перепроверить.
14	ООО «Солнечный Свет» / Solar Light	Хабаровск, просп. 60-летия Октября, 158, оф. 337; в открытых реестрах встречается одноименная организация с другим адресом, поэтому перед закупкой нужны точные реквизиты.
15	Solar В Energy / ООО «СБ Энерджи»	Официальный дилер в Хабаровске, магазин «Лампа», ул. Ким-Ю-Чена, 10.
16	ООО «Солнечный»	рп Солнечный, Хабаровский край, ул. Ленина, 18а, кв. 18, ком. 18; по открытым данным профиль деятельности не подтвержден как продажа солнечных панелей.
17	BobrovSolar («Бобров-солар»)	Хабаровск, Целинная ул., 8В; в открытых источниках у компании также указан московский офис и склад, а доставка ведется по всей России.

На первом этапе из перечня целесообразно исключить компании, по которым в открытых источниках не подтверждается профильная деятельность на рынке солнечных панелей либо имеются сомнения в актуальности реквизитов. К этой группе относятся, например, ООО «Солнечный», у которого открыто подтверждается лишь наличие юридического лица в Хабаровском крае, но не профильная специализация на солнечном оборудовании, а также «Альтернатива ДВ», которая в части реестров отмечается как ликвидированная. Для инженерного проекта это слабая база, так как для закупки и дальнейшего гарантийного сопровождения нужен действующий поставщик с понятной специализацией.

На втором этапе из списка логично убрать компании, по которым есть лишь косвенная привязка к продаже солнечных модулей, но нет ясного подтверждения ассортимента, гарантийной политики и сервисной поддержки. Сюда относятся, прежде всего, Limit Energy и часть локальных организаций, где в открытом доступе есть только общий адрес или общий профиль компании, но нет уверенно подтвержденной линейки солнечных панелей и сервисной схемы. Для объекта вроде «Холдоми» это нежелательно, поскольку монтаж будет выполняться на нескольких зданиях, а значит, важны не только поставка, но и сопровождение, замена, логистика и консультация по месту.

На третьем этапе в приоритете остаются компании, у которых одновременно подтверждаются: российская регистрация или официальный дилерский статус, доставка в Хабаровский край, наличие сервисных услуг и хотя бы частично раскрытые технические характеристики продукции. По этому признаку в основной список целесообразно оставить Nevel Group, BobrovSolar, Aurinko, Sunways Group, SPARES.RU, GWS-Energy, ECO50, Realsolar и, как локальные каналы, Солнечные технологии Дальнего Востока, Технолайн, Solar Light и Solar B Energy.

Nevel Group является технологическим ориентиром. Компания заявляет выпуск НТ-модулей, высокую эффективность, низкий температурный коэффициент, минимальную деградацию и наличие EPC/O&M-сервисов; это осо-

бенно важно для холодного климата и для объектов, где требуется стабильная годовая выработка при ограниченной площади установки. В вашем исходном материале для Nevel также отдельно указаны работа при $-40\text{ }^{\circ}\text{C}$ и испытания при $-60\text{ }^{\circ}\text{C}$, что делает такие модули особенно интересными для Хабаровского края.

BobrovSolar выглядит сильнейшим вариантом именно как канал закупки и дальнейшей организации проекта. У компании есть фактический склад и офис в Москве, указаны условия доставки по России, а так же зафиксировано представительство и шоу-рум в Хабаровске по адресу Целинная, 8В. Кроме того, на сайте BobrovSolar указана гарантия производителя и отдельные решения для баз отдыха, а в карточке типовой станции для базы отдыха приведены КПД солнечного модуля 22,16% и гарантия до 15 лет. Для проекта горнолыжной базы это важный плюс, потому что здесь нужен не только модуль, но и понятный путь от подбора оборудования до отгрузки и гарантийного обслуживания.

Aurinko также является сильным кандидатом: на официальном сайте компания указывает собственное производство и поставку солнечных решений, отдельный каталог для Хабаровска, гарантийное и сервисное обслуживание, а на карточках модулей заявляет рабочий диапазон температур от -60 до $+90\text{ }^{\circ}\text{C}$. Это делает Aurinko очень интересным вариантом для северо-восточных условий, особенно если приоритетом является именно климатическая устойчивость оборудования. Ограничение здесь другое: офис компании расположен в Екатеринбурге, то есть сервисная доступность на месте уступает локальным хабаровским поставщикам.

Sunways Group, SPARES.RU, GWS-Energy, ECO50 и Realsolar также остаются достойными вариантами, но у них несколько слабее региональная привязка к Хабаровску. Sunways имеет официальный офис в Зеленограде и заявляет производство и поставку солнечного оборудования, гарантию на модули 12 лет и сервисные направления; SPARES.RU выступает как крупный дистрибьютор с прямыми поставками и сервисом, включая официальное партнерство по ряду солнечных брендов; GWS-Energy, ECO50 и Realsolar обладают полно-

ценной инфраструктурой продаж и доставки, но находятся в Москве, Краснодаре или Санкт-Петербурге, а значит, для объекта в Хабаровском крае их сильная сторона – логистика и ассортимент, а не локальный сервис.

Локальные хабаровские организации – «Солнечные технологии Дальнего Востока», Технолайн, ООО «Солнечный Свет» и Solar В Energy – полезны именно как региональные каналы, однако по части открыто опубликованных технических характеристик и гарантий они уступают федеральным производителям и крупным дистрибьюторам. Их рационально рассматривать как вспомогательные варианты для сравнительных ценовых запросов, но не как единственную основу закупки для ответственного энергетического объекта.

Если исходить именно из задачи закупки солнечных панелей для горнолыжной базы «Холдоми», где важны и климатическая надежность, и гарантийная защищенность, и возможность реально решать вопросы на месте, наиболее сбалансированным вариантом выглядит BobrovSolar. Он сочетает наличие хабаровского представительства, официальный российский контур поставки, доставку по стране, работу с гарантийными случаями и доступ к линейкам крупных производителей. В инженерном смысле это особенно удобно: можно брать модульный ряд премиум-класса, а при необходимости – опираться на локальную логистику и сервис.

При этом Nevel следует оставить как технологический ориентир для сравнения, особенно если акцент делать на холодоустойчивость, низкую деградацию и долгий срок службы. Поэтому практический вывод для проекта можно сформулировать так: в качестве основного канала закупки целесообразно рассматривать BobrovSolar, а в качестве эталонной линейки солнечных модулей – Nevel. Такой подход одновременно учитывает и инженерные, и организационные требования проекта, что особенно важно при создании робастного цифрового двойника изолированной гибридной энергетической системы для удаленного объекта.

Так как выбор поставщика сделан, следом необходимо выбрать модель солнечной панели, характеристики которой будут использованы в цифровом

двойнике ИГЭС. В каталоге компании BobrovSolar есть линейки Jinko Solar, LONGi Solar и Trina Solar, то есть можно подобрать модуль под разные компромиссы между эффективностью, ценой и сроком службы.

Из этой группы для горнолыжной базы «Холдоми» наиболее логично смотреть на модули, которые рассчитаны на суровые условия и дают высокий КПД при минимальной деградации. У LONGi Hi-MO 5 заявлены 144 ячейки, формат 182 мм, КПД 21,3%, температурный коэффициент $-0,35\%/^{\circ}\text{C}$, NOCT $45\pm 2^{\circ}\text{C}$ и рабочий диапазон $-40\dots+85^{\circ}\text{C}$; у Trina Vertex S+ 460W – 144 ячейки, КПД 23%, температурный коэффициент $-0,29\%/^{\circ}\text{C}$, NMOT $43\pm 2^{\circ}\text{C}$, диапазон $-40\dots+85^{\circ}\text{C}$, гарантия 15 лет при годовой деградации 0,40%; у ASTRO 580W-M из каталога BobrovSolar – 144 ячейки, КПД 22,5%, температурный коэффициент $-0,29\%/^{\circ}\text{C}$ и NOCT $41\pm 2^{\circ}\text{C}$. Для Jinko Tiger Neo 580W официальная документация показывает 144 ячейки, КПД 22,45%, температурный коэффициент $-0,29\%/^{\circ}\text{C}$, NOCT $45\pm 2^{\circ}\text{C}$, максимальное напряжение системы 1000/1500 VDC и длительную линейную гарантию на 15 лет при годовой деградации 0,43%.

С учетом вышеперечисленных критериев, логичнее выбрать именно Trina Solar Vertex S+ TSM-NEG9R.28 (460W) как основную панель для дальнейшего цифрового двойника. Она почти не проигрывает Jinko по NOCT, но выигрывает по долгосрочной надежности за счет меньшей деградации, а также хорошо подходит для холодного климата за счет устойчивого температурного поведения и высокого допустимого системного напряжения. Для объекта с сезонной нагрузкой и сложными погодными условиями это более сильный вариант, чем бюджетные или менее долговечные решения.

В таблице 2.2 представлены параметры выбранной модели для дальнейшего моделирования.

Таблица 2.2 – Таблица параметров выбранной панели

Параметр	Значение
Модель	Solar Vertex S+ TSM-NEG9R.28 (460W)
Ширина ячейки	210 мм

Продолжение таблицы 2.2

Высота ячейки	210 мм,
Количество ячеек	144 (6×24)
Практический КПД	23%
Коэффициент температуры	-0,29%/°C
НОСТ	43°C
Максимальное напряжение системы	1500 VDC (IEC)

После анализа параметров выбранной панели можно сделать вывод, что именно такие характеристики наиболее важны для последующего построения цифрового двойника солнечной части гибридной энергетической системы. В дальнейшем указанные значения будут использоваться при расчёте выработки электроэнергии, учёте температурного влияния на эффективность модуля, а также при моделировании режимов работы системы в условиях изменяющейся внешней среды.

2.2 Подбор оборудования для аккумуляторной части ИГЭС

После подбора солнечных панелей следующим этапом проектирования изолированной гибридной энергетической системы для горнолыжной базы «Холдоми» является выбор накопителя энергии. Для такого объекта целесообразно рассматривать именно модульные LiFePO₄-аккумуляторные системы rack-формата, поскольку в исходных материалах проекта задана ориентировочная ёмкость накопителя более 200 кВт·ч, а значит требуется решение, которое можно наращивать по модульному принципу, размещать в стойках и интегрировать в утеплённый контейнер с инженерной обвязкой. LiFePO₄ в данном случае выбран не случайно: это безопасная и долговечная технология, пригодная для стационарных систем хранения энергии и лучше подходящая для северных условий, чем более простые свинцово-кислотные варианты [22].

Отдельно важно, что для данного проекта следует опираться только на официально зарегистрированные в Российской Федерации компании и представительства. Это снижает риски при покупке дорогостоящего оборудования:

упрощает заключение договора, получение счёта и закрывающих документов, обеспечивает понятный порядок гарантийного обслуживания, а также позволяет рассчитывать на претензионную работу и сервисную поддержку внутри страны. Для объекта уровня «Холдоми», где от надёжности системы зависит автономность всей базы, этот критерий является принципиальным. Поэтому в данном анализе учитывались только те поставщики, которые имеют официальный статус в РФ и/или обеспечивают поставку на российский рынок.

Ниже приведён перечень компаний, рассмотренных как потенциальные поставщики LiFePO₄-аккумуляторов для проекта. В таблице 2.3 отдельно отмечены адреса компаний.

Таблица 2.3 – Перечень компаний и представительств

№	Название	Адрес
1	ООО «Титанат»	г. Дубна, ул. Дружбы, 19
2	FelicityESS Russia (ООО «Вайдвелл»)	г. Санкт-Петербург, ул. 3-я Советская, 7А, кв. 14
3	SPARKCELL (ООО «Спаркселл»)	г. Москва, Большая Семёновская, 40с18
4	«Лиферыч» (ООО АЗ «Лиферыч»)	г. Челябинск, Набережная ул., 11, 3 этаж
5	ООО «ЛИКОМ»	г. Новосибирск, ул. Зелёная горка, 1
6	BestAccum	г. Москва, б-р Яна Райниса, 2к1
7	AltPower	г. Москва, ул. Металлургов, 60, стр. 1
8	БобровСолар	г. Хабаровск, ул. Целинная, 8В
9	ООО «Солнечная элек- тростанция»	г. Хабаровск, пер. Дежнёва, 18А, оф. 103
10	ООО «ТехноЛайн»	г. Хабаровск, Индустриальная ул., 1Б

Если перейти от общего российского рынка к задаче снабжения объекта в Хабаровском крае, список кандидатов необходимо сузить до тех компаний, которые либо уже присутствуют в регионе, либо готовы обеспечивать доставку без критических логистических потерь. Для горнолыжной базы это особенно важно: оборудование большой ёмкости, близкое к 200 кВт·ч, имеет значительную массу и требует не только надёжной упаковки, но и предсказуемой доставки, монтажа и последующего обслуживания. Поэтому первыми из рассмотре-

ния логично оставлять компании с прямым присутствием в Хабаровске – «ТехноЛайн», «БобровСолар» и «Солнечная электростанция».

Следующий круг отбора – это компании, которые не базируются в Хабаровском крае, но по материалам официальных сайтов компаний способны работать с регионом через доставку. К ним относятся SPARKCELL, «Лиферыч», AltPower и BestAccum. Такой вариант уже менее удобен по логистике, однако сохраняет доступ к сильным техническим сторонам: у SPARKCELL прямо заявлен рабочий диапазон до $-25\text{ }^{\circ}\text{C}$, у AltPower отмечены расширенная гарантия и температурное хранение при $-20\dots+45\text{ }^{\circ}\text{C}$, а у «Лиферыч» – пять лет гарантии. Для удалённого объекта это важные плюсы, но они не компенсируют полностью отсутствие местного склада и местного сервиса.

Ещё более ограниченным становится выбор, если дополнительно учитывать морозостойкость и пригодность к длительной автономной работе. В этом отношении слабее выглядят «ЛИКОМ» и часть магазинов, у которых не указаны специальные холодоустойчивые исполнения. Для проекта ИГЭС это нежелательно, потому что батарейный блок планируется размещать в контейнере и использовать в зимних условиях, а значит нужен либо продукт с подтверждённой работой на морозе, либо производитель, готовый обеспечить инженерную дооснастку системой терморегулирования.

С точки зрения именно объекта «Холдоми» наиболее выигрышным выглядит не просто тот поставщик, у которого лучше гарантия, а тот, который сочетает три условия: территориальную близость к Хабаровскому краю, возможность быстрой поставки и дальнейшей поддержки, а также достаточную технологическую компетенцию в области LiFePO_4 -накопителей. По этим критериям локальные компании оказываются сильнее федеральных интернет-магазинов, потому что для батарейной системы такого масштаба важны не только характеристики самих модулей, но и сопровождение проекта на месте.

Наиболее практичным вариантом из хабаровских поставщиков выглядит ООО «ТехноЛайн» с адресом Индустриальная ул., 1Б. Его преимущество состоит не в рекордной гарантии или прямо заявленной морозостойкости, а в том,

что компания уже присутствует в регионе, работает как поставщик оборудования электроснабжения и может выступать связующим звеном между подбором батарей, доставкой, консультацией и последующим сервисом. Для автономной энергосистемы горнолыжной базы это особенно важно, потому что любое удалённое обслуживание батарейного массива в зимний период связано с дополнительными рисками и затратами. Именно поэтому местный поставщик с инженерной поддержкой оказывается предпочтительнее, чем более «красивый» по гарантии, но удалённый игрок рынка.

Второе место по практической пригодности можно отдать «БобровСоляр»: наличие склада в Хабаровске и диапазон гарантии 2-5 лет делают его удобным для оперативной закупки и замены компонентов. Однако в представленных материалах компании морозостойкость моделей отдельно не подтверждена, поэтому для климатических условий Хабаровского края этот вариант требует дополнительной технической проверки перед окончательным заказом.

Далее следует SPARKCELL как сильный производитель с заявленной работоспособностью до -25°C и двухлетней гарантией. Для проекта, где критично выдерживать низкие температуры, это серьёзный плюс. Однако компания базируется в Москве, а значит поставка в Хабаровский край потребует более сложной логистики и, вероятно, больше времени на согласование партии. По совокупности факторов SPARKCELL стоит рассматривать как технически сильного, но не самого удобного регионально поставщика.

FelicityESS Russia и «Лиферыч» выглядят очень убедительно по гарантийным условиям: у FelicityESS заявлена десятилетняя гарантия, а у «Лиферыч» – пять лет. Но для Хабаровского края их слабое место заключается в удалённости от объекта и отсутствии столь же удобного местного присутствия, как у хабаровских поставщиков. Поэтому эти компании рационально оставлять в резерве как технологически надёжные, но не первоочередные для закупки для «Холдоми».

AltPower и BestAccum можно рассматривать как запасные варианты для расширения рынка предложений. AltPower интересен расширенной гарантией и

заявленным режимом хранения $-20...+45$ °С, а BestAccum – возможностью сборки LiFePO₄-систем на заказ. Однако оба поставщика удалены от объекта и не дают такого преимущества по региональному сопровождению, как хабаровские компании. Для большого стационарного накопителя это заметный минус.

Таким образом, для закупки аккумуляторов под изолированную гибридную энергосистему горнолыжной базы «Холдоми» наиболее обоснованно использовать модульные LiFePO₄ rack-системы, поскольку они лучше соответствуют задаче накопления порядка 200 кВт·ч, удобны для размещения в стойках и подходят для поэтапного наращивания ёмкости. Среди поставщиков, рассмотренных в данном разделе, наибольшую практическую ценность для Хабаровского края имеет ООО «ТехноЛайн» как местный поставщик с офисом в Хабаровске и возможностью инженерного сопровождения. В качестве альтернатив и резервных вариантов можно оставить «БобровСолар», SPARKCELL и «Лиферыч», однако их выбор требует более тщательной проверки по логистике, морозостойкости и условиям интеграции в проект.

Для дальнейшего проектирования цифрового двойника целесообразно не ограничиваться самим фактом выбора поставщика, а перейти к отбору конкретного типа аккумуляторного модуля, пригодного для последующей компоновки батарейного массива. В рамках выбранного ранее ООО «ТехноЛайн» как лучшего регионального варианта для проекта «Холдоми» наиболее логично рассматривать не отдельные бытовые АКБ, а rack-LiFePO₄-модули промышленного класса: именно такой формат лучше всего соответствует задаче поэтапного набора большой ёмкости, удобен для размещения в стойках и позволяет масштабировать накопитель до требуемого уровня около 200 кВт·ч. В самом файле у «ТехноЛайна» прямо отмечена работа с различными брендами LiFePO₄, включая MUST и DEYE, а также ориентация на комплексные решения электроснабжения, что делает его подходящей площадкой для подобного подбора.

Если сузить выбор до нескольких наиболее подходящих вариантов в рамках этого поставщика, то для цифрового двойника разумно оставить три типоразмерных решения: rack-модуль LiFePO₄ 51,2 В 100 А·ч как базовую единицу

для масштабирования, rack-модуль 51,2 В 200 А·ч как более ёмкий вариант с меньшим числом соединений и комплектную шкафную сборку на их основе как вариант для более позднего этапа закупки. Однако в качестве единой расчетной модели для дальнейших энергетических вычислений лучше взять именно модуль 51,2 В 100 А·ч: он наиболее универсален для построения массива большой мощности, удобен для резервирования и легче всего масштабируется в стойку без потери гибкости конфигурации. В таблице 2.4 представлены параметры выбранной модели аккумулятора.

Таблица 2.4 – Таблица параметров выбранного аккумулятора

Параметр	Значение
Модель	Rack LiFePO ₄ -модуль 51,2 В 100 А·ч
Номинальная ёмкость АКБ	100 А·ч
Максимально допустимая глубина разряда	10%
Коэффициент режима работы АКБ	0.8
Коэффициент максимального тока зарядки АКБ	0.3
Напряжение заряда АКБ	57.6 В

Такой выбор удобен для последующих расчетов в цифровом двойнике, потому что позволяет описывать аккумуляторную часть системы как модульную сборку, а не как единичный аккумулятор ограниченной мощности. Для объекта уровня горнолыжной базы это принципиально: в дальнейшем можно будет собрать батарейное поле из одинаковых модулей, рассчитать требуемое число стоек и затем увязать накопитель с солнечной генерацией и режимом нагрузки требуемого объекта.

2.3 Подбор оборудования для инверторной части ИГЭС

На предыдущем этапе проектирования для солнечной электростанции была уже выбрана модель солнечной панели Solar Vertex S+ TSM-NEG9R.28 (460W). В связи с этим дальнейший выбор оборудования был сосредоточен на подборе инвертора, способного работать в составе изолированной гибридной

системы и обеспечивать устойчивое электроснабжение объекта в условиях Хабаровского края. Для таких проектов принципиально важно использовать не случайных продавцов, а именно официально зарегистрированные в Российской Федерации компании и представительства. Это обусловлено тем, что для сложного энергетического оборудования особенно значимы юридическая прозрачность сделки, наличие официальной гарантии, возможность сервисного обслуживания, а также предсказуемость поставки и последующего сопровождения. Для удалённого объекта, такого как горнолыжная база «Холдоми» вблизи Комсомольска-на-Амуре, данные факторы становятся определяющими, поскольку сбой в логистике или отсутствие гарантийной поддержки напрямую повышают риски реализации всего энергетического комплекса.

В рамках анализа были рассмотрены компании, имеющие официальные адреса, работающие на российском рынке и предлагающие инверторы для солнечных электростанций. Итоговый перечень представлен в таблице 2.5.

Таблица 2.5 – Перечень компаний и представительств

№	Название	Адрес
1	ООО «Альтэко» (официальный дистрибьютор GoodWe)	г. Москва, ул. Дубки, д. 2
2	ООО «Умная Энергия» (Clever Energy)	г. Краснодар, ул. Уральская, 114, литер Г
3	ООО «Технолайн» (Интернет-магазин «Солнечный Дом»)	г. Владивосток, ул. Выселковая, д. 48, стр. 10
4	ООО «Энергия Про» (магазин «Бобровсолар»)	г. Хабаровск, ул. Целинная, д. 8В
5	ООО «Солнечный Дом» (SolarHome, официальная продажа SofarSolar)	г. Москва, ул. 10-я Парковая, д. 18-26
6	ООО «Экопроект-Энерго» (официальный дистрибьютор Sofar)	г. Краснодар, ул. Почтовая, д. 223/1
7	ООО «Теслум» (поставщик Fronius и др.)	г. Нижний Новгород, б-р Мещерский, д. 3, корп. 1
8	ООО «Просолар» (оптовый поставщик)	г. Москва, ул. 10-я Парковая, д. 19

Из приведённого перечня видно, что рынок поставщиков инверторного оборудования в России достаточно широк и включает как официальных дистрибьюторов конкретных брендов, так и крупные мультибрендовые компании. Это важно, поскольку при проектировании гибридной автономной системы вы-

бор не должен ограничиваться только ценой: не менее значимыми становятся наличие официального статуса, возможность гарантийного обслуживания и удобство логистики. В представленном списке особенно выделяются компании, работающие с оборудованием GoodWe, SofarSolar, Fronius и другими популярными производителями, поскольку именно эти марки традиционно используются в гибридных и автономных системах электроснабжения.

После формирования общего списка поставщиков был выполнен его последовательный сужающий отбор с учётом географии объекта. Для базы «Холдоми» наиболее важным критерием становится не только наличие подходящей модели инвертора, но и возможность быстрой доставки, наличия складских запасов и оперативной сервисной поддержки в Дальневосточном регионе. По этой причине в первую очередь были рассмотрены компании, имеющие представительства либо фактическую логистическую близость к Хабаровскому краю. К данной группе относятся ООО «Энергия Про» в Хабаровске и ООО «Технолайн» во Владивостоке. Остальные компании, расположенные в Москве, Краснодаре, Нижнем Новгороде и других центральных регионах, а они при всей надёжности, объективно менее удобны для снабжения удалённого объекта на Дальнем Востоке, поскольку доставка от них занимает больше времени и влечёт дополнительные логистические затраты.

Если перебирать дальневосточных поставщиков, ООО «Технолайн» сразу бросается в глаза – Владивосток, инверторы Sofar, пятилетняя гарантия на сетевые модели. Вроде достойно. Но когда прикидываешь логистику до Хабаровского края, всё уже не так радужно. Владивосток – это километры, пробки на трассе, да и вообще лишнее звено. Доставка затянется, а в пик сезона простой чреват убытками. Хочется кого-то прямо на месте. И тут ООО «Энергия Про» – сидят в Хабаровске. Ребята, с которыми можно пересечься лично, а не слать бесконечные запросы в другой часовой пояс. Для нашего объекта это решает сразу несколько проблем: сроки поставки сжимаются до минимума, транспортные расходы падают, а гарантийное и постгарантийное общение идёт лицом к лицу, без бесконечных звонков и ожиданий. Можно заглянуть на их сайт –

честно говорят: работают с автономным электроснабжением, возят гибридные и сетевые инверторы разных брендов, гарантия на стандартных условиях производителя. Никакой самодеятельности, всё прозрачно. Откликнулись быстро, каталог прислали без задержек – это тоже показатель.

«Технолайн» с их Sofar и пятилетней гарантией неплох, но, когда начинаешь считать стоимость доставки и время реакции, преимущества хабаровского поставщика становятся видны невооружённым глазом. Мало того, что склад рядом, так ещё и менеджеры на связи без задержек, а это дорогого стоит, когда зима и горнолыжный сезон в разгаре. По сути, для базы «Холдоми» выбор очевиден. «Энергия Про» даёт ровно то, что нужно: они в Хабаровске, заточены на автономные и гибридные системы, инверторов разных хватает, гарантия приемлемая. А для удалённой, климатически суровой точки такой баланс – надёжность, скорость, доступность сервиса – перевешивает всё. Поэтому без колебаний берём именно их как предпочтительного поставщика инверторного оборудования для цифрового двойника и изолированной гибридной системы базы «Холдоми».

Дальнейшее уточнение состава оборудования логично выполнить уже на уровне конкретных моделей инверторов, предлагаемых выбранным поставщиком. Для объекта типа горнолыжной базы наиболее рационально ориентироваться не на бытовые сетевые решения, а на гибридные инверторы, которые могут работать с солнечными панелями, аккумуляторным блоком и резервным источником питания. В каталоге ООО «Энергия Про» (БобровСолар) в Хабаровске представлены как раз такие решения: в структуре ассортимента выделены разделы «гибридные», «автономные», «трехфазные», а также подборки по мощности, включая 12, 15 и 25 кВт, при этом на сайте отдельно есть категории «для базы отдыха» и «промышленные», что подтверждает ориентацию на объекты с автономным или полуавтономным энергоснабжением [22].

С практической точки зрения для дальнейшего моделирования цифрового двойника можно выделить три наиболее подходящих варианта из ассортимента этого поставщика: Deye SUN-12K-SG05LP3-EU-SM2, Deye SUN-15K-SG05LP3-

EU-SM2 и Deye SUN-25K-SG01HP3-EU-AM2. Модель на 25 кВт является трехфазным гибридным решением с батарейным напряжением 160–700 В, двумя MPPT-контроллерами, поддержкой подмешивания в сеть, возможностью работы с аккумуляторами AGM/GEL/Литий-железно-фосфатного (LiFePO4) типа и диапазоном PV-входа до 40 кВт; модели на 12 и 15 кВт являются однофазными. У 12 кВт версии имеется ток заряда 240 А, а 15 кВт – 280 А и более высокий запас по мощности для локальных нагрузок. Однако для объекта уровня горнолыжной базы более перспективной выглядит трехфазная модель Deye SUN-25K-SG01HP3-EU-AM2: она рассчитана на 25 кВт, поддерживает максимальное напряжение аккумуляторов 700 В, два MPPT, PV-массив до 40 кВт, ток заряда/разряда до 50 А, работу с генератором, удалённый мониторинг по Wi-Fi и параллельное объединение до 10 инверторов, что особенно важно для масштабируемой изолированной системы.

С учётом того, что объект «Холдоми» представляет собой не одиночный дом, а инфраструктурную площадку с повышенными требованиями к надёжности, резервированию и возможности наращивания мощности, в качестве базовой модели для цифрового двойника целесообразно принять именно Deye SUN-25K-SG01HP3-EU-AM2. Этот выбор наиболее удачен по трём причинам: во-первых, трёхфазная архитектура лучше соответствует объекту с распределённой нагрузкой; во-вторых, запас по фотоэлектрической мощности и высокий ток заряда дают пространство для настройки сценариев работы в цифровом двойнике; в-третьих, модель поддерживает как автономный режим, так и работу с сетью и генератором, что делает её более универсальной для реальной гибридной энергетической схемы. Гарантийно поставщик работает по паспорту производителя, а для Deye у продукции BobrovSolar заявлен срок гарантии 5 лет. В таблице 2.6 представлены параметры выбранной модели инвертора.

Таблица 2.6 – Таблица параметров выбранного инвертора

Параметр	Значение
Модель	Deye SUN-25K-SG01HP3-EU-AM2
Номинальная мощность	25 кВт

Продолжение таблицы 2.6

Фазность	3 фазы
Напряжение аккумуляторов	160–700 В
Мощность солнечных батарей	40 кВт
Количество МРРТ-контроллеров	2
Диапазон МРРТ	150–850 В
Максимальное входное напряжение от СБ	1000 В
Ток заряда/разряда	50 А
Подмешивание в сеть	Да
Форма выходного сигнала	Чистый синус
Максимальная эффективность	97,6%
Степень защиты	IP65
Температура эксплуатации	от –40 °С до +60 °С
Параллельная работа	До 10 инверторов
Интерфейсы связи	RS485 / WIFI / CAN

Важно помнить, что данная модель инвертора совместима с моделями солнечной панели и аккумулятора, но с двумя важными оговорками. Инвертор Deye SUN-25K-SG01HP3-EU-AM2 рассчитан на литий-ионные и свинцово-кислотные батареи с диапазоном 40–60 В, а в работу был выбран аккумулятор 51,2 В – это попадает в допустимый диапазон. При этом у инвертора указан максимальный ток заряда/разряда 240 А, поэтому для цифрового двойника и для реальной установки нужно проверить, выдерживает ли BMS Rack LiFePO₄-модуля такой ток; для одного модуля 51,2 В 100 А·ч это особенно важно.

По солнечной панели тоже совместимость есть, но не в одиночку: у Solar Vertex S+ TSM-NEG9R.28 (460W) параметры составляют $V_{mp} = 45,40$ В, $V_{oc} = 51,02$ В, $I_{mp} = 10,14$ А, $I_{sc} = 10,81$ А. У инвертора Deye диапазон МРРТ 150–850 В, а стартовое напряжение – 160 В, значит, одна панель напрямую не запустит инвертор; минимально нужен ряд из 4 панелей последовательно ($4 \times 45,40 = 180,60$ В), чтобы выйти в рабочую зону МРРТ. При этом максимальное входное напряжение инвертора – 850 В, а температура на V_{oc} у панели имеет коэффициент $-0,23\%/^{\circ}\text{C}$, поэтому точное число модулей в строке лучше окончательно считать по минимальной температуре на площадке [18].

2.4 Подбор оборудования для контроллерной части ИГЭС

На предыдущих этапах проектирования для изолированной гибридной энергетической системы горнолыжной базы «Холдоми» были выбраны солнечные панели Solar Vertex S+ TSM-NEG9R.28 (460W), аккумуляторные модули Rack LiFePO₄ 51,2 В 100 А·ч и гибридный инвертор Deye SUN-25K-SG01HP3-EU-AM2. Но когда начинаешь разбирать архитектуру системы, всплывает простой вопрос: а нужны ли вообще внешние контроллеры заряда, если инвертор и так имеет встроенные MPPT-трекеры?

Смотрим документацию на Deye SUN-25K-SG01HP3-EU-AM2. Внутри – два MPPT-трекера с диапазоном 150–850 В, максимальное входное напряжение от панелей – 1000 В. Этого с головой хватает, чтобы напрямую подцепить последовательные цепочки наших модулей Solar Vertex S+ TSM-NEG9R.28 (460 Вт). Для цифрового двойника, на стадии моделирования, большего и не требуется. Однако всё же стоит прикидывать вариант с внешними MPPT-контроллерами. Не потому, что сейчас без них никак – для чисто теоретических задач работы они избыточны. Просто если завтра мы решим перенести виртуальную модель на реальную площадку «Холдоми», хорошо бы уже иметь на руках внятные рекомендации. Чтобы не метаться в спешке, а понимать, в каких случаях без внешних контроллеров не обойтись. Такой перспективный взгляд обусловлен следующими факторами, которые могут приобрести решающее значение при практическом развёртывании оборудования в удалённом и климатически сложном регионе:

1. Резервирование и отказоустойчивость – при физической эксплуатации выход из строя единственного инвертора парализует заряд аккумуляторов. Внешние контроллеры могли бы обеспечить альтернативный канал заряда, повышая общую надёжность системы.

2. Масштабируемость – в процессе реального проектирования мощность солнечного поля может быть увеличена сверх возможностей двух встроенных

MPPT-трекеров. Внешние контроллеры позволяют гибко наращивать генерацию без замены инвертора.

3. Оптимизация в сложных условиях – при частичном затенении, неоднородной ориентации скатов крыш или снежных наносах распределение панелей по нескольким независимым внешним контроллерам может дать выигрыш в годовой выработке.

4. Унификация и ремонтпригодность – на удалённом объекте наличие резервного контроллера, совместимого с другими потенциальными источниками (например, дизель-генератором), упрощает логистику запасных частей.

И хотя для создания робастного цифрового двойника и проведения теоретических расчётов энергобаланса встроенных MPPT-контроллеров инвертора Deye SUN-25K-SG01HP3-EU-AM2 вполне достаточно, выполненный в данном разделе анализ внешних контроллеров носит характер проактивного инженерного прогноза. Его результат позволит в будущем, при переходе к стадии рабочего проектирования и монтажа, оперативно выбрать оптимальную конфигурацию системы управления зарядом без необходимости повторного исследования рынка и технических параметров.

Для подбора контроллера необходимо учитывать параметры уже выбранного оборудования. Солнечная панель Solar Vertex S+ TSM-NEG9R.28 (460W) имеет напряжение холостого хода $V_{oc} = 53,80$ В и напряжение в точке максимальной мощности $V_{mp} = 45,40$ В. Для выхода в рабочую зону MPPT инвертора требуется минимум 4 панели последовательно ($4 \times 45,40 = 180,60$ В), что даёт суммарное напряжение холостого хода около 204 В. Аккумуляторная система построена на модулях 51,2 В (48-вольтовая шина). Исходя из этого, внешние контроллеры должны поддерживать:

- номинальное напряжение аккумуляторной батареи 48 В;
- входное напряжение от солнечных панелей не менее 200 В с запасом на низкотемпературный рост V_{oc} (коэффициент $-0,23\%$ °C);
- ток заряда, соответствующий мощности подключаемого массива панелей.

Ниже в таблице 2.7 представлен перечень потенциально подходящих MPPT-контроллеров, найденных в открытых источниках, с указанием страны-производителя и поставщиков, имеющих официальное присутствие на российском рынке.

Таблица 2.7 – Перечень потенциальных MPPT-контроллеров для ИГЭС

№	Модель	Страна-производитель	Поставщик в РФ
1	Victron Energy SmartSolar MPPT RS 450/100	Нидерланды	SolarHome (Москва)
2	Victron Energy SmartSolar MPPT 150/70–250/100	Нидерланды	SolarHome (Москва)
3	Epever Tracer 10420AN (100A, 200B)	Китай	SolarHome (Москва)
4	Epever Tracer 10415AN (100A, 150B)	Китай	SolarHome (Москва)
5	MUST PC18-10015F MPPT 100A	Китай	Realsolar (Москва)
6	Delta SmartWatt MPPT 4860 (60A)	Россия	BobrovSolar, Realsolar
7	SRNE SR-MC48100 MPPT 100A	Китай	SolarBreeze, SolarHome
8	SRNE SR-ML4860 MPPT 60A	Китай	BobrovSolar
9	КЭС «Энергия» MPPT Pro 200/60	Россия	МикроАРТ, Andi-Grupp

Для удалённого объекта в Хабаровском крае решающее значение имеют не только технические характеристики контроллера, но и возможность его приобретения у поставщика, способного обеспечить доставку в регион и последующее гарантийное обслуживание. Анализ регионального рынка позволяет выделить поставщиков, имеющих фактическое присутствие в Хабаровске или Владивостоке.

BobrovSolar (БобровСолар) – имеет представительство и склад в Хабаровске по адресу ул. Целинная, 8В, что подтверждено в исходных материалах проекта. Компания предлагает контроллеры Delta SmartWatt MPPT 4860 и SRNE SR-ML4860 с гарантией 2–5 лет.

ООО «Технолайн» – расположен во Владивостоке, что обеспечивает относительно быструю доставку в Хабаровский край. Компания специализируется на оборудовании для автономного электроснабжения.

ООО «Энергия Про» (БобровСолар) – находится непосредственно в Хабаровске, является предпочтительным поставщиком инверторного оборудования и может выступить поставщиком контроллеров под заказ.

Aurinkos – компания, работающая в Хабаровске, предлагает контроллеры Victron Energy SmartSolar MPPT 100/20, однако представленные модели имеют недостаточную мощность для проектируемой системы.

SolarHome (Москва) – федеральный поставщик, осуществляющий доставку по всей России, включая Хабаровский край. Срок поставки оборудования Victron Energy может достигать 60–80 рабочих дней, что является существенным недостатком для оперативной реализации проекта.

Исходя из критерия территориальной близости и наличия сервисной поддержки, из общего списка остаются модели, доступные через хабаровских поставщиков или поставщиков с отлаженной логистикой в регион. В таблице 2.8 приведён сокращённый перечень.

Таблица 2.8 – Контроллеры, доступные для приобретения в Хабаровском крае

№	Модель	Поставщик	Расположение поставщика
1	Delta SmartWatt MPPT 4860	BobrovSolar	Хабаровск
2	SRNE SR-ML4860	BobrovSolar	Хабаровск
3	Victron SmartSolar MPPT RS 450/100	SolarHome (Москва)	доставка по РФ
4	Epever Tracer 10420AN	SolarHome (Москва)	доставка по РФ
5	MUST PC18-10015F	Realsolar (Москва)	доставка по РФ

Для окончательного выбора необходимо сопоставить технические параметры контроллеров с требованиями проектируемой системы. Солнечные панели Solar Vertex S+ TSM-NEG9R.28 (460W) при последовательном соединении 4 штук дают напряжение холостого хода около 215 В. С учётом низких зимних температур в Хабаровском крае (до -40°C) и отрицательного температурного коэффициента напряжения, V_{oc} может возрасти примерно на 10–12%, достигая 237–240 В. Следовательно, контроллер должен иметь максимальное входное напряжение не менее 250 В с запасом. Этому требованию удовлетворяют толь-

ко Victron SmartSolar MPPT RS 450/100 (450 В) и Epever Tracer 10420AN (200 В с небольшим запасом). Контроллеры Delta и SRNE с ограничением 150 В не подходят для последовательного включения четырёх панелей.

Дополнительные критерии выбора включают:

- наличие сервисной поддержки в регионе: ни один из московских поставщиков не имеет собственного сервисного центра в Хабаровском крае, что для удалённого объекта является существенным риском;

- возможность параллельного включения: для масштабирования системы в будущем важна возможность параллельной работы нескольких контроллеров;

- совместимость с LiFePO₄: контроллер должен поддерживать настройку параметров заряда для литий-железо-фосфатных аккумуляторов.

Victron SmartSolar MPPT RS 450/100 демонстрирует наилучшие технические показатели: высокое входное напряжение (до 450 В), ток заряда 100 А, поддержку 48-вольтовых систем, КПД до 98% и встроенную логику заряда LiFePO₄. Однако длительные сроки поставки (до 60–80 рабочих дней) и отсутствие локального сервиса в Хабаровске делают этот вариант менее привлекательным для оперативной реализации проекта.

Epever Tracer 10420AN является более доступным по цене и срокам поставки, но его максимальное входное напряжение 200 В оставляет минимальный запас относительно расчётных 237–240 В при низких температурах, что создаёт риск превышения допустимого напряжения в особо холодные дни.

Учитывая все факторы, наиболее сбалансированным решением для горнолыжной базы «Холдоми» является отказ от установки внешних контроллеров на начальном этапе и использование штатных MPPT-контроллеров инвертора Deye SUN-25K-SG01HP3-EU-AM2. Встроенные контроллеры инвертора обеспечивают два MPPT-трекера с диапазоном 150–850 В, что полностью покрывает потребности проектируемого массива солнечных панелей.

Проведённый анализ показал, что для реальной эксплуатации горнолыжной базы «Холдоми» в условиях Хабаровского края оптимальным решением с точки зрения надёжности и резервирования могло бы стать комбинированное

использование встроенных MPPT-контроллеров инвертора Deye SUN-25K-SG01HP3-EU-AM2 и дополнительных внешних контроллеров (например, Victron SmartSolar MPPT RS 450/100). Однако на текущем этапе проектирования стоит иная задача: разработка робастного цифрового двойника изолированной гибридной энергетической системы и выполнение теоретических расчётов энергобаланса, режимов работы и влияния климатических факторов. Физическая установка оборудования на объекте в настоящий момент не предусмотрена – проект находится на стадии математического моделирования.

В этих условиях использование только штатных MPPT-контроллеров, встроенных в выбранный инвертор, имеет ряд существенных преимуществ:

1. Упрощение математической модели – встроенные контроллеры уже являются неотъемлемой частью инвертора, их поведение описывается паспортными характеристиками (диапазон MPPT 150–850 В, максимальный ток заряда 50 А, два независимых трека). Введение внешних контроллеров потребовало бы дополнительных уравнений связи, учёта их собственных КПД, алгоритмов слежения и возможных конфликтов с управляющей логикой инвертора, что излишне усложняет модель без прироста точности на стадии теоретических изысканий.

2. Сокращение числа переменных и ускорение симуляции – цифровой двойник предполагает многократные прогоны с различными входными данными (инсоляция, температура, график нагрузки). Каждый дополнительный элемент (внешний контроллер) увеличивает размерность системы, время расчёта и риск численной неустойчивости. Для поставленной задачи – получения достоверных оценок выработки, потребления, глубины разряда аккумуляторов и доли замещения резервной сети – встроенных контроллеров инвертора более чем достаточно.

3. Достаточность для валидации основных алгоритмов – цель цифрового двойника на данном этапе – не финальная настройка конкретного оборудования, а проверка принципиальной работоспособности гибридной схемы, подбор оптимальных соотношений мощностей и ёмкостей. Все ключевые процессы

(заряд аккумуляторов от солнечных панелей, ограничение тока, защита от перегрева и перезаряда) встроенные MPPT-контроллеры реализуют на том же уровне качества, что и внешние решения для рассматриваемого масштаба системы.

4. Отсутствие необходимости в резервировании на стадии моделирования – в реальной эксплуатации внешние контроллеры рассматривались бы как резерв на случай выхода инвертора из строя. Однако при создании цифрового двойника мы оперируем детерминированными математическими зависимостями, где отказы оборудования не моделируются (если это не входит в отдельную задачу надёжности). Поэтому резервные каналы управления зарядом на данном этапе избыточны.

Таким образом, для текущего этапа исследования – разработки робастного цифрового двойника и выполнения теоретических расчётов – вполне достаточно встроенных MPPT-контроллеров инвертора Deye SUN-25K-SG01HP3-EU-AM2. Это решение позволяет сохранить модель компактной, прозрачной и легко верифицируемой, избежав неоправданного усложнения. В качестве технической рекомендации на будущее следует отметить, что при переходе от цифрового моделирования к реальному проектированию и монтажу оборудования на горнолыжной базе «Холдоми» вопрос об установке внешних контроллеров должен быть пересмотрен – с учётом окончательной мощности солнечного поля, требований к резервированию и анализа стоимости простоя. Однако на настоящем этапе работы внешние контроллеры не включаются в состав моделируемой системы, а их подбор носит предварительный, рекомендательный характер [12, 16].

Проведённый в разделах 2.1–2.3 подбор основного генерирующего и накопительного оборудования (солнечные панели, аккумуляторные модули, гибридный инвертор) в совокупности с обоснованным выше отказом от внешних MPPT-контроллеров на текущем этапе формирует минимально достаточную конфигурацию для разработки цифрового двойника. Поскольку настоящая работа ограничена рамками теоретического моделирования и не предполагает фи-

зической реализации системы на объекте, в состав виртуальной модели включаются лишь те элементы, которые непосредственно влияют на энергобаланс и динамику зарядно-разрядных процессов.

К числу таких элементов относятся:

- солнечные панели – моделируются по вольт-амперным характеристикам, температурным коэффициентам и данным инсоляции;
- аккумуляторные модули – моделируются через ёмкость, допустимую глубину разряда, внутреннее сопротивление и эффективность заряда-разряда;
- инвертор – моделируется через номинальную мощность, КПД, диапазоны МРРТ и ограничения по току;
- встроенные МРРТ-контроллеры – учитываются как часть инвертора без выделения в отдельные объекты;
- кабельные потери – учитываются параметрически через задание коэффициента полезного действия передачи энергии;
- система мониторинга (упрощённо) – имитируется сбором расчётных величин в каждой итерации модели.

Напротив, следующие категории оборудования исключаются из состава цифрового двойника как не влияющие на вычисление энергетических потоков на принятом уровне абстракции: аппараты защиты (автоматические выключатели, УЗИП, предохранители), заземление и молниезащита, контейнер с системой терморегулирования, крепёжные конструкции, распределительные щиты.

Таким образом, для построения верифицируемой и вычислительно эффективной модели изолированной гибридной энергосистемы горнолыжной базы «Холдоми» достаточно ограничиться перечнем основного оборудования, дополненным параметрическим представлением потерь в пассивных элементах [10, 11, 17, 22]. Такой подход соответствует принципу необходимой достаточности сложности модели и позволяет сосредоточить исследование на ключевых энергетических закономерностях функционирования системы в условиях изменяющейся внешней среды.

3 Интеграция данных

Интеграция данных в разрабатываемом цифровом двойнике изолированной гибридной энергетической системы (ЦД ИГЭС) – один из ключевых этапов, от которого зависят полнота, согласованность и актуальность моделирования [3]. В обновлённой версии информационно-математического обеспечения сбор сведений выполняется автоматически из внешнего метеорологического источника Open-Meteo, который предоставляет архивные и прогнозные значения облачности и температуры воздуха. Затем полученные данные сохраняются в локальную базу данных SQLite и используются для построения и сопровождения цифрового двойника ИГЭС.

Главное при сборке всего воедино – свести в одну кучу погоду, время и координаты. Берём точку на карте, её локальный часовой пояс, и все метеосводки привязываем именно к этому месту и этому времени. Иначе как потом считать солнечную радиацию или анализировать режимы работы установки – просто никак. Без чёткой временной и пространственной привязки модель рассыплется.

С временем вообще отдельная песня. Запросы к Open-Meteo уходят в UTC – так устроен их API. Потом, когда данные приходят, пересчитываются временные метки в местное время объекта. Это не просто формальность. Так отсекается вся путаница с часовыми поясами, переходами на летнее/зимнее – всё, что может испортить картину. В базе оставляем обе метки: локальную и исходную UTC. Потом, когда начнётся анализ, эта пара прозрачно покажет, что и когда было на самом деле [19].

Пространство тоже не забываем. Широта и долгота будущей энергосистемы защиты заранее. Это не какие-то абстрактные числа – именно они определяют, под каким углом падает солнце, сколько рассеянного света пробьётся через облака. В расчётах инсоляции без точной геопривязки не обойтись.

Сам сбор данных идёт по расписанию: и историю тянем, и прогноз. Для прошлых периодов – архивный API Open-Meteo, для текущих и будущих дат –

прогнозный. Так поток информации не рвётся. Можно смотреть и накопленную статистику, и ожидаемые изменения. В каждой записи – время, облачность, температура воздуха. Для моделей солнечных потоков этого хватает с запасом.

Когда данные получены, они проходят первичную чистку и заливаются в SQLite. Там создаётся (или пополняется) таблица с источником, локальным временем, исходным UTC, облачностью и температурой. Чтобы не плодились дубли, перед вставкой проверяем по локальному времени – если запись за этот час уже есть, новая не ляжет. Так при повторных запусках интеграции база не пухнет от копий.

Обработка данных включает и фильтрацию по заданному временному интервалу. Это позволяет включать в базу только те записи, которые действительно относятся к рассматриваемому периоду моделирования. Дополнительно упорядочиваются временные ряды, что необходимо для последующего анализа динамики погодных условий и их влияния на солнечную генерацию [19].

Подводя итог, обновлённый процесс интеграции данных в цифровой двойник ИГЭС представляет собой автоматизированную процедуру получения, преобразования, временной нормализации и хранения метеорологической информации. Использование Open-Meteo как внешнего источника данных вместе с SQLite как средой накопления и хранения создаёт надёжную основу для дальнейшего построения математических моделей и анализа влияния облачности и температуры на эффективность солнечных энергетических установок и систем.

3.1 Сбор данных о географических координатах, времени и метеорологических параметрах

Для информационно-математического обеспечения цифрового двойника солнечных энергетических потоков необходимо собирать данные о географических координатах, временных метках и метеорологических характеристиках, влияющих на формирование солнечной радиации и эффективность её преобразования в электрическую энергию. В обновлённой реализации в качестве ис-

точника метеоданных используется сервис Open-Meteo, предоставляющий архивные и прогнозные значения облачности и температуры воздуха.

Географию фиксируем сразу. Широта и долгота – это не какие-то переменные, которые плавают от запуска к запуску. Они жёстко заданы под конкретный объект, под «Холдоми». Именно эти цифры идут в каждый запрос к метеосервису. Иначе данные придут осреднённые по огромному квадрату, а необходима точечная привязка. Хочется, чтобы облачность и температура были из нашего распада, а не с побережья или с соседней сопки. Так что координаты – первое, что определяет, насколько честно потом посчитается солнечная радиация.

С временной зоной всё немного хитрее. Open-Meteo отдаёт всё в UTC, и это правильно: универсальная шкала не врёт. Но жить и анализировать в UTC неудобно – когда на часах полдень по Гринвичу, у нас уже глубокая ночь. Поэтому берутся все временные метки, полученные от сервиса, и пересчитываются в локальное время Хабаровского края. В базе остаются обе версии: исходная UTC и местная. Так сохраняется прозрачность: всегда можно восстановить, откуда что взялось, и при этом не нужно каждый раз лезть в голову с переводом часов. Никакой путаницы с зимним и летним временем – просто две колонки, и всё [19].

Интервал, за который собираются данные, задается любой. Для исторических срезов – архивный API Open-Meteo. Для сегодняшнего дня и прогнозов – их прогнозный эндпоинт. После того как всё пришло, временные метки приводятся к одному формату и выстраиваются по порядку. Это рутинная работа, но без неё в данных будет каша.

А по сути, из всего многообразия метеопараметров нас интересуют два: облачность и температура воздуха на два метра. Они идут как главные входные переменные для оценки солнечной радиации. Облачность режет прямую и рассеянную, температура через NOCT-модель задирает или опускает КПД панелей. Остальное – либо производные, либо шум. Эти две величины собираются, накапливаются и укладываются в SQLite, формируя непрерывные временные

ряды. И цифровой двойник потом не захлёбывается на дырках в данных – ряды сплошные.

Когда всё собрано, оно не сваливается это в кучу. Данные проходят структурирование: источник, локальное время, UTC, облачность, температура. Именно в таком порядке они ложатся в таблицу. Получается основа – ровные, непротиворечивые записи, готовые для математических моделей солнечных потоков. Модели потом будут крутиться на этих цифрах, описывая поведение системы в конкретном климате, на конкретном склоне. Без этой предварительной возни с координатами, временем и структурой всё дальнейшее моделирование рискует оказаться просто игрой с красивыми графиками.

3.2 Процесс интеграции данных в цифровой двойник

Чтобы цифровой двойник солнечных потоков опирался на живые метеосводки, приходится собирать небольшой конвейер. Данные запрашиваются из Open-Meteo, пересчитываются по времени, фильтруются от дублей и укладываются в базу. Поэтапно.

Первый шаг – Open-Meteo. Архив или прогноз – зависит от дат, которые нам нужны. Так покрывается весь интервал без разрывов, и в дело идут как исторические наблюдения, так и свежие предсказания.

Дальше – время. UTC-метки перегоняются в локальное время площадки. Но исходная метка остаётся. На всякий случай. Это даёт однозначную картину и позволяет при нужде откатить всё обратно.

Третий этап – чистим повторы. В базу попадают только те записи, которые вписываются в интервал моделирования и не дублируют уже сохранённое. Целостность держится, дубли исключены, даже если скрипт загрузки дёрнут повторно.

Наконец, SQLite. Таблица с полями: источник, локальное время, UTC, облачность, температура воздуха. Потом эти залежи скармливаются математиче-

ским моделям – тем, что обсчитывают солнечные потоки и их зависимость от погоды.

По сути, обновлённый механизм интеграции – это автоматика, довольно живучая по отношению к повторным запускам. Метеоданные поступают регулярно, нормализуются и аккуратно складываются в структурированном виде. Надёжная основа, чтобы дальше анализировать, моделировать и искать оптимальные режимы для солнечных установок.

4 Математическая модель ЦД и методы моделирования

Цифровой двойник изолированной гибридной энергосистемы (ЦД ИГЭС) начинают строить не с кода. С расчётов. С математических моделей, описывающих всё подряд – от получения метеосводок до поведения аккумуляторов глухой ночью. Этот раздел – как раз про такой фундамент.

Модели здесь не висят в воздухе сами по себе. Они выстроены цепочкой: что получили на выходе одной, то сразу идёт на вход другой. Разрозненные цифры понемногу превращаются в связное описание энергетического объекта. Для цифрового двойника такое единство – не прихоть, а требование: он обязан повторять логику реальной системы, а значит, ошибка на любом этапе мгновенно обесценит всю цифровую копию. Поэтому каждая модель здесь решает свою конкретную задачу и передаёт эстафету дальше [7].

Открывает раздел модель ETL-процесса (Extract-Transform-Load), отвечающая за поток исходных метеорологических данных. Она описывает, как облачность и температура извлекаются из внешнего API, очищаются от пропусков и дублей, а затем записываются в локальное хранилище. Именно на этом этапе задаётся временная точность всего будущего расчёта: модель сохраняет время и в локальном, и в UTC-формате, чтобы ни сдвиг часового пояса, ни переход на летнее время не исказили картину. Казалось бы, рутинная инженерная задача – но именно с неё стартует вся последующая физика.

Следом вводится детерминированная модель расчёта солнечного потока. Она объединяет астрономический блок, вычисляющий положение Солнца по юлианской дате и эклиптической долготе, с эмпирическими поправками на состояние атмосферы. Стоит отметить, что модель не просто определяет зенитный угол и мощность внеатмосферного излучения; она учитывает сезонное изменение альбедо поверхности, ослабление в воздушной массе и, что особенно важно для дальнейших расчетов, снижение потока под влиянием облачности. Даже под сплошной облачностью расчёт оставляет небольшую долю энергии.

Не то чтобы физика была строгой – скорее инженерная подпорка, чтобы не плодить неоправданные нули в пасмурные периоды.

Когда с физикой солнечного света разобрались, фокус смещается на энергосистему. Модель суточного профиля и расчёта параметров гибридной системы работает уже с мощностью и режимами потребления, а не с потоком. Строится трёхуровневый график нагрузки: минимум, норма, максимум. От него уже пляшут: мощность инвертора, допустимые токи аккумуляторов, потери в кабелях.

Дальше – согласование компонентов. Тут работает дискретная оптимизационная модель. На ограниченной площади крыши ищут плотнейшую раскладку фотоэлектрических модулей, перебирая портрет и альбом. Но геометрический максимум – лишь половина. Полученное поле сразу проверяют на электрическую совместимость с инвертором: напряжение строк солнечных панелей должно лезть в узкое рабочее окно MPPT-контроллера, а суммарные токи – не выходить за допустимые пределы. Параллельно подбирается аккумуляторный банк: по нагрузке, току и требуемой автономии. Модель не гонится за абстрактным оптимумом. Нужна жизнеспособная конфигурация, где геометрия, электрические ограничения и запас энергии не конфликтуют, а поддерживают друг друга.

Замыкает цепочку динамическая модель энергобаланса. Час за часом она сопоставляет генерацию с потреблением. Логика простая: сначала нагрузку пытается закрыть солнечная панель, затем – аккумуляторы, и только когда заряд падает до критического минимума, подключается внешняя сеть. Каждый час система получает своё состояние. Это позволяет увидеть не усреднённую картинку, а реальный суточный цикл – с пиками, спадами и неизбежными дефицитами. В итоге все пять моделей собираются в сквозной расчётный контур. Онто и есть математическое ядро двойника, без которого программная оболочка осталась бы лишь интерфейсом без содержания [17, 29].

4.1 Математическая модель процесса получения, преобразования и хранения данных облачности и температуры (ETL-модель)

Тип модели: модель ETL-процесса (Extract-Transform-Load) для метеорологических временных рядов [19]. Эта модель описывает полный цикл работы с метеорологическими данными: от их извлечения из внешнего источника до приведения к единому виду и сохранения в базе. Для цифрового двойника изолированной гибридной энергетической системы, которая опирается на солнечную энергию, такой контур особенно важен, потому что именно он задаёт качество исходной информации. Если на входе появляются пропуски, временные сдвиги или повторы, это почти сразу отражается на точности последующих расчётов. Поэтому здесь акцент сделан не только на формальных преобразованиях, но и на контроле корректности каждого шага.

Модель использует следующие входные переменные:

- φ – географическая широта (в градусах, $\varphi \in [-90,90]$);
- λ – географическая долгота (в градусах, $\lambda \in [-180,180]$);
- $\tau_{start}^{loc}, \tau_{end}^{loc}$ – границы интересующего интервала времени в местном часовом поясе, заданные в формате ГГГГ-ММ-ДДТч:мм;
- Z_{loc} – идентификатор местного часового пояса (строка, например «Asia/Vladivostok»);
- D_{now} – текущая календарная дата в UTC (результат округления текущего момента до начала суток);
- файл базы данных *screenshots* с таблицей, схема которой содержит поля: *datetime* (строка локального времени (уникальный ключ)), *datetime_utc* (строка времени в UTC), *clouds* (облачность (вещественное число)), *temperature* (температура на высоте 2 м (вещественное число)).

Здесь, по сути, задаются все опорные параметры, с которыми дальше работает ETL-контур. Широта и долгота определяют точку наблюдения, временной интервал задаёт границы отбора, а часовой пояс нужен для того, чтобы временные метки были интерпретированы одинаково и не “плавали” при по-

следующей обработке. Таблица *screenshots* служит хранилищем уже подготовленных записей; в ней важна не только сама погода, но и то, как именно она привязана ко времени.

Первым этапом идет разбор строкового представления времени и приведение к UTC.

Входные метки времени от API имеют вид YYYY-MM-DDTНН:ММ (без секунд). Сначала их алгоритм приводит к полному виду YYYY-MM-DDTНН:ММ:SS, добавляя :00. После этого строка преобразуется в осведомлённый (aware) объект даты-времени UTC:

$$t_{iso} \xrightarrow{\text{normalize}} t_{full} \in S_{ISO}, \text{parse}(t_{full}) = dt_{utc} \in \tau_{UTC}, \quad (1)$$

где τ_{UTC} – множество дат-времени с приписанным часовым поясом UTC. Функция *parse* выполняет разбор по ISO 8601 (международный стандарт записи даты и времени в одном и том же, однозначном формате); если строку разобрать не удастся, запись отбрасывается.

На практике этот шаг нужен для того, чтобы все дальнейшие операции опирались на единый формат времени. Иначе одна и та же метка могла бы трактоваться по-разному в зависимости от источника. Для временных рядов это особенно чувствительно: одна ошибка в интерпретации часа уже меняет порядок записей и нарушает корректность фильтрации.

Следом идет конвертация часового пояса. Преобразование момента времени из UTC в произвольный часовой пояс Z задаётся оператором:

$$\text{Convert}(dt_{utc}, Z) = dt_{utc} \cdot \text{astimezone}(Z), \quad (2)$$

где $\text{astimezone}(Z)$ – стандартная функция смещения на величину, соответствующую часовому поясу Z в конкретный момент времени, с учётом летнего времени, если оно применяется. Результат dt_{loc} принадлежит множеству τ_Z .

$$S_{loc}(dt_{loc}) = \text{strftime}(dt_{loc}, '%Y - %m - %d %H: %M: %S'), \quad (3)$$

$$S_{loc}(dt_{uts}) = \text{strftime}(dt_{uts}, '%Y - %m - %d %H: %M: %S').$$

Здесь важен именно двойной способ хранения: локальная метка нужна для удобной работы с данными на стороне целевой системы, а UTC-метка сохраняет исходное значение без потерь. Это удобное решение, потому что в слу-

чае проверки или повторной обработки всегда можно восстановить первичный временной контекст. Иначе говоря, система не «теряет память» о том, откуда пришла запись.

Вторым этапом следует локализация и преобразование граничных значений. В алгоритм передаются границы $\tau_{start}^{loc}, \tau_{end}^{loc}$ как наивные (naive) даты-времени, подразумевая местный пояс Z_{loc} . Сама по себе такая запись не говорит, это 13:00 по UTC, по Владивостоку или по Москве. То есть время как бы «без привязки к месту». Именно поэтому её и называют naive – «наивной», в смысле неосведомлённой о часовом поясе. Следовательно, наивные даты-времени – это временные метки, не содержащие информации о часовом поясе и потому требующие явной локализации перед дальнейшей обработкой.

Локализация, то есть привязка к поясу, выполняется по правилу:

$$\tau_{start}^Z = \begin{cases} localize_{Z_{loc}}(\tau_{start}^{loc}), & \text{если } \tau_{start}^{loc} \text{ – наивное,} \\ \tau_{start}^{loc} \cdot astimezone(Z_{loc}), & \text{иначе.} \end{cases} \quad (4)$$

Аналогично задаётся τ_{end}^{loc} . Затем оба момента переводятся в UTC:

$$\tau_{start}^{UTS} = Convert(\tau_{start}^Z, UTS), \tau_{end}^{UTS} = Convert(\tau_{end}^Z, UTS), \quad (5)$$

при условии корректности: $\tau_{end}^{UTS} \geq \tau_{start}^{UTS}$.

На первый взгляд этот блок выглядит чисто техническим, но именно он задаёт точность всей дальнейшей работы. Если временные границы локализованы неверно, тогда даже корректные погодные данные окажутся смещёнными относительно нужного интервала. А это уже напрямую влияет на результаты моделирования.

На третьем этапе происходит взаимодействие с API (интерфейс программирования приложений; набор правил, протоколов и инструментов, позволяющий одной программе взаимодействовать с другой) Open-Meteo. Модель использует два экземпляра API: архивный (ARCHIVE) и прогнозный (FORECAST). Запрос к API описывается функцией:

$$F(API, \varphi, \lambda, d_{start}, d_{end}): API \in \{ARCHIVE, FORECAST\}, \quad (6)$$

где d_{start}, d_{end} – даты в формате ГГГГ-ММ-ДД (целые календарные сутки в UTC).

Отклик API – это JSON-объект, из которого извлекаются три массива: временные метки $T = [t_1, t_2, \dots, t_n]$, облачность $C = [c_1, c_2, \dots, c_n]$ и температура $Temp = [T_1, T_2, \dots, T_n]$. Функция F (6) возвращает упорядоченный по времени список кортежей:

$$F(\dots) = (t_{i_k}, c_{i_k}, T_{i_k})_{k=1}^n, \quad t_{i_1} \leq t_{i_2} \leq \dots \leq t_{i_n}. \quad (7)$$

Здесь стоит обратить внимание на самую логику внешнего источника. Open-Meteo в этой модели не просто поставяет набор чисел, а отдаёт именно временной ряд, уже пригодный для последующей обработки. Это очень удобно: не нужно вручную собирать погодные данные из разных мест и сводить их к единому виду. Но, разумеется, сырые результаты всё равно требуют проверки и нормализации, чем и занимается следующий этап алгоритма модели.

Четвертый этап – процесс фильтрации, преобразования и нагрузки. Для каждого элемента $r = (t_{iso}, c, T)$ из объединённого набора архивных и прогнозных записей выполняются следующие нижеописанные шаги.

Проверка заполненности. Запись считается действительной, если:

$$c \neq null \wedge T \neq null. \quad (8)$$

Иначе запись пропускается. Это простая, но очень важная проверка. Без неё в базу могли бы попасть неполные наблюдения, а такие записи потом мешают анализу: где-то нарушается непрерывность ряда, где-то появляются ошибки при расчётах. Поэтому сначала отсеиваются все пустые значения, и только потом данные передаются дальше.

Временной фильтр. Из t_{iso} в ходе работы алгоритма получается dt_{utc} (1). Запись допускается к дальнейшей обработке, если:

$$\tau_{start}^{UTS} \leq dt_{utc} \leq \tau_{end}^{UTS}. \quad (9)$$

Проверка по времени ограничивает обработку только тем интервалом, который действительно интересует на данном этапе алгоритм. Это удобно не только с точки зрения логики, но и с точки зрения производительности: лишние записи не тратят ресурсы и не попадают в итоговую таблицу.

Конвертация в локальное время. На данном шаге формируются сразу две строковые формы одного и того же момента времени:

$$dt_{loc} = Convert(dt_{utc}, Z_{loc}): s_{loc} = S_{loc}(dt_{loc}), s_{utc} = S_{utc}(dt_{utc}). \quad (10)$$

Локальная строка используется как основной ключ при хранении, а UTC-строка остаётся в записи как опорное значение.

Проверка уникальности и вставка. Пусть $R_{ex}(s)$ – отношение, проверяющее наличие записи в таблице *screenshots* с полем $datetime = s$:

$$R_{ex}(s_{loc}) = \begin{cases} 1, & \text{если } \exists \text{ запись с } datetime = s_{loc}. \\ 0, & \text{иначе.} \end{cases} \quad (11)$$

Если $R_{ex}(s_{loc}) = 0$, то выполняется вставка кортежа $(site, s_{loc}, s_{utc}, c, T)$ в таблицу *screenshots*. Если запись уже присутствует, то она игнорируется. Это решение защищает систему от повторной загрузки одних и тех же данных. Проверка, кстати, идёт именно по локальному времени. Так удобнее. Так логичнее для всей дальнейшей возни с таблицей. Для прикладной системы это, по сути, главный барьер, который не даёт плодиться дублям.

На выходе получают две переменные: N_{req} – общее число записей, пришедших из API до фильтрации, и N_{add} – сколько из них реально добавлено после всех проверок. По ним сразу видно, насколько результативно прошла загрузка. Если разница между ними аховая – значит, в исходных данных либо полно пропусков, либо добрая часть записей просто вываливается за границы интервала.

Разбиение временного диапазона на архивную и прогнозную части.

Текущая дата (если она участвует в моделировании) $D_{now}(UTC)$ делит временную ось на периоды, обслуживаемые разными API.

Архивный диапазон (даты строго раньше D_{now}):

$$D_{arch} = [\tau_{start}^{UTS}; \min(\tau_{end}^{UTS}, D_{now} - 1 \text{ день})]. \quad (12)$$

Если $\tau_{start}^{UTS} < D_{now}$, то выполняется запрос (6):

$$A_1 = F(ARCHIVE, \varphi, \lambda, \min(D_{arch}), \max(D_{arch})). \quad (13)$$

Прогнозный диапазон:

$$D_{fc} = [\max(\tau_{start}^{UTS}, D_{now}); \tau_{end}^{UTS}]. \quad (14)$$

Если $\tau_{start}^{UTS} \geq D_{now}$, то выполняется запрос:

$$A_2 = F(FORECAST, \varphi, \lambda, \min(D_{fc}), \max(D_{fc})). \quad (15)$$

Общий набор исходных записей:

$$R = A_1 \cup A_2. \quad (16)$$

Для каждой части вызывается функция вставки (11) с одними и теми же границами фильтрации $[\tau_{start}^{UTS}, \tau_{end}^{UTS}]$. Это гарантирует, что записи на стыке архивного и прогнозного интервалов будут корректно отобраны и не продублируются.

Агрегация результатов. Итоговые показатели работы модели для одного запуска:

$$N_{t_req} = |A_1| + |A_2|, N_{t_add} = N_{add}(A_1) + N_{add}(A_2), \quad (17)$$

где N_{add} – количество успешных вставок.

По этим двум цифрам сразу видно, сколько записей пришло от источника, а сколько действительно осело в базе после всех фильтров.

Теперь про ограничения. Уникальность держится на локальном времени *datetime* – так исключаются дубликаты на целевом сайте. С часовыми поясами тоже порядок: все сравнения выполняются в UTC, поэтому переходы на летнее и зимнее время обрабатываются корректно. Целостность: записи, где нет данных об облачности или температуре, просто игнорируются. И последнее – разделение источников. Благодаря D_{now} архивные и прогнозные данные не пересекаются по датам.

4.2 Математическая модель расчёта солнечного потока

Тип модели: детерминированная физико-математическая модель расчёта солнечной радиации с последующей эмпирической коррекцией по облачности и температуре воздуха.

Эта модель уже немного другого рода, чем ETL-описание из предыдущего раздела. Здесь речь идёт не просто о переносе и хранении сведений, а о вычислении физически осмысленной величины – солнечного потока – на основе времени наблюдения, географического положения и метеорологических характеристик. Примечательно, что в модели сочетаются сразу два подхода: астро-

номический расчёт положения Солнца и прикладная коррекция, связанная с состоянием атмосферы и температурой окружающего воздуха.

Модель использует следующие входные переменные:

- φ – географическая широта (в градусах, $\varphi \in [-90,90]$);
- λ – географическая долгота (в градусах, $\lambda \in [-180,180]$);
- dt – момент времени в UTC, по которому выполняется расчёт;
- c – облачность, %;
- T_{amb} – температура окружающего воздуха, °C;
- d_{oy} – номер дня года;
- $NOCT$ – номинальная рабочая температура фотоэлектрического модуля, °C;
- η_{mod} – базовый КПД модуля;
- k_{temp} – температурный коэффициент КПД;
- τ – оптическая толщина атмосферы;
- $G_{sc} = 1361,0 \text{ Вт/м}^2$ – солнечная постоянная.

Отдельно стоит отметить, что модель работает не с «сырыми» данными, а с уже подготовленным временным рядом. В нём время упорядочено по возрастанию, облачность ограничена диапазоном от 0 до 100, а температура приведена к числовому виду. Если каких-то значений нет – подставляем умолчания: температуру 25 °C, облачность ноль. Без этого расчёт начнёт сыпаться при первых же пропусках. А пропуски случаются. Данные, они почти всегда чуть-чуть да неидеальные.

Разберём алгоритм по шагам, так нагляднее. Первый шаг – временная метка приводится к единой шкале.

Момент времени, для которого считается солнечный поток, рассматривается как UTC. Если метка уже содержит часовой пояс – переводим в универсальное время. Если метка «наивная», без указания пояса – считаем её UTC и не трогаем смещения. Без такой нормализации астрономические вычисления просто плывут, им нужна одна шкала, иначе всё вразнобой.

Далее из даты извлекается номер дня года:

$$d_{oy} = \text{day_of_year}(dt). \quad (18)$$

Эта величина используется как сезонный аргумент. По сути, она связывает расчёт с положением Земли на орбите в конкретный день года. Без этого модель была бы слишком грубой: зимний и летний режимы одинаковыми не бывают.

Следующим шагом для астрономического блока следует вычисление юлианской даты (астрономическая система непрерывного отсчета дней, показывающая количество суток, прошедших с полудня (12:00 по Гринвичу) 1 января 4713 года до н. э.):

$$JD = \text{JulianDay}(dt), \quad (19)$$

а затем вводится смещение относительно эпохи J2000:

$$d = JD - 2451545.0. \quad (20)$$

Юлианская дата здесь играет роль удобного непрерывного счётчика времени. Через неё проще выводить параметры солнечного положения, чем через обычный календарный формат.

Далее третьим шагом определяется средняя астрономическая и средняя долгота Солнца:

$$g = (357,529 + 0,98560028 \cdot d) \text{ mod } 360, \quad (21)$$

$$q = (280,459 + 0,98564736 \cdot d) \text{ mod } 360.$$

После этого вычисляется приближенная эклиптическая долгота:

$$L = (q + 1,915 \cdot \sin(g) + 0,020 \cdot \sin(2g)) \text{ mod } 360. \quad (22)$$

Здесь тригонометрические поправки нужны не ради красоты: они учитывают эллиптичность орбиты и чуть точнее описывают реальное положение Солнца. Если смотреть формально, это уже не простая календарная зависимость, а компактная астрономическая модель. Наклон же эклиптики задается выражением:

$$\varepsilon = 23,439 - 3,56 \cdot 10^{-7} \cdot d. \quad (23)$$

Далее через наклон эклиптики (23) и эклиптическую долготу (22) вычисляется солнечная деклинация:

$$\delta = \arcsin(\sin(\varepsilon) \cdot \sin(L)). \quad (24)$$

Деклинация показывает, как Солнце «поднимается» или «опускается» относительно небесного экватора. Для расчёта освещённости земной поверхности это один из ключевых углов.

Четвертым шагом следует вычисление часового угла и зенита. Для перехода от небесной геометрии к положению Солнца над конкретной точкой используется приближённое местное звёздное время. Сначала определяется гринвичское среднее звездное время, затем оно сдвигается на долготу наблюдателя. В упрощённой записи это даёт часовой угол:

$$H = LMST - \alpha, \quad (25)$$

где α – приближение прямого восхождения;

$LMST$ – локальное звёздное время.

После этого вычисляется зенитный угол:

$$\cos(z) = \sin(\varphi) \cdot \sin(\delta) + \cos(\varphi) \cdot \cos(\delta) \cdot \cos(H), \quad (26)$$

$$z = \arccos(\cos(z)).$$

Если $z \geq 90^\circ$, Солнце находится ниже горизонта, и прямая составляющая потока на поверхности принимается равной нулю. Это короткое правило, но очень важное: ночью модель не должна «придумывать» солнечную радиацию.

Пятым шагом идет непосредственно расчет солнечного потока. Первый множитель, влияющий на интенсивность потока, учитывает сезонное изменение расстояния между Землёй и Солнцем:

$$F_{dist} = 1 + 0,003 \cdot \cos\left(2\pi \cdot \frac{d_{oy}}{365}\right). \quad (27)$$

Эта поправка небольшая, но она добавляет физической достоверности. На первый взгляд разница кажется незначительной, однако для длинных временных рядов она уже становится заметной.

Дополнительно для достоверности вычислений вводится сезонный коэффициент альбедо. Коэффициент Альбедо измеряется в безразмерных единицах от 0 до 1, где 0 означает полное поглощение излучения, а 1 – полное отражение:

$$Al = 0,5 \cdot \left(1 + 0,6 \cdot \sin\left(2\pi \cdot \frac{d_{oy} - 172 - 91}{365}\right)\right). \quad (28)$$

После этого формируется отражающий множитель:

$$F_{refl} = 1 + Al. \quad (29)$$

Коэффициент Альбедо в формуле (29) записан как $(1 + Al)$, чтобы учесть влияние отражательной способности поверхности на поток солнечной энергии.

Замысел здесь довольно понятный: поверхность под Солнцем не только получает излучение, но и частично отражает его, а характер отражения меняется по сезонам. Поэтому в модели присутствует именно сезонная зависимость, а не одна постоянная величина.

Так же есть потребность учитывать и внеатмосферные, и поверхностные составляющие. Внеатмосферная составляющая потока на горизонтальную поверхность вычисляется по вот такой формуле:

$$I_{ex} = G_{sc} \cdot F_{dist} \cdot \cos(z) \cdot F_{refl}. \quad (30)$$

Если значение становится отрицательным, оно приводится к нулю. Это естественное ограничение, ведь поток не может быть меньше нуля.

Далее используется приближение относительной массы воздуха по формуле Кастена-Янга (Kasten–Young) [24]:

$$m = \frac{1}{\cos(z) + 0,50572 \cdot (96,07995 - z)^{-1,6364}}, z < 90^\circ. \quad (31)$$

Если зенитный угол больше или равен 90° , масса воздуха считается очень большой, а итоговая передача излучения – практически нулевой.

Атмосферное ослабление задается выражением:

$$\tau_{atm} = e^{-\tau \cdot m}. \quad (32)$$

Тогда поток у поверхности принимает вид:

$$I_{sur} = I_{ex} \cdot \tau_{atm}. \quad (33)$$

Вот здесь модель становится особенно прикладной и уже не просто определяет, где находится Солнце, а переводит это положение в оценку реальной доступной радиации у поверхности Земли.

Следующий шаг связан с оценкой температуры фотоэлектрической ячейки, который представлен в виде:

$$T_{cell} = T_{amb} + \frac{NOCT-20}{800} \cdot I_{sur}. \quad (34)$$

После этого вводится температурный коэффициент эффективности:

$$f_T = 1 + k_{temp} \cdot (T_{cell} - 25). \quad (35)$$

Итоговый солнечный поток, скорректированный по температуре, задается формулой:

$$I_{sol} = I_{sur} \cdot f_T. \quad (36)$$

Если результат оказывается нечисловым или отрицательным, он заменяется нулём. Вот так просто. Это уже не физическая теория в чистом виде, а инженерная модель, которая должна работать устойчиво.

Седьмым шагом следует оценка облачности. На этом и последующих этапах солнечный поток используется не только как итоговая характеристика, но и как основа для оценки облачности в согласованной временной шкале. Здесь вводится зависимость:

$$C_{est} = I_{sol} \cdot \left(1 - \frac{c}{105}\right), \quad (37)$$

где c – исходная облачность, полученная из внешнего источника, %.

Формула (37) делит процент облачности на 105 – такой вот масштабный коэффициент, чтобы даже под сплошными тучами оставалась хоть какая-то солнечная энергия. Совсем в ноль не уходит. При максимальной облачности всё равно просачивается немного, с учётом потерь на прохождение через облака и атмосферу [6, 9]. Связь сугубо прикладная. На строгую атмосферную теорию не тянет, зато увязывает интенсивность потока с тем, насколько небо затянуто. По сути, чем облачность выше, тем сильнее ослабляется итоговая величина. При нуле облачности поток почти целый, с ростом – падает.

Тут главное не перепутать: получаем не облачность как атмосферный параметр, а производную оценку солнечного потока с учётом облачного влияния. Именно это и нужно для дальнейшего анализа.

Последний шаг – агрегирование. Модель считает последовательность значений, и их можно накопить за выбранный период. Если считать, что значения заданы с часовым шагом, то суммарная энергия на единицу площади оценивается так:

$$E = \frac{1}{1000} \sum_{i=1}^n I_{sol,i}. \quad (38)$$

Результат выражается в кВт·ч/м².

Это уже следующий уровень модели. Здесь поток превращается в интегральную характеристику, которая удобна для оценки солнечного ресурса за выбранный период. Иными словами, модель умеет не только считать мгновенное значение, но и накапливать его во времени.

Свойства и ограничения модели сводятся к следующему: она детерминирована, то есть при одинаковых входных данных всегда даёт один и тот же результат, без случайных отклонений; основные формулы обладают физической интерпретируемостью, поскольку опираются на астрономические и теплофизические соотношения, а не на произвольные преобразования; вместе с тем отдельные параметры остаются эмпирическими – значения τ , $NOCT$, k_{temp} и коэффициент облачности задаются как прикладные величины и могут подстраиваться под условия региона. Однако область применения модели ограничена, потому что при зенитном угле выше 90° поток обнуляется, и ночные значения не рассматриваются как полезная солнечная генерация; наконец, модель чувствительна к качеству исходных данных, и ошибки во времени, координатах или температуре сразу отражаются на результате [25, 29, 39].

4.3 Математическая модель суточного энергетического профиля и расчёта параметров гибридной изолированной энергосистемы

Тип модели: детерминированная параметрическая математическая модель суточной нагрузки, накопления энергии и согласования основных элементов гибридной энергетической системы [17].

Эта модель описывает уже не обработку данных в чистом виде, а инженерную сторону системы: как распределяется нагрузка в течение суток, какой запас требуется от инвертора, как соотносятся между собой солнечная панель, аккумуляторная батарея и преобразователь, и какие потери стоит учитывать при передаче энергии. На первый взгляд она выглядит как набор отдельных расчётов, но на деле все блоки связаны между собой довольно жёстко. Одно значение здесь сразу тянет за собой другое.

В основе модели лежит набор фиксированных параметров, задающих режим работы системы:

- φ – географическая широта (в градусах, $\varphi \in [-90,90]$);
- λ – географическая долгота (в градусах, $\lambda \in [-180,180]$);
- P_{max}^{load} – суммарная энергия или нагрузка максимального режима за рабочий день, кВт;
- t_{max} – продолжительность максимального режима в течение суток, ч;
- k_{norm} – обычная нагрузка, доля от часовой максимальной нагрузки;
- k_{min} – минимальная нагрузка, доля от часовой максимальной нагрузки;
- параметры солнечного модуля;
- параметры аккумуляторного банка (АКБ);
- параметры инвертора;
- k_{pv} – потери на DC-кабелях от солнечных панелей;
- k_{batt} – потери на кабелях АКБ;
- k_{ac} – потери на AC-стороне.

Примечательно, что здесь модель строится не только вокруг энергии как таковой, но и вокруг режимов её распределения. Это важный момент: система должна не просто «иметь мощность», а работать в нескольких состояниях – минимальном, обычном и максимальном. Для изолированного энергокомплекса это принципиально.

Модель суточной нагрузки. Сначала суммарная нагрузка максимального режима переводится в часовую среднюю мощность:

$$P_{max}^{avg} = \frac{P_{max}^{load}}{t_{max}}. \quad (39)$$

Поскольку исходная величина задаётся за рабочий день, а расчёт требуется в ваттах и киловаттах, вводится преобразование:

$$P_{max}^w = P_{max}^{avg} \cdot 1000. \quad (40)$$

Отсюда формируется базовая точка отсчёта. Дальше уже от неё считаются обычная и минимальная нагрузки. Это удобно, потому что все режимы оказываются связанными через одно опорное значение.

В созданной алгоритме обычная нагрузка задаётся фиксировано как k_{norm} от часовой максимальной мощности:

$$P_{norm} = k_{norm} \cdot P_{max}^w, \quad (41)$$

а минимальная – как k_{min} :

$$P_{min} = k_{min} \cdot P_{max}^w. \quad (42)$$

В итоге формируется вектор суточных режимов нагрузки:

$$P_{load} = [P_{norm}, P_{max}^w, P_{min}]. \quad (43)$$

Здесь, по сути, задаётся трёхуровневый профиль потребления. Он не усложняет модель без нужды, но и не сводит её к одной средней цифре, что было бы слишком грубо.

Как уже было упомянуто выше, суточный цикл разбивается на три интервала: минимальная нагрузка, максимальная нагрузка, обычная нагрузка.

Длительность каждого интервала определяется функцией:

$$h_{int} = \begin{cases} t_{end} - t_{start}, & t_{end} \geq t_{start}, \\ (24 - t_{start}) + t_{end}, & t_{end} < t_{start}. \end{cases} \quad (44)$$

Для суток формируется профиль:

$$P_{24h} = [P_{min}, \dots, P_{min}, P_{max}^w, \dots, P_{max}^w, P_{norm}, \dots, P_{norm}], \quad (45)$$

где $P_{min}, \dots, P_{min} = h_{min}$;

$P_{max}^w, \dots, P_{max}^w = h_{max}$;

$P_{norm}, \dots, P_{norm} = h_{norm}$.

Суточная потребность в электроэнергии определяется как сумма энергий по каждому режиму:

$$E_{min} = P_{min} \cdot h_{min}, \quad E_{norm} = P_{norm} \cdot h_{norm}, \quad E_{max} = P_{max}^w \cdot h_{max}. \quad (46)$$

Полная суточная энергия равна:

$$E_{day} = E_{min} + E_{norm} + E_{max}. \quad (47)$$

Данная величина показывает, сколько энергии система должна покрыть за сутки с учётом режима работы нагрузки, а не в усреднённом, почти абстрактном виде.

Солнечный модуль. Для расчётной части фотополя (полезной площади) используются параметры одного фотоэлектрического модуля выбранной солнечной панели:

- a – длина ячейки, м;
- b – ширина ячейки, м;
- N_{cell} – число ячеек, шт;
- N_{pan} – число солнечных панелей, шт;
- η_{ph} – коэффициент полезного действия (КПД), %;
- k_{ph} – температурный коэффициент, °С;
- $NOCT$ – номинальная рабочая температура фотоэлектрической ячейки, °С;
- V_{mp} – рабочее напряжение, В;
- V_{oc} – напряжение холостого хода, В;
- I_{mp} – ток в точке максимальной мощности, А;
- I_{sc} – ток короткого замыкания, А.

Эти величины задают фундамент электрической части модели. Они нужны не для того, чтобы согласовывать солнечную генерацию с входными пределами инвертора и понимать, как поведёт себя система при разных уровнях освещённости и температуры.

Аккумуляторный банк. Аккумуляторная часть описывается как банк на базе литий-железо-фосфатного аккумулятора модульного типа с распространённым для такого вида аккумуляторов рабочим напряжением 51,2 В. Аккумуляторы этой разновидности предназначены для применения в телекоммуникационной, компьютерной, радиотехнической и климатической сферах. Устройства выполнены в стандартном формате для установки в телекоммуникационные шкафы и стойки 19". Для обеспечения стабильной и длительной работы батареи данного вида оснащены интеллектуальной системой управления BMS (Battery Management System). Она контролирует ключевые параметры, включая ток, напряжение и температуру, а также выполняет защитные функции и авто-

матическую балансировку элементов, что способствует увеличению срока службы системы.

Количество аккумуляторов по параллели обозначается как n_{bat} . Тогда номинальная ёмкость АКБ равна:

$$C_{nom} = c_{bat} \cdot n_{bat}. \quad (48)$$

где c_{bat} – ёмкость одного аккумулятора, А·ч.

Ниже так же представлены параметры, которые далее используются в алгоритме:

- V_{bat} – рабочее напряжение, В;
- k_{min}^{SOC} – минимальный допустимый уровень заряда;
- k_{mod} – коэффициент использования батареи;
- k_{ch} – допустимая доля тока заряда.

Из них получаются три характерные ёмкости (C_{us} – доступная ёмкость одного АКБ, А·ч; C_{lim} – фактическая ёмкость одного АКБ, А·ч; C_{DoD} – минимально допустимая ёмкость одного АКБ, А·ч):

$$\begin{aligned} C_{us} &= c_{nom} \cdot k_{mod}, \\ C_{lim} &= C_{nom} \cdot (1 - k_{min}^{SOC}) \cdot k_{mod}, \\ C_{DoD} &= C_{nom} \cdot k_{min}^{SOC} \cdot k_{mod}. \end{aligned} \quad (49)$$

Максимальный допустимый ток заряда оценивается как:

$$I_{ch,max} = C_{nom} \cdot k_{ch}. \quad (50)$$

Здесь модель уже не просто хранит паспортные данные аккумулятора, а переводит их в рабочие ограничения. Это очень полезно для последующих расчётов: алгоритм сразу понимает, сколько ёмкости реально доступно и где проходит граница безопасного режима.

Гибридный инвертор. В расчёте используется гибридный инвертор трехфазного типа, с возможностью удаленного мониторинга по Wi-Fi, с увеличенным напряжением от солнечных батарей, которое объединяет в себе функции инвертора, солнечного зарядного устройства и зарядного устройства для аккумуляторов, обеспечивая надежное бесперебойное питание. Та же инверто-

ры данного типа имеют встроенные MPPT (Maximum Power Point Tracking, высокоэффективное устройство для солнечных энергосистем, которое отслеживает точку максимальной мощности солнечной панели и преобразует напряжение для заряда аккумуляторов) контролеры.

Ключевые параметры гибридного инвертора задаются следующим образом:

- P_{inv} – номинальная мощность, кВт;
- η_{inv} – КПД, %;
- V_{start} – порог запуска инвертора, В;
- $V_{bt,max}$ – максимальное рабочее напряжение для АКБ, В;
- $V_{bt,min}$ – минимальное рабочее напряжение для АКБ, В;
- $I_{ch,max}$ – максимальный ток заряда АКБ, А;
- $I_{dis,max}$ – максимальный ток разряда АКБ, А;
- $V_{MPPT,max}$ – максимальное напряжение для MPPT контролера, В;
- $V_{MPPT,min}$ – минимальное напряжение для MPPT контролера, В;
- $V_{PV,max}$ – максимальное напряжение для солнечной панели, В;
- $V_{PV,min}$ – минимальное напряжение для солнечной панели, В;
- N_{MPPT} – количество MPPT модулей, шт;
- P_{AC} – номинальная активная мощность на переменном токе, Вт;
- S_{AC} – максимальная полная мощность, ВА;
- $I_{MPPT,max}$ – предельно допустимый входной ток MPPT в рабочем режиме, А;
- $I_{sc,max}$ – предельно допустимый ток короткого замыкания на входе MPPT;
- $I_{AC,nom}$ – номинальный выходной ток, А;
- $I_{AC,max}$ – максимально допустимый выходной ток, А.

Модель не ограничивается одним паспортным значением мощности. Это было бы слишком грубо. Она вводит целый набор электрических границ, и

именно они позволяют проверять, насколько нагрузка и источник питания вообще совместимы между собой.

Расчёт потребности в мощности инвертора и тока батареи. Чтобы инвертор работал не на пределе, вводится запас k_{res} . Это коэффициент запаса по мощности инвертора. Его вводят затем, чтобы расчётная нагрузка не совпадала с предельной мощностью устройства один в один. Смысл простой: если инвертор подбирать ровно под рассчитанную нагрузку, то в реальной работе он будет почти всё время трудиться на грани возможностей. А это не лучший режим. У инвертора бывают кратковременные пики потребления, небольшие потери, отклонения параметров оборудования, да и сама нагрузка редко бывает идеально ровной.

Тогда требуемая мощность инвертора оценивается как:

$$P_{inv,req} = P_{load} \cdot k_{res}, \quad (51)$$

где P_{load} – расчётная нагрузка, кВт.

Ток, необходимый от аккумуляторной батареи при пиковом режиме, определяется формулой:

$$I_{bat} = \frac{P_{inv,req} \cdot 1000}{V_{bat} \cdot \eta_{inv}}. \quad (52)$$

Число 1000 в формуле (52) нужно для перевода киловатт в ватты, так как требуемая мощность инвертора $P_{inv,req}$ записана в киловаттах, а напряжение батареи V_{bat} записано в вольтах.

Фактическое же ограничение по аккумуляторному батарейному току берётся как минимум из двух величин:

$$I_{bat,al} = \min(I_{ch,max}, I_{dis,max}). \quad (53)$$

Здесь стоит отметить важный инженерный момент: система не просто считает, хватит ли энергии. Она ещё и проверяет, выдержат ли батарея и инвертор требуемый ток.

Отдельно стоит упомянуть потери в кабельных линиях, что соединяют между собой солнечные панели, аккумуляторы и инверторы. Энергия в реаль-

ной схеме неизбежно теряется по пути – на DC-стороне, на батарейных цепях и на AC-участке. Игнорировать это было бы слишком наивно.

Потери энергии задаются тремя коэффициентами:

- $k_{cab_{pv}}$ – потери на DC-кабелях от панелей;

- $k_{cab_{batt}}$ – потери на кабелях АКБ;

- $k_{cab_{ac}}$ – потери на AC-стороне.

Суммарный коэффициент потерь равен:

$$k_{cable} = k_{cab_{pv}} \cdot k_{cab_{batt}} \cdot k_{cab_{ac}}. \quad (54)$$

Этот множитель нужен для того, чтобы итоговая оценка системы не выглядела слишком оптимистичной.

Ранее уже упоминались DC и AC стороны и далее эти определения так же будут встречаться, поэтому следует пояснить, что это такое. DC-сторона – это часть системы, где энергия передаётся в виде постоянного тока. Сюда относятся солнечные панели, аккумуляторная батарея и та цепь, по которой энергия от них поступает к инвертору. Именно на этой стороне напряжение и ток сохраняют постоянную полярность, что удобно для накопления энергии и прямой работы фотоэлектрических модулей. AC-сторона, напротив, представляет собой участок системы с переменным током. На ней инвертор преобразует энергию в форму, пригодную для питания потребителей здания: освещения, оборудования, бытовых и офисных нагрузок. Иными словами, DC-сторона отвечает за сбор и хранение энергии, а AC-сторона – за её использование в сети переменного тока.

На выходе модель позволяет оценить суточное энергопотребление в ватт-часах E_{day} (47) и киловатт-часах:

$$E_{day,kw} = \frac{E_{day}}{1000}. \quad (55)$$

Параллельно вычисляются и частные показатели по каждому режиму. Это удобно, потому что сразу видно, какая часть суточного графика даёт основную нагрузку.

Свойства и ограничения модели можно сформулировать так: она детерминирована, то есть при одинаковых входных параметрах всегда даёт один и тот же результат; обладает инженерной интерпретируемостью, поскольку каждый блок связан с конкретным физическим элементом системы – нагрузкой, панелью, аккумулятором, инвертором или кабельной линией; часть параметров носит эмпирический характер и может уточняться под реальные условия объекта. Модель ограничена заданным суточным профилем нагрузки и не описывает произвольные нештатные режимы; кроме того, итоговые оценки чувствительны к исходным паспортным значениям оборудования и к выбранным коэффициентам потерь [11, 26].

Подводя итог, эта модель описывает не просто набор чисел, а связанный энергетический контур гибридной изолированной системы. В ней есть и суточный профиль потребления, и аккумуляторный резерв, и согласование с инвертором, и учёт потерь. За счёт этого она хорошо подходит для инженерного расчёта и для последующего сравнения режимов работы системы.

4.4 Математическая модель согласования фотоэлектрического поля, инверторов и аккумуляторного банка

В основе расчёта лежит детерминированная дискретная оптимизационная модель с ограничениями [17]. По сути, она решает две связанные задачи сразу: сначала подбирает наиболее плотную и при этом электрически корректную раскладку фотоэлектрических модулей на требуемой площадке, а затем согласует получившуюся генераторную часть с инверторами и аккумуляторным банком. Модель не ищет абстрактно «лучший» вариант вообще; она довольно приземлённо стремится к реальной, выполнимой конфигурации.

Геометрическая часть модели. Пусть кровля имеет размеры L и W , а фотоэлектрический модуль – длину l_p и ширину w_p . По краям кровли оставляется технологический отступ t , между модулями – зазор g . Тогда рабочие размеры поверхности составляют:

$$L_{\text{раб}} = L - 2 \cdot m, \quad W_{\text{раб}} = W - 2 \cdot m. \quad (56)$$

Число модулей, которое помещается вдоль длины и ширины на требуемой площадке, определяется целочисленным образом:

$$n_L = \left[\frac{L_{\text{раб}} + g}{l_p + g} \right], \quad n_W = \left[\frac{W_{\text{раб}} + g}{w_p + g} \right]. \quad (57)$$

Тогда общее количество устанавливаемых модулей равно:

$$N_p = n_L \cdot n_W. \quad (58)$$

На практике рассматриваются две ориентации модуля – «портретная» и «альбомная». Для каждой из них размеры l_p и w_p просто меняются местами, после чего выбирается вариант с наибольшим числом модулей. На первый взгляд это выглядит как обычный перебор, но именно здесь и кроется основная оптимизация: модель не допускает лишних компромиссов по площади.

Согласованность ориентации солнечных панелей. После геометрической раскладки проверяется, можно ли соединить солнечные панели в строки так, чтобы напряжение каждой строки попадало в допустимый диапазон инвертора. Для этого длина строки s должна удовлетворять условиям:

$$s_{\min} = \left[\frac{\max(V_{MPPT,\min}, V_{start})}{V_{mp}} \right], \quad (59)$$

$$s_{\max} = \left[\frac{\min(V_{MPPT,\max}, V_{PV,\max})}{V_{oc}} \right],$$

где $V_{PV,\max}$ – напряжение в точке максимальной мощности одного модуля, В;

V_{oc} – напряжение холостого хода, В;

$V_{MPPT,\min}, V_{MPPT,\max}$ – границы рабочего диапазона MPPT, В;

V_{start} – порог запуска инвертора, В.

Допустимой считается только та длина строки, которая делит число модулей в ряду без остатка:

$$n_L \bmod s = 0. \quad (60)$$

Если подходящее значение s найдено, то число строк на один ряд составляет $q = \frac{n_L}{s}$, а общее число строк – $N_s = n_W \cdot q$.

При этом отбирается не любая подходящая длина строки, а максимально возможная из допустимых. Это уменьшает число строк и упрощает последую-

щую электрическую разводку. Стоит отметить, что при одинаковом числе модулей предпочтение отдаётся конфигурации с меньшим числом строк – она обычно аккуратнее с точки зрения компоновки и менее перегружает инверторную часть.

Формально задача выбора лучшей компоновки сводится к максимизации количества установленных модулей при выполнении электрических ограничений: $\max N_p$ при условиях существования корректной длины строки, допустимого количества строк и соблюдения всех пределов по напряжению и току. Если несколько вариантов дают одинаковое число модулей, выбирается тот, где электрическая часть согласована лучше и число строк меньше. Иначе говоря, модель сначала смотрит на площадь, потом – на жизнеспособность схемы, и только потом на «красоту» структуры.

Подбор числа инверторов. После формирования фотоэлектрического поля оценивается необходимое число инверторов. Оно определяется сразу по двум каналам: по нагрузке и по стороне генерации. По нагрузке:

$$n_{inv}^{(P)} = \left\lceil \frac{P_{inv,req}}{P_{inv}} \right\rceil, \quad (61)$$

где $P_{inv,req}$ – требуемая мощность системы, кВт;

P_{inv} – номинальная мощность одного инвертора, кВт.

По фотоэлектрической стороне сначала оценивается ограничение по числу строк на один инвертор:

$$n_{str}^{MPPT} = \left\lceil \frac{I_{MPPT,max}}{I_{sc}} \right\rceil, \quad (62)$$

$$n_{str}^{inv} = m_{MPPT} \cdot n_{str}^{MPPT},$$

где m_{MPPT} – число MPPT-входов;

I_{sc} – ток короткого замыкания одного модуля.

Тогда минимально необходимое число инверторов по строкам:

$$n_{inv}^{(S)} = \left\lceil \frac{N_s}{n_{str}^{inv}} \right\rceil. \quad (63)$$

Дополнительно учитывается мощность фотоэлектрического поля:

$$P_{PV} = N_p \cdot \frac{V_{mp} \cdot I_{mp}}{1000}, \quad (64)$$

$$n_{inv}^{(PV)} = \left[\frac{P_{PV}}{P_{AC}} \right].$$

Итоговое количество инверторов берётся как:

$$n_{inv} = \max \left(n_{inv}^{(P)}, n_{inv}^{(S)}, n_{inv}^{(PV)} \right). \quad (65)$$

После чего проверяется верхняя граница допустимого параллельного включения.

Так же вычисляется суммарная номинальная мощность всех инверторов в связке:

$$P_{tot} = n_{inv} \cdot P_{inv}. \quad (66)$$

Подбор аккумуляторного банка (АКБ). Аккумуляторная часть строится вокруг напряжения одного модуля V_b . Число последовательно соединённых модулей на один инвертор определяется двумя условиями: по допустимому диапазону напряжения инвертора и по току разряда.

Минимальное число модулей в серии:

$$k_{s,min} = \left[\frac{V_{bat,min}}{V_b} \right]. \quad (67)$$

Оценка по току:

$$k_{s,I} = \left[\frac{P_{inv}/n_{inv}}{I_{dis,max} \cdot V_b \cdot \eta_{inv}} \right], \quad (68)$$

где $I_{dis,max}$ – допустимый разрядный ток инвертора, А;

η_{inv} – КПД инвертора, %.

Тогда определение числа аккумуляторов определяется по формуле:

$$k_s = \max(k_{s,min}, k_{s,I}), \quad (69)$$

с ограничением сверху:

$$k_s \leq \left[\frac{V_{bat,max}}{V_b} \right].$$

Напряжение аккумуляторного банка равно:

$$V_{bank} = k_s \cdot V_b. \quad (70)$$

Далее определяется число параллельных ветвей. Определение данного числа по току:

$$n_{bat,I} = \left[\frac{I_{inv}}{I_{sting}} \right], \quad (71)$$

где I_{string} – допустимый ток одной ветви ($c_{bat} \cdot k_{ch}$), А.

Определение параллельных ветвей по энергии:

$$n_{bat,E} = \left\lceil \frac{E_{day}/n_{inv}}{E_{string}} \right\rceil, \quad (72)$$

где E_{string} – энергия одной строки АКБ ($k_s \cdot c_{bat} \cdot V_b \cdot (1 - C_{DoD}) \cdot k_{mod}$), Вт·ч.

Тогда итоговое число параллельных ветвей на один инвертор:

$$n_{bat} = \max(n_{bat,I}, n_{bat,E}, 1). \quad (73)$$

Полная номинальная емкость АКБ выражается по формуле (48), а полезная емкость учитывает ограничение по глубине разряда и режиму эксплуатации по второй формуле (49). Энергетическая емкость банка АКБ тогда задается как:

$$E_{bank} = C_{lim} \cdot V_{bank}. \quad (74)$$

Под конец модель проверяет, проходит ли вся связка по ключевым ограничениям:

$$\begin{aligned} P_{tot} &\geq P_{inv,req}, \\ V_{bat,min} &\leq V_{bank} \leq V_{bat,max}, \\ I_{bat,inv} &\leq I_{dis,max}, \\ E_{bank} &\geq E_{day} \cdot T_{aut}, \\ S_{min} &< S < S_{max}, \end{aligned} \quad (75)$$

где $I_{bat,inv}$ – ток разряда батареи, приходящийся на один инвертор, А;

T_{aut} – требуемая автономия, сут.

Ток разряда батареи, приходящийся на один инвертор, определяется через мощность, передаваемую на данный инвертор, напряжение аккумуляторного банка и КПД преобразования. При равномерном распределении нагрузки между инверторами расчетная зависимость принимает вид:

$$I_{bat,inv} = \frac{P_{inv,req}}{n_{inv} \cdot V_{bank} \cdot \eta_{inv}}. \quad (76)$$

Ток на входе МРРТ определяется суммарным током параллельно подключённых фотоэлектрических строк. В рабочем режиме он оценивается по току в точке максимальной мощности, а при проверке запаса по входу МРРТ – по

току короткого замыкания. Для обеспечения корректной работы системы должны выполняться дополнительные условия:

$$I_{MPPT,act} \leq I_{MPPT,max}, I_{sc,act} \leq I_{sc,max}, \quad (77)$$

где $I_{MPPT,act}$ – ток короткого замыкания по контролеру МРРТ, А;

$I_{sc,act}$ – номинальный ток короткого замыкания МРРТ, А;

$I_{sc,max}$ – предельно допустимый ток короткого замыкания МРРТ, А.

Если к одному МРРТ подключено параллельных строк n_{str}^{MPPT} , то токи в параллели складываются. Поэтому фактический ток на входе МРРТ в рабочей точке можно записать так:

$$I_{MPPT,act} = n_{str}^{MPPT} \cdot I_{mp}. \quad (78)$$

Здесь идея простая: напряжение у параллельных строк остаётся почти одинаковым, а токи суммируются. Именно поэтому МРРТ нужно проверять не только по напряжению строки, но и по суммарному входному току.

Для более осторожной оценки часто используют ток короткого замыкания, потому что он выше рабочего тока. Тогда расчётный ток по МРРТ принимают как:

$$I_{sc,act} = n_{str}^{MPPT} \cdot I_{sc}. \quad (79)$$

Если хотя бы одно из условий (75) и (77) нарушается, конфигурация считается непригодной и модель пересматривает длину строки, число инверторов или число аккумуляторных ветвей. Подводя итог, можно сказать, что модель сочетает в себе дискретную оптимизацию, инженерное ограничительное проектирование и энергетический баланс. Она не просто считает параметры, а выстраивает согласованную систему, где геометрия, электрические ограничения и запас энергии работают вместе.

4.5 Модель энергетического баланса ИГЭС

В данном разделе корректнее говорить о дискретной детерминированной динамической модели энергетического баланса. Она описывает, как система ведёт себя на каждом часовом интервале: сначала солнечная генерация покрывает

вает нагрузку, потом подключается аккумулятор, и лишь затем, при необходимости, в работу входит внешняя сеть для подзарядки. Модель не пытается усложнить картину лишними предположениями. Она просто честно считает баланс.

Все расчёты в модели выполняются по часовым интервалам суточного диапазона, то есть для каждого часа рассматривается отдельное состояние системы $i = 1, \dots, N$. Это позволяет последовательно оценивать изменение солнечной генерации, нагрузки, заряда аккумулятора и обмена энергией между элементами схемы без привязки к непрерывному времени. Иначе говоря, каждый час выступает как самостоятельный шаг дискретизации, на котором и строится энергетический баланс.

Пусть расчёт ведётся по шагам с длительностью интервала h_{int} . Тогда суточный режим нагрузки удобно задавать в виде вектора P_{load} , а его отдельные уровни – через характерные мощности максимального, обычного и минимального режимов. Сначала вводится средняя часовая мощность максимального режима по формуле (39). Далее обычная и минимальная нагрузки выражаются через доли от этого уровня по формулам (41) и (42).

Суточная энергия нагрузки определяется суммированием по всем шагам (на основе формулы (47)):

$$E_{day} = \sum_{i=1}^N P_{load,i} \cdot h_{int}. \quad (80)$$

Если шаг равен одному часу, запись становится особенно наглядной. Каждый интервал просто добавляет свой вклад в общий расход энергии.

Фотоэлектрическая часть системы начинается с базовых геометрических и энергетических параметров модуля, которые ранее были определены в разделе 4.3 [27]. На первый взгляд они выглядят совсем просто, но именно на них держится весь расчёт выработки. Активная площадь одной солнечной ячейки определяется через её линейные размеры:

$$S_{cell} = a \cdot b. \quad (81)$$

Чем больше площадь одной ячейки, тем больше энергии она способна принять и преобразовать. Тогда активная площадь одного модуля выражается как:

$$S_{mod} = S_{cell} \cdot N_{cell}. \quad (82)$$

Примечательно, что именно эта величина задаёт масштаб самого фотоэлектрического элемента: не отдельной ячейки, а уже законченного модуля, который участвует в формировании массива.

Если теперь учитывать число установленных панелей, то суммарная активная площадь фотоэлектрического поля записывается так:

$$S_{PV} = S_{mod} \cdot N_{pan}. \quad (83)$$

Иными словами, это уже не площадь одного элемента, а вся рабочая площадь генератора целиком. Для модели это ключевая величина, потому что именно она связывает солнечное излучение с возможной электрической мощностью.

Солнечная часть модели строится на энергии излучения на единицу площади E , которая уже рассчитана по формуле (38). Мощность фотоэлектрического поля на DC-стороне можно записать так:

$$P_{PV,i} = E_i \cdot \eta_{ph} \cdot S_{PV}. \quad (84)$$

Это простая часть модели, но без неё не собрать корректную мощность поля. После генерации постоянного тока учитываются потери на DC-кабелях:

$$P_{PV,i}^{dc} = P_{PV,i} \cdot k_{cab_{pv}}. \quad (85)$$

После верхних вычислений, поток проходит через инвертор, и доступная мощность на переменной стороне вычисляется по выражению:

$$P_{PV,i}^{ac} = P_{PV,i} \cdot \eta_{inv} \cdot k_{cab_{ac}}. \quad (86)$$

Занимательно то, что именно эта мощность из формулы (86) потом сравнивается с текущей нагрузкой. Сначала солнечная генерация покрывает нагрузку напрямую:

$$P_{PV,i}^{load} = \min(P_{load,i}, P_{PV,i}^{ac}). \quad (87)$$

После появляется остаточный дефицит. Как в выражении:

$$P_{PV,i}^{def} = \max(p_{load,i} - P_{PV,i}^{load}, 0). \quad (88)$$

Если генерация, наоборот, выше потребления, то возникает избыток:

$$P_{PV,i}^{sur} = \max\left(P_{PV,i}^{dc} - \frac{P_{PV,i}^{load}}{\eta_{inv} \cdot k_{cab_ac}}, 0\right). \quad (89)$$

Вот этот избыток уже можно направлять на заряд аккумуляторного банка. Или – АКБ. Но заряд не бывает произвольным – его ограничивают сразу несколько факторов: доступная мощность, допустимый ток заряда и свободный объём ёмкости. Поэтому фактическая мощность заряда задаётся по формуле:

$$P_i^{ch} = \max\left(P_{PV,i}^{sur} \cdot k_{bat}, P_{max}^{ch}, \frac{E_{bat,max} - E_{i-1}}{h_{int}}\right), \quad (90)$$

где P_{max}^{ch} связано с максимально допустимым током заряда: $I_{ch,max} \cdot V_{bank}$, кВт.

Энергетические границы банка записываются через V_{bank} :

$$E_{bat,max} = C_{lim} \cdot V_{bank}, \quad E_{bat,min} = C_{DoD} \cdot V_{bank}. \quad (91)$$

Именно $E_{bat,min}$ выступает контрольным порогом, при достижении которого логика управления начинает рассматривать подключение сети. Ниже $E_{bat,min}$ АКБ не желательно разряжаться, так как это может привести к необратимой деградации и нарушению баланса всей ИГЭС. Если после прямого питания нагрузка всё ещё не закрыта, аккумулятор переходит в разряд:

$$P_i^{dis} = \max\left(\frac{P_{PV,i}^{def}}{\eta_{inv} \cdot k_{cab_ac} \cdot k_{batt}}, P_{max}^{dis}, \frac{E_{bat,min} - E_{i-1}}{h_{int}}\right). \quad (92)$$

Фактическая мощность, отдаваемая батареей на АС-сторону, равна:

$$P_{PV,i}^{ac} = P_i^{dis} \cdot \eta_{inv} \cdot k_{cab_ac} \cdot k_{batt}. \quad (93)$$

Именно она закрывает остаток нагрузки после солнечной генерации. Сначала работает PV. Потом – батарея. Уже потом, если нужно, подключается сеть.

Состояние аккумулятора на каждом шаге обновляется по балансовому уравнению:

$$E_i = E_{i-1} + (P_i^{ch} + P_i^{grid} - P_i^{dis}) \cdot h_{int}. \quad (94)$$

Здесь P_i^{grid} – мощность, поступающая из внешней сети на заряд АКБ. Она включается не всегда, а только в том случае, когда солнечного избытка нет и аккумулятор опустился до контрольного минимума. На практике это выглядит довольно просто: если батарея села ниже порога, сеть подхватывает её заряд. Если нет – сеть «молчит»:

$$P_i^{grid} = \begin{cases} 0, & \text{если заряд от PV есть} \\ 0, & \text{если } E_i^{af} > E_{bat,min} \\ \min\left(P_{max}^{ch}, \frac{E_{bat,max} - E_i^{af}}{h_{int}}\right), & \text{если } E_i^{af} \leq E_{bat,min} \text{ и } P_{PV}^{ch} = 0 \end{cases}, \quad (95)$$

где E_i^{af} – энергия АКБ после возможного разряда на текущем шаге, Вт·ч.

Итоговая мощность, реально покрытая системой, определяется как:

$$P_i^{serv} = P_{PV,i}^{load} + P_{PV,i}^{ac}. \quad (96)$$

После этого остаточный дефицит записывается так:

$$P_{bat,i}^{def} = \max(P_{load,i} - P_i^{serv}, 0). \quad (97)$$

Если эта величина положительна, значит в рассматриваемом интервале система не справилась с нагрузкой полностью. Если равна нулю, баланс закрыт [8].

Для оценки накопленной выработки и суммарного потребления используются обычные суммы по всем интервалам:

$$E_{day}^{PV} = \sum_{i=1}^N P_{PV,i}^{dc} \cdot h_{int}, \quad E_{day}^{load} = \sum_{i=1}^N P_{load,i} \cdot h_{int}. \quad (98)$$

Аналогично можно вычислить суммарный заряд АКБ, суммарный разряд и объём энергии, взятый из сети:

$$E_{day}^{grid} = \sum_{i=1}^N P_i^{grid} \cdot h_{int}. \quad (99)$$

Данная модель описывает гибридную изолированную (солнечно-аккумуляторную) энергетическую систему как последовательность дискретных энергетических состояний, в которых каждый час закрывается свой баланс [28].

5 Анализ результатов и интерпретация

5.1 Интерпретация данных Цифрового Двойника в ИГЭС

Когда математические модели отлажены, а вычислительное ядро готово, начинается самый важный этап – осмысление того, что именно цифровой двойник сообщает об объекте. Здесь интересен не просто перечень цифр, а та внутренняя логика, которую расчёты проявляют для горнолыжной базы «Холдоми». Цифровой двойник изолированной гибридной энергетической системы изначально задумывался как инструмент, позволяющий заглянуть в будущее энергобаланса, не дожидаясь физического монтажа оборудования.

Разработанный комплекс – он не только для узких спецов по солнечной энергетике. Подачу намеренно упростили, добавили сценарный подход, так что заглянуть в результаты может и инженер на объекте, и экономист, и управленец – любой, кто принимает решения о закупке оборудования и режимах эксплуатации. Скажем прямо: полноценный цифровой двойник с двусторонней связью и моментальной реакцией на отказы – штука архисложная. А для объекта, где реальная система в «железе» ещё не существует, такая глубина просто избыточна. Можно было бы наvertеть десятки нештатных режимов, ввести стохастические факторы... только модель немедленно превратится в «чёрный ящик», понятный одному разработчику. Упрощение здесь – осознанный инженерный шаг. Оно позволяет любому заинтересованному, даже не посвящённому в тонкости фотоэлектрического преобразования, проследить цепочку «погода – генерация – накопление – дефицит» и сделать из увиденного практически выводы.

Главное, на что смотрят при интерпретации – баланс и дефицит. Модель последовательно, час за часом, гонит систему через три этапа покрытия нагрузки: сперва солнечная генерация, потом накопленная энергия аккумуляторов и, наконец, внешняя сеть. Механизм исправно обрабатывает и в тестовых, и в граничных условиях. Расчёты фиксируют, в какие именно часы даже мощного солнечного поля не хватает, чтобы закрыть пиковые потребности объекта, – в

случае ГК «Холдоми» это, в основном, вечернее время работы в высокий сезон. Тут-то в дело и вступает батарейный банк. И только когда его резерв опускается до критического минимума, алгоритм подключает резервный ввод от центральной сети. Такой прозрачный каскад – работать с ним одно удовольствие [6, 16]. Жёсткой привязки к физическим датчикам и обратным связям нет, но прогноз от этого не делается бесполезным. Наоборот, двойник превращается в испытательный полигон без риска.

А если представить этот процесс графически – рисунок 5.1, – картина становится особенно наглядной. Именно визуальные образы, а не сухие колонки чисел, позволяют быстро оценить, насколько велика потребность в резервировании. По сути, один взгляд на такую картинку заменяет долгое копание в таблицах. Пользователь цифрового двойника, глядя на такие кривые, способен моментально понять: в январе система полностью автономна только при существенном запасе ёмкости, а в межсезонье и летом, при низкой нагрузке и ясном небе, зависимость от сети может быть почти нулевой [42].

Отдельного упоминания заслуживает то, как двойник интерпретирует сезонную специфику «Холдоми». Эта горнолыжная база интересна не только удалённостью, но и резко меняющейся нагрузкой. Зимой и летом, в пики сезона, потребление подскакивает до расчётных максимумов, а в мёртвый сезон объекты базы, по сути, потребляют минимум на освещение и вентиляцию. Цифровой двойник даёт возможность сравнивать эти сценарии друг с другом, не прибегая к отдельным расчётам. При этом видно, что робастность модели проявляется как раз в такие моменты перепадов. Заложенные в неё зависимости не ломаются и не уходят в нереалистичные отрицательные значения – даже если температура падает до минус сорока, а сплошная облачность срезает прямую радиацию почти до нуля. Двойник честно показывает: в этот день придётся рассчитывать только на аккумуляторные батареи и сеть.

Солнечный поток с учётом облачности



Выработанная и необходимая энергия



Дефицит и покрытие АКБ

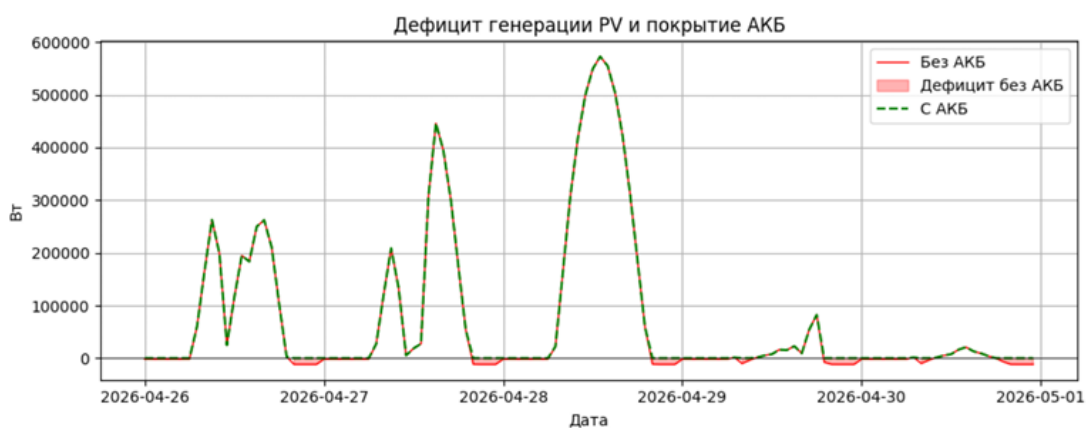


Рисунок 5.1 – Графики баланса и дефицита

Когда данные, полученные от двойника, ложатся на стол специалиста по эксплуатации, они превращаются в конкретные цифры для планирования. Становится ясно, какой запас ёмкости действительно востребован, а какой – излишен. Видна доля участия каждого источника в общем балансе. Примечательно, что по мере накопления статистики и возможного появления в будущем реаль-

ных данных с объекта, базовая имитационная модель может быть откалибрована заново. Но и сейчас, без физического канала обратной связи, она позволяет оценить порядок величин, уловить узкие места и, главное, понять, не будет ли система хронически зависеть от неожиданных перетоков из сети.

5.2 Веб-приложение для ЦД ИГЭС и его испытание

После того как расчетная модель цифрового двойника была собрана и проверена на уровне алгоритмов, понадобился удобный инструмент, через который с ней можно работать без лишних проблем [20, 28]. Так появилось веб-приложение для ЦД ИГЭС – не просто оболочка для запуска расчета, а живой интерфейс, в котором все ключевые параметры собраны в одном месте и подаются в понятной форме. Это хорошо совпадает с идеей самой работы: цифровой двойник здесь задуман как автономная имитационная модель, которая получает метеоданные из внешнего источника, а затем выдает показатели выработки, накопления и дефицита энергии; при этом архитектура модели оставлена с запасом на будущее расширение, если на объекте появится физическая система и каналы обратной связи.

Для создания приложения был выбран Python-стек, потому что он позволил собрать все в одном рабочем контуре без тяжелой инфраструктуры. Веб-часть удобно строится через Streamlit (библиотека Python с открытым исходным кодом позволяющая создавать интерактивные веб-приложения и дашборды, используя только код на Python): экран получается понятным, с боковой панелью параметров, кнопкой запуска расчета и моментальным обновлением результата. Для выбора географической точки пригодилась интерактивная карта, а для отображения графиков – библиотека для построения диаграмм; обработка временных рядов и таблиц естественно легла на инструменты для анализа данных, а локальное хранение промежуточной информации – на простую базу SQLite. На первый взгляд набор инструментов выглядит скромно, но именно в этом и есть его преимущество: приложение не перегружено, быстро запускает-

ся и не требует сложной настройки на каждом компьютере. Это особенно уместно для предпроектного анализа, где важны скорость проверки гипотез и ясность интерфейса. Open-Meteo при этом выбран не случайно: в предыдущих разделах прямо отмечены его свободный доступ, возможность программного получения рядов данных, достаточная точность для предпроектных расчетов и применимость для удаленных регионов.

Внешний вид приложения построен так, чтобы пользователь не терялся в параметрах. Слева расположена панель ввода, где задаются координаты объекта, часовой пояс, период моделирования, нагрузка, характеристики солнечной панели, аккумулятора, инвертора, потери в цепях и размеры крыши. В отдельном блоке задается начальный заряд аккумуляторной батареи, что немаловажно для проверки разных сценариев работы системы. Такой подход делает интерфейс похожим на инженерный пульт управления: все нужные настройки находятся рядом, и не приходится открывать несколько окон, чтобы изменить хотя бы одну величину. В основном окне приложения выводится заголовок, краткая подсказка и кнопка запуска расчета, как показано на рисунке 5.2.

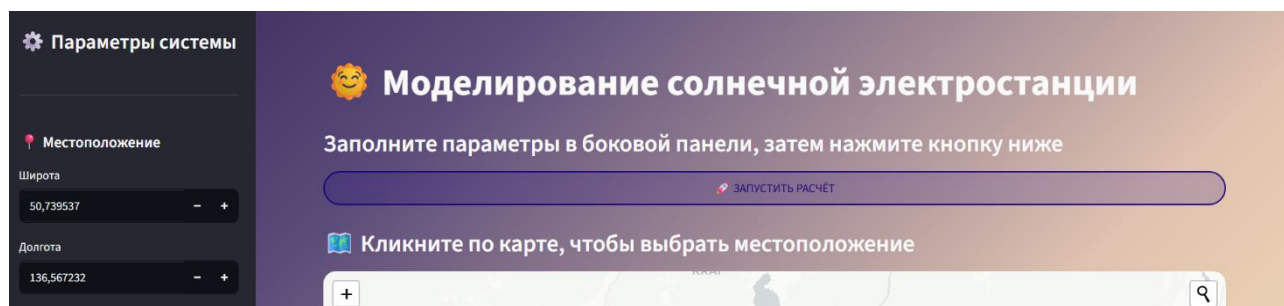


Рисунок 5.2 – Стартовый экран приложения

Отдельного внимания заслуживает карта. В приложении она нужна для практической привязки объекта к местности, если это необходимо пользователю. Пользователь может щелкнуть по нужной точке, после чего координаты автоматически подставляются в поля ввода. Это очень удобный момент: солнечная генерация напрямую зависит от широты и долготы, а значит, даже небольшое смещение точки способно изменить итоговые значения. На экране карта

выступает как простой и наглядный способ задать место моделирования, без ручного переписывания чисел (рисунок 5.3).

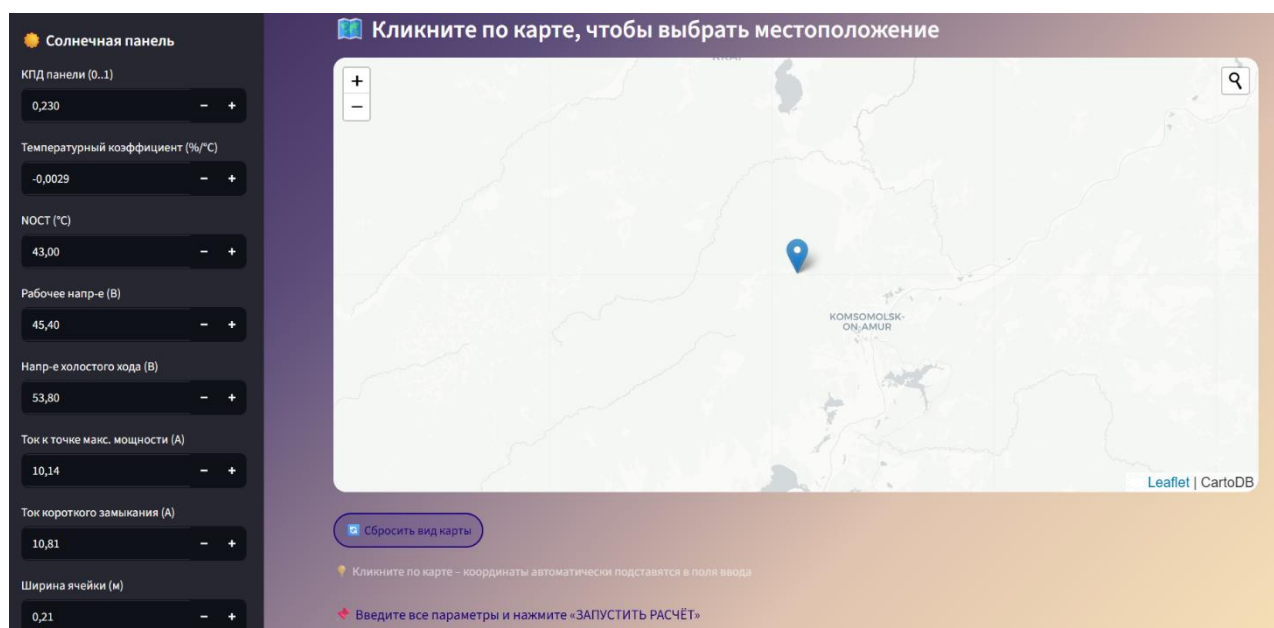


Рисунок 5.3 – Карта выбора местоположения

Когда пользователь нажимает кнопку запуска, приложение скрывает карту и переходит к расчету. Сначала оно получает метеоданные для выбранного интервала, затем приводит временные метки к единому виду и сохраняет их в локальную базу. Далее рассчитываются солнечный поток, выработка фотоэлектрической части, работа аккумулятора, баланс нагрузки и потребление из сети. Логика здесь построена довольно аккуратно: входные параметры собираются в одну структуру, а итоговые данные появляются уже в виде набора графиков и сводных метрик. Это хорошо согласуется с тем, как в самой работе описан цифровой двойник: сначала идет сбор и нормализация погодной информации, потом – вычисление солнечного потока, а затем – балансовый расчет по часам.

В блоке результатов приложение показывает несколько графиков, они представлены на рисунке 5.4. Первый из них сравнивает выработанную и требуемую энергию, второй отражает изменение энергии в аккумуляторе, третий демонстрирует заряд и разряд батареи, а четвертый наглядно показывает дефицит генерации с учетом и без учета накопителя. И вот здесь как раз становится

видно, зачем вообще был нужен цифровой двойник: он не просто выдает сухую таблицу чисел, а сразу показывает, где система справляется, а где начинает проседать. Для пользователя это очень удобно – график читается быстрее, чем длинный отчет, и лучше помогает уловить проблемные часы.



Рисунок 5.4 – Результаты моделирования в виде графиков

Сводная часть результатов тоже получилась полезной. Приложение выводит пиковый дефицит после учета аккумулятора, суммарное потребление из сети и конечный заряд батареи. Эти показатели дают понимание, насколько выбранная конфигурация жизнеспособна и нужен ли дополнительный запас по мощности или емкости. Кроме того, предусмотрена выгрузка PDF-отчета по нажатию кнопки, и это очень практичная деталь: инструмент начинает работать не как разовая демонстрация, а как полноценный рабочий модуль. Данная часть веб-приложения показана на рисунке 5.5.

Испытание приложения проводилось на нескольких сценариях с изменением координат, периода расчета и параметров оборудования. На рисунках 5.2 – 5.5 отображена проверка моделирования в зимний период с 01.01.2026 по 07.01.2026 гг. На граничных режимах модель ведет себя устойчиво: при сильной облачности и низкой температуре она не «ломается», не уходит в отрицательные значения и честно показывает, когда системе уже приходится опираться

ся на аккумулятор и внешнюю сеть. Это важное качество, потому что робастность в такой задаче – не только про теорию, но и про способность модели оставаться полезной в неудобных и неидеальных условиях. В тексте работы прямо отмечено, что даже при резких перепадах погоды и сезонной нагрузки цифровой двойник позволяет сравнивать сценарии без отдельных ручных расчетов, а по мере появления новых данных его можно будет донастроить и уточнить.

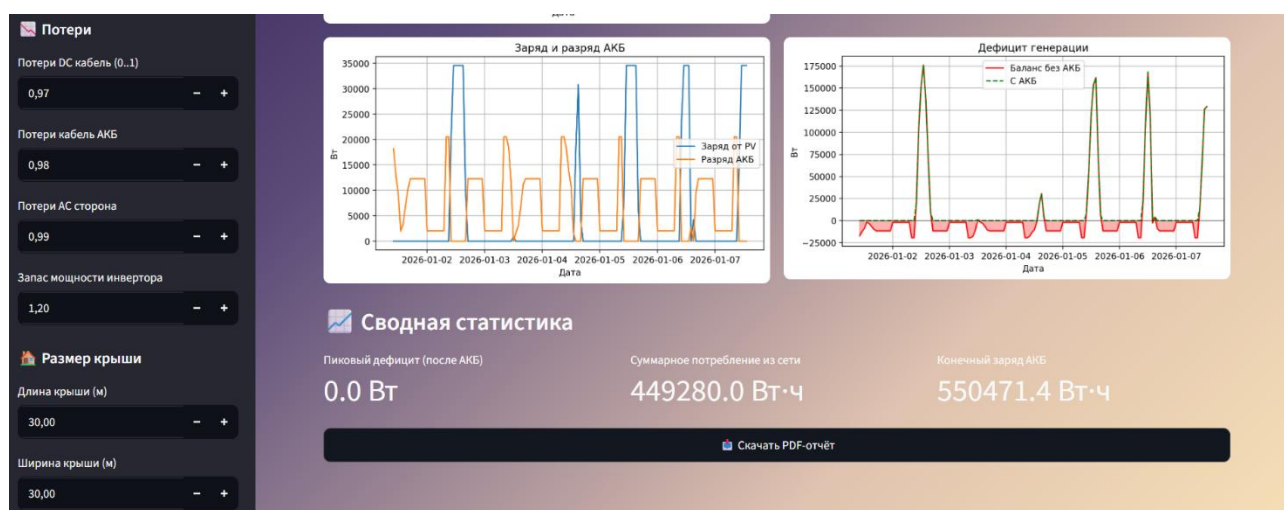


Рисунок 5.5 – Сводная статистика и кнопка выгрузки PDF-отчета

Подводя итог, веб-приложение для ЦД ИГЭС стало удобной формой представления расчетной модели и одновременно ее испытательным полигоном. Оно объединяет ввод исходных данных, выбор местоположения, автоматический расчет, визуализацию результатов и выгрузку отчета в один понятный рабочий процесс.

5.3 Оценка достижения поставленных целей и задач

Возвращаясь к целям, сформулированным в начале работы, нужно признать, что созданный инструмент не является классическим эксплуатационным двойником. И не должен им быть – на данном этапе. Он решает задачу предпроектного анализа и имитационного моделирования, то есть именно ту задачу,

которая была поставлена для нынешней стадии проекта. Отсутствие реальной энергосистемы на объекте не позволило замкнуть контур обратной связи, и это, конечно, накладывает свой отпечаток. Однако даже в таком виде двойник даёт несравненно больше, чем статичная таблица в пояснительной записке.

Инженерная ценность того, что получилось, укладывается в несколько тезисов. Модель убедительно показала: сочетание панелей, подобранных под реальную геометрию крыши, и литий-железо-фосфатных модульных накопителей способно закрыть внушительную долю сезонной нагрузки. Не сто процентов. И это нормально. Важно другое – видны границы, за которыми начинается дефицит. Дальше: алгоритм раскладки панелей с учётом электрических ограничений – это не головоломка ради красивой картинки. Это та самая деталь, которая приближает виртуальную копию к реальной установке оборудования и запуску его в работу. Двойник уже сейчас отвечает на вопрос «влезут ли эти панели на эту крышу и будет ли от них толк?» – и отвечает однозначно, опираясь на конкретные цифры.

С точки зрения робастности подход показал себя состоятельным. Модель устойчиво работает при экстремальных входных параметрах, не теряя адекватности. Она не стесняется показывать отрицательные сценарии – длительные периоды низкой инсоляции, когда аккумуляторный банк выжимается до предела. Но и угрозы не преувеличивает.

Если подытожить, в этой работе удалось заложить фундамент цифрового двойника изолированной гибридной энергосистемы для горнолыжной базы «Холдоми». Следующий логический шаг – развёртывание реального оборудования и подключение к нему аппаратного канала двойника. Задача, которая естественно вырастает из полученных результатов, а не висит в пустоте. И, возможно, когда на склонах ГК «Холдоми» появятся реальные солнечные панели, им не придётся работать вслепую – их поведение будет заранее понято, измерено и предсказано благодаря той математической базе, что описана в этой дипломной работе.

Заключение

В ходе работы была решена поставленная задача по разработке робастного цифрового двойника изолированной гибридной энергетической системы (на примере горнолыжной базы «Холдоми»). Тема оказалась не просто актуальной, а действительно практичной: для удалённых объектов с сезонной нагрузкой, переменной погодой и высокой ценой ошибки цифровой двойник становится удобным инструментом для анализа, прогноза и выбора режима энергоснабжения.

Поставленная цель так же была достигнута. В работе последовательно уточнены ключевые понятия, связанные с цифровым двойником, гибридной и изолированной энергетической системой, а также с робастностью модели. Это позволило выстроить не абстрактное описание, а понятную терминологическую и инженерную основу для дальнейшего проектирования. Цифровой двойник здесь – не просто имитация. Это динамическая система, которая умеет учитывать погодные качели, сезонные скачки нагрузки и всю специфику удалённого объекта.

Попутно решались и другие задачи. Обосновали выбор базы «Холдоми» – показали, почему ей нужна автономка или хотя бы полуавтономная схема. Подобрали железо: солнечные панели, аккумуляторы, инверторы. Отдельно копались в метеоданных – облачность и температура напрямую рулят выработкой, так что источники и интеграция в модель были принципиальны. На всём этом построили математическую модель: приход солнца, работа накопителя, часовой баланс мощности.

На выходе получилось веб-приложение для двойника ИГЭС. В нём собраны ввод параметров, тыканье по карте, запуск расчёта, графики и отчёты. Модель перестала быть просто схемой – стала рабочим инструментом, где можно прогонять разные сценарии. Приложение сразу выводит на экран выработку, потребление, состояние батареи, дефицит и долю замещения сети. Сла-

бые места видны мгновенно, и сразу ясно, где надо добавить мощности или ёмкости.

Испытания показали: модель устойчива, меняй координаты, интервал, параметры – держится. В плохую погоду не плывёт, не выдаёт бред. Для робастного двойника это ключевое. Результаты расчётов хорошо идут в предпроектный анализ: виден порядок энергопотоков, насколько активно работают аккумуляторы, как часто придётся обращаться к внешней сети.

Точку ставить рано. Работа не закрывает тему, а задаёт вектор. Архитектуру изначально заложили с прицелом на расширение: появится реальная установка – можно будет добавить аппаратный канал и канал калибровки, превратить имитацию в полноценный двойник. Дальше – верификация на живых данных, более точный учёт и, возможно, переход к управлению энергосистемой в реальном времени.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Abdelsattar, M., Mesalam, A., Diab, A. A. Z. et al. Optimal sizing of a proposed stand-alone hybrid energy system in a remote region of southwest Egypt applying different meta-heuristic algorithms // *Neural Computing and Applications*. – 2024. – Vol. 36. – P. 16251–16269.

2 Adeyinka, A. M., Esan, O. C., Ijaola, A. O. et al. Advancements in hybrid energy storage systems for enhancing renewable energy-to-grid integration // *Sustainable Energy Research*. – 2024. – Vol. 11. – Article No. 26.

3 Altshuller, G. 40 principles: TRIZ keys to technical innovation. – Worcester : Technical Innovation Center, Inc., 2002. – Vol. 1.

4 Amoussou, I., Paddy, E. Y., Agajie, T. F. et al. Enhancing residential energy access with optimized stand-alone hybrid solar-diesel-battery systems in Buea, Cameroon // *Scientific Reports*. – 2024. – Vol. 14. – Article No. 15543.

5 Åström, K. J. Model Uncertainty and Robust Control / K. J. Åström // *Lecture Notes on Iterative Identification and Control Design* / ed. P. Albertos, A. Sala. – Lund : Lund University, 2000. – P. 1–19.

6 Bayer, D. A., Bayer, M. P. Digital twin of a local energy system based on real smart meter data // *Energy Informatics*. – 2023. – Vol. 6. – Article No. 8.

7 El Boujdaini, L., Mezrhab, A., Moussaoui, M. A. et al. Sizing of a stand-alone PV–wind–battery–diesel hybrid energy system and optimal combination using a particle swarm optimization algorithm // *Electrical Engineering*. – 2022. – Vol. 104. – P. 3339–3359.

8 Grieves, M. Virtually Intelligent Product Systems: Digital and Physical Twins // *Complex Systems Engineering: Theory and Practice* / ed. S. Flumerfelt. – American Institute of Aeronautics and Astronautics, 2019. – C. 175–200.

9 Gurumoorthi, G., Senthilkumar, S., Karthikeyan, G. et al. A hybrid deep learning approach to solve optimal power flow problem in hybrid renewable energy systems // *Scientific Reports*. – 2024. – Vol. 14. – Article No. 19377.

10 Kandemir, E., Hasan, A., Kvamsdal, T. et al. Predictive digital twin for wind energy systems: a literature review // *Energy Informatics*. – 2024. – Vol. 7. – Article No. 68.

11 Maurizio, Lenzerini. Интеграция данных: теоретическая перспектива // *PODS*. – 2002. – С. 233-246.

12 Messenger, R. *Photovoltaic Systems Engineering* / Roger Messenger, Jerry Ventre. – 4th edition. – New York : CRC Press, 2017. – 536 p.

13 Mundu, M. M., Nnamchi, S. N., Sempewo, J. I. et al. Simulation modeling for energy systems analysis: a critical review // *Energy Informatics*. – 2024. – Vol. 7. – Article No. 75.

14 Peng, Y., Zhang, Y. Energy optimization of stand-alone electrical grid considering the optimal performance of the hydrogen storage system and consumers // *Journal of Engineering and Applied Science*. – 2024. – Vol. 71. – Article No. 61.

15 Qiu, K., Entchev, E. Modeling, design and optimization of integrated renewable energy systems for electrification in remote communities // *Sustainable Energy Research*. – 2024. – Vol. 11. – Article No. 10.

16 Sleiti, A. K., Kapat, J. S., Vesely, L. Digital twin in energy industry: Proposed robust digital twin for power plant and other complex capital-intensive large engineering systems // *Energy Reports*. – 2022. – Vol. 8. – P. 3704-3726.

17 Yu, S., You, L., Zhou, S. A review of optimization modeling and solution methods in renewable energy systems // *Frontiers of Engineering Management*. – 2023. – Vol. 10. – P. 640–671.

18 Yu, W. et al. Energy digital twin technology for industrial energy management: Classification, challenges and future // *Renewable and Sustainable Energy Reviews*. – 2022. – Vol. 161. – Article No. 112407.

19 Баженов, А. Н. Обработка и анализ интервальных данных / А. Н. Баженов, С. И. Жилин, С. И. Кумков, С. П. Шарый. – Москва ; Ижевск : Институт компьютерных исследований, 2024. – 356 с.

20 Бердоносков, В. Д. Информационно-математическое обеспечение цифрового двойника изолированной гибридной энергетической системы / В. Д.

Бердонос, Е. О. Шелопугина // Вычислительные технологии и прикладная математика : Материалы III научной конференции с международным участием, Комсомольск-на-Амуре, 07–11 октября 2024 года. – Комсомольск-на-Амуре : Комсомольский-на-Амуре государственный университет, 2024. – С. 60-64.

21 Бердонос, В. Д. Информационно-математическая поддержка цифрового двойника автономной гибридной энергетической системы / В. Д. Бердонос, О. Н. Фейгенсон, Е. О. Шелопугина // Вычислительные технологии и прикладная математика. СТАМ 2024. Труды Springer по математике и статистике / ред. А. Гибадуллин, С. Гордин. – Том 500. – Чам : Springer, 2025. – С. 1-15. – DOI: 10.1007/978-3-031-90546-9_1.

22 Виссарионов, В. И. Солнечная энергетика : учеб. пособие / В. И. Виссарионов, Г. В. Дерюгина, В. А. Кузнецова, Н. К. Малинин. – Москва : Издательский дом МЭИ, 2008. – 320 с.

23 ГОСТ Р 57700.37-2021. Компьютерные модели и моделирование. Цифровые двойники изделий. Общие положения : принят и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 16 сентября 2021 г. № 979-ст : введен впервые : дата введения 2022-01-01 / разработан Федеральным государственным унитарным предприятием "Российский федеральный ядерный центр - Всероссийский научно-исследовательский институт экспериментальной физики". – Москва : Госстандарт России : Изд-во стандартов, 2021. – IV, 9 с.

24 Захид, М. А. Расчёт оптимального угла наклона для выработки максимальной энергии от солнечной панели с использованием моделей ясного неба и интенсивности падающих лучей / М. А. Захид, Г. Т. Чаван, Я. Х. Чо, Д. Ж. Ы // Светотехника. – 2020. – № 5. – С. 67-72.

25 Исаков, С. В. Оценка поступления солнечной радиации на естественные поверхности с применением геоинформационных систем / С. В. Исаков, В. А. Шкляев // Географический вестник. – 2012. – № 1. – С. 72-80.

26 Ковалев, С. П. Применение нейронных сетей глубокого обучения в математическом обеспечении цифровых двойников электроэнергетических систем

/ С. П. Ковалев // Системы и средства информатики. – 2021. – Т. 31, № 1. – С. 133-144.

27 Ковалев, С. П. Применение онтологий при разработке распределенных автоматизированных информационно-измерительных систем / С. П. Ковалев // Автометрия. – 2008. – № 2. – С. 41-49.

28 Ковалев, С. П. Проектирование информационного обеспечения цифровых двойников энергетических систем / С. П. Ковалев // Системы и средства информатики. – 2020. – Т. 30, № 1. – С. 66-81.

29 Массель, Л. В. Семантическое моделирование при построении цифровых двойников энергетических объектов и систем / Л. В. Массель, А. Г. Массель // Онтология проектирования. – 2023. – Т. 13, № 1. – С. 44-54.

30 Митрофанов, С. В. Математическое моделирование гибридной ветро-солнечной станции для электроснабжения собственных нужд / С. В. Митрофанов, К. А. Перепелкин // Вестник Южно-Уральского государственного университета. Серия: Энергетика. – 2022. – № 2. – С. 18-26.

31 Патент № 2024Э03922 Российская Федерация. Программа расчёта посуточного потока солнечного излучения с учётом Альбеда: № 2024614614 : заявл. 19.02.2024 : опубл. 28.02.2024 / Шелопугина Е. О. ; заявитель Шелопугина Е. О. – 2 с.

32 Патент № 2024Э11790 Российская Федерация. Программа моделирования потока солнечного излучения с учётом уровня облачности и коэффициента Альбеда : № 2024660719 : заявл. 24.04.2024 : опубл. 08.05.2024 / Шелопугина Е. О. ; заявитель Шелопугина Е. О. – 2 с.

33 Патент № 2024Э17427 Российская Федерация. Программный комплекс цифрового двойника изолированной гибридной энергетической системы : № 2024664115 : заявл. 08.06.2024 : опубл. 17.06.2024 / Шелопугина Е. О. – 2 с.

34 Прохоров, А. М. Цифровой двойник. Анализ, тренды, мировой опыт / А. М. Прохоров, М. Н. Лысачев. – 1-е изд., исправ. и доп. / науч. ред. А. И. Боровков. – Москва : ООО «АльянсПринт», 2020. – 401 с.

35 Свидетельство о государственной регистрации программы для ЭВМ № 2025666897 Российская Федерация. Модуль коррекции данных цифрового двойника для обеспечения соответствия показаниям изолированной гибридной энергосистемы : заявл. 19.06.2025 : опубл. 01.07.2025 / Е. О. Шелопугина.

36 Тюхов, И. И. Мониторинг погодноклиматических условий для солнечной энергетики / И. И. Тюхов, А. Х. Раупов // Альтернативная энергетика и экология. – 2014. – № 2 (142). – С. 99-108.

37 Харченко, Н. В. Индивидуальные солнечные установки. – Москва : Энергоатомиздат, 1991. – 208 с.

38 Цифровые двойники в высокотехнологичной промышленности: монография / А. И. Боровков, Ю. А. Рябов, Л. А. Щербина [и др.]. – Санкт-Петербург : ПОЛИТЕХ-ПРЕСС, 2022. – 491 с.

39 Шелопугина, Е. О. Астрономический анализ: точные расчеты времени восхода и заката солнца с использованием библиотек языка программирования Python / Е. О. Шелопугина, Д. О. Аршинский, В. Д. Бердоносков // Наука, инновации и технологии: от идей к внедрению : Материалы Всероссийской научно-практической конференции молодых ученых, Комсомольск-на-Амуре, 16–17 ноября 2023 года. – Комсомольск-на-Амуре : Комсомольский-на-Амуре государственный университет, 2023. – С. 610-613.

40 Шелопугина, Е. О. Исследование потока солнечной энергии: моделирование и анализ с использованием языка программирования Python / Е. О. Шелопугина, В. Д. Бердоносков // Наука, инновации и технологии: от идей к внедрению : Материалы Всероссийской научно-практической конференции молодых ученых, Комсомольск-на-Амуре, 16–17 ноября 2023 года. – Комсомольск-на-Амуре : Комсомольский-на-Амуре государственный университет, 2023. – С. 614-616.

41 Шелопугина, Е. О. К вопросу о корректировке данных цифрового двойника ИГЭС на основе информации с экспериментальной установки / Е. О. Шелопугина, В. Д. Бердоносков // Молодёжь и наука: актуальные проблемы фундаментальных и прикладных исследований : Материалы VIII Всероссий-

ской национальной научной конференции молодых учёных, Комсомольск-на-Амуре, 07–11 апреля 2025 года. Часть 2. – Комсомольск-на-Амуре : Комсомольский-на-Амуре государственный университет, 2025. – С. 611-614.

42 Шелопугина, Е. О. Моделирование критически низких периодов солнечной активности в ИГЭС / Е. О. Шелопугина, В. Д. Бердоносков // молодёжь и наука: актуальные проблемы фундаментальных и прикладных исследований : материалы всероссийской науч.-практ. конф. молодых ученых, Комсомольск-на-Амуре, 8-12 ноября 2024 г. в. / редкол. : А. В. Космыцин (отв. ред.) [и др.]. – Комсомольск-на-Амуре : ФГБОУ ВО «КНАГУ», 2024. – С. 1-3.

43 Шелопугина, Е. О. Прогнозирование и оптимизация дефицита энергии в изолированной гибридной энергосистеме с помощью цифрового двойника / Е. О. Шелопугина, В. Д. Бердоносков // Молодёжь и наука: актуальные проблемы фундаментальных и прикладных исследований : Материалы VIII Всероссийской национальной научной конференции молодых учёных, Комсомольск-на-Амуре, 07–11 апреля 2025 года. Часть 2. – Комсомольск-на-Амуре : Комсомольский-на-Амуре государственный университет, 2025. – С. 607-611.

ПРИЛОЖЕНИЕ А

(обязательное)

Программный комплекс цифрового двойника изолированной гибридной энергетической системы в формате веб-приложения

```
### solar_calc.py ###
# -*- coding: utf-8 -*-
import sqlite3
import pandas as pd
import numpy as np
import pytz
import os
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
from tabulate import tabulate
import requests
import math
import warnings
from fpdf import FPDF

warnings.filterwarnings('ignore')

# = ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ (будут переопределены из params) =
DEFAULT_LAT = 50.739537
DEFAULT_LON = 136.567232
LOCAL_BASE_PATH = os.getcwd()

def get_unique_filename(base_path, filename):
    return os.path.join(base_path, filename)

def download_file(file_name):
    file_path = os.path.join(LOCAL_BASE_PATH, file_name)
    if not os.path.exists(file_path):
        raise FileNotFoundError(f"Файл {file_name} не найден в {LOCAL_BASE_PATH}.")
    return file_path

def upload_file_to_drive(file_name):
    pass

# = ФУНКЦИИ ПАРСИНГА ОБЛАЧНОСТИ =
API_ARCHIVE = "https://archive-api.open-meteo.com/v1/archive"
API_FORECAST = "https://api.open-meteo.com/v1/forecast"

def init_db(conn):
    cur = conn.cursor()
    cur.execute("""
```

```

CREATE TABLE IF NOT EXISTS screenshots (
    id            INTEGER PRIMARY KEY AUTOINCREMENT,
    site          TEXT     NOT NULL,
    datetime      TEXT     NOT NULL UNIQUE,
    datetime_utc  TEXT,
    clouds        REAL     NOT NULL,
    temperature   REAL     NOT NULL
) """)
conn.commit()
cur.execute("PRAGMA table_info(screenshots)")
existing = [r[1] for r in cur.fetchall()]
if 'datetime_utc' not in existing:
    try:
        cur.execute("ALTER TABLE screenshots ADD COLUMN datetime_utc
TEXT")
        conn.commit()
    except sqlite3.OperationalError:
        pass

def fetch_cloudcover(url, lat, lon, start_date, end_date):
    params = {
        "latitude": lat,
        "longitude": lon,
        "hourly":    "cloudcover,temperature_2m",
        "start_date": start_date,
        "end_date":  end_date,
        "timezone":  "UTC"
    }
    resp = requests.get(url, params=params)
    resp.raise_for_status()
    data = resp.json()
    times = data.get("hourly", {}).get("time", [])
    clouds = data.get("hourly", {}).get("cloudcover", [])
    temps  = data.get("hourly", {}).get("temperature_2m", [])
    records = list(zip(times, clouds, temps))
    records.sort(key=lambda x: x[0])
    return records

def insert_new(conn, site_url, records, start_utc=None, end_utc=None):
    DATA_TZ = pytz.UTC
    cur = conn.cursor()
    added = 0
    for t_iso, cloud, temp in records:
        if cloud is None or temp is None:
            continue
        if 'T' in t_iso:
            time_part = t_iso.split('T')[1]
            if len(time_part) == 5:
                t_full = t_iso + ":00"
            else:
                t_full = t_iso

```

```

else:
    t_full = t_iso + "T00:00:00"
try:
    dt_naive = datetime.fromisoformat(t_full)
    dt_utc = dt_naive.replace(tzinfo=pytz.UTC)
except Exception:
    try:
        dt_naive = datetime.strptime(t_full, "%Y-%m-%dT%H:%M:%S")
        dt_utc = pytz.UTC.localize(dt_naive)
    except Exception:
        continue
if start_utc is not None and dt_utc < start_utc:
    continue
if end_utc is not None and dt_utc > end_utc:
    continue
try:
    dt_local = dt_utc.astimezone(DATA_TZ)
except Exception:
    dt_local = dt_utc
dt_local_str = dt_local.strftime('%Y-%m-%d %H:%M:%S')
dt_utc_str = dt_utc.strftime('%Y-%m-%d %H:%M:%S')
cur.execute("SELECT 1 FROM screenshots WHERE datetime = ?",
(dt_local_str,))
if cur.fetchone():
    continue
cur.execute(
    "INSERT INTO screenshots (site, datetime, datetime_utc, clouds,
temperature) VALUES (?, ?, ?, ?, ?)",
    (site_url, dt_local_str, dt_utc_str, cloud, temp)
)
added += 1
conn.commit()
return added

def fetch_and_store_clouds(db_path, lat, lon, start_local_str,
end_local_str, local_tz_str="Asia/Vladivostok"):
    """Загружает данные облачности и температуры из open-meteo и сохраняет в
БД"""
    try:
        local_tz = pytz.timezone(local_tz_str)
    except:
        local_tz = pytz.UTC
    start_local_naive = datetime.fromisoformat(start_local_str)
    end_local_naive = datetime.fromisoformat(end_local_str)
    start_local = local_tz.localize(start_local_naive)
    end_local = local_tz.localize(end_local_naive)
    start_utc = start_local.astimezone(pytz.UTC)
    end_utc = end_local.astimezone(pytz.UTC)
    db_path = os.path.expanduser(db_path)
    db_dir = os.path.dirname(db_path)
    if db_dir:

```

```

        os.makedirs(db_dir, exist_ok=True)
    conn = sqlite3.connect(db_path)
    init_db(conn)
    today = datetime.now(pytz.UTC).date()
    req_start_date = start_local.date().isoformat()
    req_end_date = end_local.date().isoformat()
    total_added = 0
    if start_utc.date() < today:
        arch_end = min(end_utc.date(), today - timedelta(days=1))
        arch_records = fetch_cloudcover(API_ARCHIVE, lat, lon,
req_start_date, arch_end.isoformat())
        added = insert_new(conn, f"ARCHIVE {API_ARCHIVE}", arch_records,
start_utc=start_utc, end_utc=end_utc)
        total_added += added
    if end_utc.date() >= today:
        fc_start = max(start_utc.date(), today)
        fc_records = fetch_cloudcover(API_FORECAST, lat, lon,
fc_start.isoformat(), req_end_date)
        added = insert_new(conn, f"FORECAST {API_FORECAST}", fc_records,
start_utc=start_utc, end_utc=end_utc)
        total_added += added
    conn.close()
    return db_path

# = ФУНКЦИИ АСТРОНОМИИ И РАСЧЁТА ЭНЕРГИИ =
def _julian_day(dt):
    Y = dt.year; M = dt.month; D = dt.day + (dt.hour + dt.minute/60.0 +
dt.second/3600.0)/24.0
    if M <= 2: Y -= 1; M += 12
    A = int(Y/100); B = 2 - A + int(A/4)
    JD = int(365.25*(Y + 4716)) + int(30.6001*(M+1)) + D + B - 1524.5
    return JD

def _sun_declination_and_hourangle(dt_utc, lon, lat):
    JD = _julian_day(dt_utc)
    d = JD - 2451545.0
    g = (357.529 + 0.98560028 * d) % 360
    q = (280.459 + 0.98564736 * d) % 360
    L = (q + 1.915 * math.sin(math.radians(g)) + 0.020 *
math.sin(math.radians(2*g))) % 360
    eps = 23.439 - 3.56e-7 * d
    eps_rad = math.radians(eps); Lrad = math.radians(L)
    decl = math.asin(math.sin(eps_rad) * math.sin(Lrad))
    T = d / 36525.0
    GMST = (280.46061837 + 360.98564736629 * (JD - 2451545.0) + 0.000387933
* T**2 - T**3 / 38710000.0)
    LMST_deg = (GMST + lon) % 360
    H_deg = (LMST_deg - L) % 360
    H = math.radians(H_deg if H_deg <= 180 else H_deg - 360)
    return decl, H

```

```

def _sun_zenith_deg(dt_utc, lon, lat):
    decl, H = _sun_declination_and_hourangle(dt_utc, lon, lat)
    lat_rad = math.radians(lat)
    cos_zen = math.sin(lat_rad)*math.sin(decl) +
math.cos(lat_rad)*math.cos(decl)*math.cos(H)
    cos_zen = max(min(cos_zen, 1.0), -1.0)
    return math.degrees(math.acos(cos_zen))

def _earth_distance_factor(dayofyear):
    return 1.0 + 0.033 * math.cos(2.0 * math.pi * (dayofyear / 365.0))

def _airmass_kasten(zenith_deg):
    if zenith_deg >= 90.0: return 9999.0
    z = zenith_deg
    return 1.0 / (math.cos(math.radians(z)) + 0.50572 * (96.07995 - z) ** -
1.6364)

def optimal_tilt_azimuth(latitude):
    lat_abs = abs(latitude)
    tilt = lat_abs
    azimuth = 180.0 if latitude > 0 else 0.0 if latitude < 0 else 0.0
    return tilt, azimuth

def transpose_ghi_to_poa(dt_utc, ghi, lat, lon, tilt, azimuth, albedo=0.2,
solar_constant=1361.0):
    zen = _sun_zenith_deg(dt_utc, lon, lat)
    cos_z = math.cos(math.radians(zen)) if zen < 90.0 else 0.0
    if cos_z <= 0 or ghi <= 0: return 0.0
    doy = dt_utc.timetuple().tm_yday
    dist_factor = _earth_distance_factor(doy)
    I0 = solar_constant * dist_factor * cos_z
    kt = ghi / I0 if I0 > 0 else 0.0; kt = max(kt, 0.0)
    if kt <= 0.35: fd = 1.0 - 0.249 * kt
    elif kt <= 0.75: fd = 1.557 - 1.84 * kt
    else: fd = 0.177
    fd = max(0.0, min(fd, 1.0))
    dhi = ghi * fd
    dni = (ghi - dhi) / cos_z if cos_z > 0 else 0.0
    decl, H = _sun_declination_and_hourangle(dt_utc, lon, lat)
    lat_rad = math.radians(lat); tilt_rad = math.radians(tilt); az_rad =
math.radians(azimuth)
    cos_theta = (math.sin(decl) * math.sin(lat_rad) * math.cos(tilt_rad)
- math.sin(decl) * math.cos(lat_rad) * math.sin(tilt_rad) *
math.cos(az_rad)
+ math.cos(decl) * math.cos(lat_rad) * math.cos(tilt_rad) *
math.cos(H)
+ math.cos(decl) * math.sin(lat_rad) * math.sin(tilt_rad) *
math.cos(az_rad) * math.cos(H)
+ math.cos(decl) * math.sin(tilt_rad) * math.sin(az_rad) *
math.sin(H))
    cos_theta = max(0.0, cos_theta)

```

```

beam = dni * cos_theta
diffuse = dhi * (1.0 + math.cos(tilt_rad)) / 2.0
ground = ghi * albedo * (1.0 - math.cos(tilt_rad)) / 2.0
poa = beam + diffuse + ground
return max(poa, 0.0)

def calculate_solar_flux_and_save_to_db(file_name, latitude, longitude, photoEfficiency, k_photoEffTemp, photoNOCT, timezone='Asia/Vladivostok'):
    conn = sqlite3.connect(file_name)
    df = pd.read_sql_query("SELECT datetime_utc, clouds, temperature FROM screenshots", conn)
    conn.close()
    df['datetime'] = pd.to_datetime(df['datetime_utc'])
    df.sort_values('datetime', inplace=True); df.reset_index(drop=True, inplace=True)
    if 'temperature' not in df.columns: df['temperature'] = 25.0
    df['temperature'] = pd.to_numeric(df['temperature'], errors='coerce').fillna(25.0)
    if 'clouds' not in df.columns: df['clouds'] = 0.0
    df['clouds'] = pd.to_numeric(df['clouds'], errors='coerce').fillna(0.0)
    df['clouds'] = df['clouds'].clip(0, 100)
    solar_constant = 1361.0; tau = 0.4
    solar_flux_rows = []
    for idx, row in df.iterrows():
        dt = row['datetime']
        if hasattr(dt, 'tzinfo') and dt.tzinfo is not None: dt_utc = dt.tz_convert('UTC').replace(tzinfo=None)
        else: dt_utc = dt
        doy = dt_utc.timetuple().tm_yday
        zen = _sun_zenith_deg(dt_utc, longitude, latitude)
        cos_z = math.cos(math.radians(zen)) if zen < 90.0 else 0.0; cos_z = max(cos_z, 0.0)
        dist_factor = _earth_distance_factor(doy)
        albedo_coeff = 0.5 * (1.0 + 0.6 * math.sin(2.0 * math.pi * (doy - 172 - 91) / 365.0))
        reflected_factor = 1.0 + albedo_coeff
        I_ex = solar_constant * dist_factor * cos_z * reflected_factor; I_ex = max(I_ex, 0.0)
        airmass = _airmass_kasten(zen)
        atm_trans = math.exp(-tau * airmass) if airmass < 999 else 0.0
        I_surface = I_ex * atm_trans
        amb_temp = float(row.get('temperature', 25.0))
        cell_temp = amb_temp + (photoNOCT - 20.0) / 800.0 * I_surface
        temp_factor = 1.0 + k_photoEffTemp * (cell_temp - 25.0)
        solar_flux_value = I_surface * temp_factor
        if not np.isfinite(solar_flux_value): solar_flux_value = 0.0
        solar_flux_value = max(solar_flux_value, 0.0)
        try:
            tz_obj = pytz.timezone(timezone)
            datetime_local = dt_utc.replace(tzinfo=pytz.UTC).astimezone(tz_obj)

```

```

    except:
        datetime_local =
pd.Timestamp(dt_utc).tz_localize('UTC').tz_convert(timezone)
        datetime_str = datetime_local.strftime('%Y-%m-%d %H:%M:%S')
        solar_flux_rows.append((datetime_str, float(solar_flux_value)))
        new_db_file = get_unique_filename(LOCAL_BASE_PATH,
'1.2_ClearSky_Albedo.db')
        conn_new = sqlite3.connect(new_db_file)
        cursor_new = conn_new.cursor()
        cursor_new.execute(''CREATE TABLE IF NOT EXISTS Solar_Flux (datetime
TIMESTAMP, solar_flux FLOAT)'')
        cursor_new.executemany("INSERT INTO Solar_Flux (datetime, solar_flux)
VALUES (?, ?)", solar_flux_rows)
        cursor_new.execute("UPDATE Solar_Flux SET solar_flux = 0 WHERE so-
lar_flux IS NULL")
        conn_new.commit(); conn_new.close()
        return new_db_file

def calculate_cloudiness(solar_flux_file1, solar_flux_file2, time-
zone='Asia/Vladivostok'):
    cloudiness_db_file = get_unique_filename(LOCAL_BASE_PATH,
'3.2_SimSolarFlux.db')
    conn1 = sqlite3.connect(solar_flux_file1); df1 =
pd.read_sql_query("SELECT datetime, solar_flux FROM Solar_Flux", conn1);
conn1.close()
    conn2 = sqlite3.connect(solar_flux_file2); df2 =
pd.read_sql_query("SELECT datetime, clouds FROM screenshots", conn2);
conn2.close()
    df1['datetime'] = pd.to_datetime(df1['datetime']); df1['solar_flux'] =
df1['solar_flux'].astype(float)
    df2['datetime'] = pd.to_datetime(df2['datetime'], format='mixed');
df2['clouds'] = pd.to_numeric(df2['clouds'], errors='coerce').fillna(0)
    df_merged = pd.merge(df1, df2, on='datetime', how='inner')
    if len(df_merged) == 0: raise ValueError("Нет совпадающих записей")
    df_merged['cloudiness'] = df_merged['solar_flux'] * (1 -
(df_merged['clouds'] / 105))
    df_cloudiness = df_merged[['datetime', 'cloudi-
ness']].rename(columns={'cloudiness': 'solar_flux'})
    conn_cloudiness = sqlite3.connect(cloudiness_db_file)
    df_cloudiness.to_sql('Cloudiness', conn_cloudiness, if_exists='replace',
index=False)
    conn_cloudiness.close()
    return cloudiness_db_file

# = ОСНОВНАЯ ФУНКЦИЯ (запускает весь расчёт) =
def run_simulation(params):
    """
params: словарь со всеми введёнными пользователем параметрами.
Возвращает путь к сгенерированному PDF-отчёту.
    """
    # Извлекаем параметры

```

```

lat = params['lat']; lon = params['lon']; local_tz =
params.get('timezone', 'Asia/Vladivostok')
start_str = params['start_date']; end_str = params['end_date']
# Параметры нагрузки
max_load_power_kw = params['max_load_power_kw']
max_load_work_hours = params['max_load_work_hours']
load_min_start_hour = params['load_min_start_hour']; load_min_end_hour =
params['load_min_end_hour']
load_max_start_hour = params['load_max_start_hour']; load_max_end_hour =
params['load_max_end_hour']
load_normal_start_hour = params['load_normal_start_hour'];
load_normal_end_hour = params['load_normal_end_hour']
# Панель
photoEfficiency = params['photoEfficiency']; k_photoEffTemp =
params['k_photoEffTemp']; photoNOCT = params['photoNOCT']
panel_Vmp = params['panel_Vmp']; panel_Voc = params['panel_Voc']; pan-
el_Imp = params['panel_Imp']; panel_Isc = params['panel_Isc']
photoCellWidth = params['photoCellWidth']; photoCellHigh =
params['photoCellHigh']; photoCellNum = params['photoCellNum']
# Аккумулятор
numberbattery_1 = params['numberbattery_1']; charge_voltage =
params['charge_voltage']; battery_nominal_ah = params['battery_nominal_ah'];
battery_voltage = params['battery_voltage']
k_min_SoC = params.get('k_min_SoC', 0.1); k_mode_battery =
params.get('k_mode_battery', 0.8); kCharge = params.get('kCharge', 0.3)
# Инвертор
inverter_nominal_power = params['inverter_nominal_power']; invert-
er_efficiency = params['inverter_efficiency']
inverter_battery_voltage_nominal =
params['inverter_battery_voltage_nominal']; inverter_battery_voltage_min =
params['inverter_battery_voltage_min']; inverter_battery_voltage_max =
params['inverter_battery_voltage_max']
inverter_max_battery_charge_current =
params['inverter_max_battery_charge_current']; invert-
er_max_battery_discharge_current =
params['inverter_max_battery_discharge_current']
inverter_mppt_min = params['inverter_mppt_min']; inverter_mppt_max =
params['inverter_mppt_max']; inverter_pv_max_voltage =
params['inverter_pv_max_voltage']; inverter_pv_start_voltage =
params['inverter_pv_start_voltage']
inverter_mppt_count = params['inverter_mppt_count']; invert-
er_pv_nominal_power = params['inverter_pv_nominal_power']
inverter_pv_max_current_per_mppt =
params['inverter_pv_max_current_per_mppt']; inverter_pv_max_isc_per_mppt =
params['inverter_pv_max_isc_per_mppt']; max_inverters_in_parallel =
params['max_inverters_in_parallel']
# Потери
k_cable_pv = params['k_cable_pv']; k_cable_batt =
params['k_cable_batt']; k_cable_ac = params['k_cable_ac']
reserve_factor_inverter = params.get('reserve_factor_inverter', 1.2)

```

```

    roof_length = params.get('roof_length', 30); roof_width =
params.get('roof_width', 30)

# ----- 1. Получение данных облачности -----
db_clouds = get_unique_filename(LOCAL_BASE_PATH, '1.3_clouds_temp.db')
fetch_and_store_clouds(db_clouds, lat, lon, start_str, end_str, local_tz)

# ----- 2. Расчёт производных величин нагрузки -----
def _interval_hours(start_h, end_h):
    start_h = int(start_h) % 24; end_h = int(end_h) % 24
    if end_h >= start_h: return end_h - start_h
    return (24 - start_h) + end_h
load_min_hours = _interval_hours(load_min_start_hour, load_min_end_hour)
load_max_hours = _interval_hours(load_max_start_hour, load_max_end_hour)
load_normal_hours = _interval_hours(load_normal_start_hour,
load_normal_end_hour)
load_power_max_kw = max_load_power_kw / max_load_work_hours
load_power_max_w = load_power_max_kw * 1000.0
load_power_normal_w = 0.60 * load_power_max_w
load_power_min_w = 0.10 * load_power_max_w
def get_load_power_w(hour):
    hour = int(hour) % 24
    if load_min_start_hour <= hour < load_min_end_hour: return
load_power_min_w
    elif load_max_start_hour <= hour < load_max_end_hour: return
load_power_max_w
    else: return load_power_normal_w

# ----- 3. Раскладка панелей на крыше -----
min_panels_in_series = math.ceil(inverter_mppt_min / panel_Vmp)
max_panels_in_series = math.floor(inverter_mppt_max / panel_Voc)
max_panels_absolute = math.floor(inverter_pv_max_voltage / panel_Voc)
def optimize_pv_on_roof(roof_length_m, roof_width_m, panel_length_m=1.762, panel_width_m=1.134, edge_margin_m=0.30, gap_m=0.02):
    def count_fit(roof_dim, panel_dim):
        usable_dim = roof_dim - 2 * edge_margin_m
        if usable_dim <= 0: return 0
        return max(0, math.floor((usable_dim + gap_m) / (panel_dim + gap_m)))
    candidates = []
    for orientation, dim_along_length, dim_along_width in [("portrait", panel_width_m, panel_length_m), ("landscape", panel_length_m, panel_width_m)]:
        cols = count_fit(roof_length_m, dim_along_length); rows = count_fit(roof_width_m, dim_along_width)
        total_panels = cols * rows
        used_length = cols * dim_along_length + max(0, cols - 1) * gap_m; used_width = rows * dim_along_width + max(0, rows - 1) * gap_m
        free_length = roof_length_m - 2 * edge_margin_m - used_length;
        free_width = roof_width_m - 2 * edge_margin_m - used_width

```

```

        valid_string_lengths = [s for s in range(min_panels_in_series,
min(max_panels_in_series, cols) + 1) if cols % s == 0]
        if valid_string_lengths:
            panels_per_string = max(valid_string_lengths);
strings_per_row = cols // panels_per_string; total_strings = rows *
strings_per_row; electrical_ok = True
        else: panels_per_string = None; strings_per_row = None; to-
tal_strings = None; electrical_ok = False
            candidates.append({"orientation": orientation,
"cols_along_length": cols, "rows_along_width": rows, "total_panels": to-
tal_panels,
                                "panels_per_string": panels_per_string,
"strings_per_row": strings_per_row, "total_strings": total_strings,
                                "electrical_ok": electrical_ok,
"used_length_m": round(used_length,3), "used_width_m": round(used_width,3),
                                "free_length_m": round(free_length,3),
"free_width_m": round(free_width,3)})
            best = max(candidates, key=lambda x: (x["total_panels"],
int(x["electrical_ok"]), -(x["total_strings"] if x["total_strings"] is not
None else 10**9)))
            return best, candidates
        best_layout, _ = optimize_pv_on_roof(roof_length_m=roof_length,
roof_width_m=roof_width)
        installed_panels = best_layout["total_panels"]
        photoCellSquare = photoCellWidth * photoCellHigh * photoCellNum * in-
stalled_panels

# ----- 4. Балансировка оборудования -----
        required_inverter_power = (max_load_power_kw) * reserve_factor_inverter
        required_inverter_power_w = required_inverter_power * 1000
        n_inv_by_load = math.ceil(required_inverter_power / invert-
er_nominal_power)
        if best_layout["electrical_ok"] and best_layout["total_strings"] is not
None:
            total_strings = best_layout["total_strings"]
            max_strings_per_mppt = math.floor(inverter_pv_max_current_per_mppt /
panel_Imp) if panel_Imp>0 else 1
            if max_strings_per_mppt < 1: max_strings_per_mppt = 1
            mppt_per_inv = inverter_mppt_count
            max_strings_per_inv = mppt_per_inv * max_strings_per_mppt
            n_inv_by_strings = math.ceil(total_strings / max_strings_per_inv)
            total_pv_power_kw = installed_panels * (panel_Vmp * panel_Imp) /
1000.0
            pv_power_per_inv_kw = inverter_pv_nominal_power / 1000.0
            n_inv_by_pv_power = math.ceil(total_pv_power_kw /
pv_power_per_inv_kw)
            n_inv_by_pv = max(n_inv_by_strings, n_inv_by_pv_power)
        else:
            n_inv_by_pv = 1; total_strings = 0; max_strings_per_mppt = 1
            n_inverters_required = max(n_inv_by_load, n_inv_by_pv)
            if n_inverters_required > max_inverters_in_parallel:

```

```

n_inverters_required = max_inverters_in_parallel

# Адаптация конфигурации панелей
if best_layout["electrical_ok"] and total_strings:
    pv_panels_min = math.ceil(max(inverter_mppt_min, invert-
er_pv_start_voltage) / panel_Vmp)
    pv_panels_max = math.floor(min(inverter_mppt_max, invert-
er_pv_max_voltage) / panel_Voc)
    adapted = False
    possible_lengths = [l for l in range(pv_panels_min,
min(pv_panels_max, installed_panels)+1) if installed_panels % l == 0]
    possible_lengths.sort(reverse=True)
    for length in possible_lengths:
        strings = installed_panels // length
        strings_per_inv_test = math.ceil(strings / n_inverters_required)
        strings_per_mppt_test = math.ceil(strings_per_inv_test / invert-
er_mppt_count)
        if strings_per_mppt_test <= max_strings_per_mppt:
            panels_per_string = length; total_strings = strings; adapted
= True; break
        if not adapted:
            for n_inv_try in range(n_inverters_required+1,
max_inverters_in_parallel+1):
                for length in possible_lengths:
                    strings = installed_panels // length
                    strings_per_inv_test = math.ceil(strings / n_inv_try)
                    strings_per_mppt_test = math.ceil(strings_per_inv_test /
inverter_mppt_count)
                    if strings_per_mppt_test <= max_strings_per_mppt:
                        n_inverters_required = n_inv_try; panels_per_string
= length; total_strings = strings; adapted = True; break
                    if adapted: break
            if not adapted:
                for remove_strings in range(1, total_strings):
                    new_panels = installed_panels - panels_per_string * re-
move_strings
                    if new_panels <= 0: break
                    possible_lengths_red = [l for l in range(pv_panels_min,
min(pv_panels_max, new_panels)+1) if new_panels % l == 0]
                    possible_lengths_red.sort(reverse=True)
                    for length in possible_lengths_red:
                        strings = new_panels // length
                        strings_per_inv_test = math.ceil(strings /
n_inverters_required)
                        strings_per_mppt_test = math.ceil(strings_per_inv_test /
inverter_mppt_count)
                        if strings_per_mppt_test <= max_strings_per_mppt:
                            installed_panels = new_panels; panels_per_string =
length; total_strings = strings; adapted = True; break
                        if adapted: break
            if adapted:

```

```

        best_layout["panels_per_string"] = panels_per_string
        best_layout["total_strings"] = total_strings
        best_layout["total_panels"] = installed_panels
        total_pv_power_kw = installed_panels * (panel_Vmp * panel_Imp) /
1000.0

# Расчёт высоковольтной батареи
n_inv = n_inverters_required
battery_series_count_min = math.ceil(inverter_battery_voltage_min / bat-
tery_voltage)
battery_series_count_max = math.floor(inverter_battery_voltage_max /
battery_voltage)
power_per_inv_w = required_inverter_power_w / n_inv
battery_series_count_min_by_current = math.ceil(power_per_inv_w / (in-
verter_max_battery_discharge_current * battery_voltage * invert-
er_efficiency))
battery_series_count = max(battery_series_count_min, bat-
tery_series_count_min_by_current)
if battery_series_count > battery_series_count_max: battery_series_count
= battery_series_count_max
battery_bank_voltage = battery_series_count * battery_voltage
battery_current_per_inv = power_per_inv_w / (battery_bank_voltage * in-
verter_efficiency)
battery_string_current_limit = battery_nominal_ah * kCharge
parallel_by_current = math.ceil(battery_current_per_inv / bat-
tery_string_current_limit)
daily_energy_wh = (load_power_min_w * load_min_hours +
load_power_normal_w * load_normal_hours + load_power_max_w * load_max_hours)
daily_energy_per_inv = daily_energy_wh / n_inv
usable_per_string_wh = (battery_series_count * battery_nominal_ah * bat-
tery_voltage * (1 - k_min_SoC) * k_mode_battery)
parallel_by_energy = math.ceil(daily_energy_per_inv / usa-
ble_per_string_wh)
battery_parallel_count_per_inv = max(parallel_by_current, paral-
lel_by_energy, 1)
numberbattery_per_inv = battery_series_count * bat-
tery_parallel_count_per_inv
numberbattery_total = n_inv * numberbattery_per_inv
battery_parallel_count = battery_parallel_count_per_inv * n_inv
numberbattery = numberbattery_total
battery_voltage_final = battery_bank_voltage
nominal_capacity_battery = battery_nominal_ah * battery_parallel_count
batteryCapacityLim = nominal_capacity_battery * (1 - k_min_SoC) *
k_mode_battery
batteryCapacity = nominal_capacity_battery * k_mode_battery
maxChargeCurrent = nominal_capacity_battery * kCharge
battery_DoD = nominal_capacity_battery * k_min_SoC * k_mode_battery
battery_energy_max_wh = batteryCapacity * battery_bank_voltage
battery_energy_control_min_wh = battery_DoD * battery_bank_voltage
battery_floor_ah = 0.001; battery_energy_floor_wh = battery_floor_ah *
battery_bank_voltage

```

```

battery_charge_limit_w = maxChargeCurrent * charge_voltage
battery_discharge_limit_w = n_inv * invert-
er_max_battery_discharge_current * battery_bank_voltage

# ----- 5. Солнечная энергия и облачность -----
solar_flux_file2 = db_clouds
solar_flux_file1 = calculate_solar_flux_and_save_to_db(solar_flux_file2,
lat, lon, photoEfficiency, k_photoEffTemp, photoNOCT, local_tz)
cloudiness_db_file = calculate_cloudiness(solar_flux_file1, so-
lar_flux_file2, local_tz)

# ----- 6. Баланс и дефицит -----
start_date = pd.to_datetime(start_str); end_date =
pd.to_datetime(end_str)
conn = sqlite3.connect(cloudiness_db_file)
hes_data = pd.read_sql_query("SELECT * FROM Cloudiness", conn)
conn.close()
hes_data['datetime'] = pd.to_datetime(hes_data['datetime'])
hes_data['solar_flux'] = pd.to_numeric(hes_data['solar_flux'], er-
rors='coerce')
hes_data = hes_data.dropna(subset=['datetime', 'solar_flux']).copy()
filtered_hes_data = hes_data[(hes_data['datetime'] >= start_date) &
(hes_data['datetime'] <= end_date)].copy().reset_index(drop=True)
filtered_hes_data['datetime'] = fil-
tered_hes_data['datetime'].dt.round('60min')
filtered_hes_data['solar_flux'] =
pd.to_numeric(filtered_hes_data['solar_flux'], errors='coerce').fillna(0.0)
filtered_hes_data['load_power_current_w'] = fil-
tered_hes_data['datetime'].dt.hour.apply(get_load_power_w)

full_range = pd.date_range(start=filtered_hes_data['datetime'].min(),
end=filtered_hes_data['datetime'].max(), freq='60min')
full_range_date_df = pd.DataFrame({'datetime': full_range})
full_range_filtered_hes_data = pd.merge(full_range_date_df, fil-
tered_hes_data, on='datetime', how='left')
full_range_filtered_hes_data =
full_range_filtered_hes_data.drop_duplicates(subset='datetime').reset_index(
drop=True)
full_range_filtered_hes_data['hour_diff'] =
(full_range_filtered_hes_data['datetime'] -
full_range_filtered_hes_data['datetime'].shift()).dt.total_seconds() / 3600
if len(full_range_filtered_hes_data) > 1:
full_range_filtered_hes_data.loc[0, 'hour_diff'] =
full_range_filtered_hes_data.loc[1, 'hour_diff']
else: full_range_filtered_hes_data.loc[0, 'hour_diff'] = 1.0
full_range_filtered_hes_data['solar_flux'] =
pd.to_numeric(full_range_filtered_hes_data['solar_flux'], er-
rors='coerce').interpolate().ffill().bfill().fillna(0.0)
full_range_filtered_hes_data['load_power_current_w'] =
full_range_filtered_hes_data['datetime'].dt.hour.apply(get_load_power_w)

```

```

full_range_filtered_hes_data['load_energy_step_wh'] =
full_range_filtered_hes_data['load_power_current_w'] *
full_range_filtered_hes_data['hour_diff']
full_range_filtered_hes_data['load_energy_wh'] =
full_range_filtered_hes_data['load_energy_step_wh'].cumsum()

numberpanel = installed_panels
panel_active_area_m2 = photoCellWidth * photoCellHigh * photoCellNum
pv_array_active_area_m2 = panel_active_area_m2 * numberpanel
full_range_filtered_hes_data['solar_power_dc_w'] =
full_range_filtered_hes_data['solar_flux'] * photoEfficiency *
pv_array_active_area_m2
full_range_filtered_hes_data['solar_power_after_pv_cable_w'] =
full_range_filtered_hes_data['solar_power_dc_w'] * k_cable_pv
full_range_filtered_hes_data['pv_ac_available_w'] =
full_range_filtered_hes_data['solar_power_after_pv_cable_w'] * invert-
er_efficiency * k_cable_ac

# Моделирование работы АКБ
start_soc = params.get('initial_battery_soc', 0.6)
full_range_filtered_hes_data['battery_energy'] = np.nan
full_range_filtered_hes_data.loc[0, 'battery_energy'] = bat-
tery_energy_floor_wh + (battery_energy_max_wh - battery_energy_floor_wh) *
start_soc
for col in
['pv_to_load_w', 'pv_used_for_load_dc_w', 'pv_surplus_after_load_w', 'battery_c
harge_available_w', 'battery_charge_power_w',
'bat-
tery_discharge_power_w', 'battery_power', 'battery_energy_step_wh', 'battery_ac
_supply_w', 'served_power_w',
'grid_import_w', 'grid_energy_step_wh', 'deficit_after_pv_w', '
deficit_after_battery_w', 'power_balance_w']:
full_range_filtered_hes_data[col] = 0.0
grid_charge_enabled = False
for i in range(len(full_range_filtered_hes_data)):
if i == 0: prev_energy = float(full_range_filtered_hes_data.at[i,
'battery_energy'])
else: prev_energy = float(full_range_filtered_hes_data.at[i-1, 'bat-
tery_energy'])
dt = float(full_range_filtered_hes_data.at[i, 'hour_diff']); dt = dt
if pd.notna(dt) and dt>0 else 1.0
solar_after_pv_cable_w = float(full_range_filtered_hes_data.at[i,
'solar_power_after_pv_cable_w'])
pv_ac_available_w = float(full_range_filtered_hes_data.at[i,
'pv_ac_available_w'])
load_current_w = float(full_range_filtered_hes_data.at[i,
'load_power_current_w'])
pv_to_load_w = min(load_current_w, pv_ac_available_w)
if inverter_efficiency * k_cable_ac > 0: pv_used_for_load_dc_w =
pv_to_load_w / (inverter_efficiency * k_cable_ac)
else: pv_used_for_load_dc_w = 0.0

```

```

deficit_after_pv_w = max(load_current_w - pv_to_load_w, 0.0)
pv_surplus_after_load_w = max(solar_after_pv_cable_w -
pv_used_for_load_dc_w, 0.0)
battery_charge_available_w = pv_surplus_after_load_w * k_cable_batt
max_charge_possible_w = max((battery_energy_max_wh - prev_energy) /
dt, 0.0)
max_discharge_possible_w = max((prev_energy - bat-
tery_energy_floor_wh) / dt, 0.0)
charge_from_pv = min(battery_charge_available_w, bat-
tery_charge_limit_w, max_charge_possible_w)
if deficit_after_pv_w > 0:
    if inverter_efficiency * k_cable_batt * k_cable_ac > 0: re-
quired_discharge_w = deficit_after_pv_w / (inverter_efficiency *
k_cable_batt * k_cable_ac)
    else: required_discharge_w = 0.0
    discharge_power_w = min(required_discharge_w, bat-
tery_discharge_limit_w, max_discharge_possible_w)
    battery_ac_supply_w = discharge_power_w * inverter_efficiency *
k_cable_batt * k_cable_ac
    else: discharge_power_w = 0.0; battery_ac_supply_w = 0.0
    energy_after_discharge = prev_energy - discharge_power_w * dt
    if energy_after_discharge < battery_energy_floor_wh: ener-
gy_after_discharge = battery_energy_floor_wh
    if charge_from_pv > 0: grid_charge_enabled = False
    else:
        if energy_after_discharge <= battery_energy_control_min_wh:
grid_charge_enabled = True
        if charge_from_pv > 0:
            charge_power_w = charge_from_pv; grid_charge_power = 0.0
        else:
            charge_power_w = 0.0
            if grid_charge_enabled:
                max_grid_charge = max((battery_energy_max_wh - ener-
gy_after_discharge) / dt, 0.0)
                grid_charge_power = min(battery_charge_limit_w,
max_grid_charge)
            else: grid_charge_power = 0.0
        next_energy = energy_after_discharge + (charge_power_w +
grid_charge_power) * dt
        if next_energy > battery_energy_max_wh: next_energy = bat-
tery_energy_max_wh
        if next_energy < battery_energy_floor_wh: next_energy = bat-
tery_energy_floor_wh
        served_power_w = pv_to_load_w + battery_ac_supply_w
        deficit_after_battery_w = max(load_current_w - served_power_w, 0.0)
        if k_cable_batt > 0: power_balance_w = solar_after_pv_cable_w -
pv_used_for_load_dc_w - (charge_power_w / k_cable_batt)
        else: power_balance_w = solar_after_pv_cable_w -
pv_used_for_load_dc_w
        full_range_filtered_hes_data.at[i, 'pv_to_load_w'] = pv_to_load_w

```

```

        full_range_filtered_hes_data.at[i, 'pv_used_for_load_dc_w'] =
pv_used_for_load_dc_w
        full_range_filtered_hes_data.at[i, 'pv_surplus_after_load_w'] =
pv_surplus_after_load_w
        full_range_filtered_hes_data.at[i, 'battery_charge_available_w'] =
battery_charge_available_w
        full_range_filtered_hes_data.at[i, 'battery_charge_power_w'] =
charge_power_w
        full_range_filtered_hes_data.at[i, 'battery_discharge_power_w'] =
discharge_power_w
        full_range_filtered_hes_data.at[i, 'battery_energy'] = next_energy
        full_range_filtered_hes_data.at[i, 'battery_ac_supply_w'] = bat-
tery_ac_supply_w
        full_range_filtered_hes_data.at[i, 'served_power_w'] =
served_power_w
        full_range_filtered_hes_data.at[i, 'grid_import_w'] =
grid_charge_power
        full_range_filtered_hes_data.at[i, 'grid_energy_step_wh'] =
grid_charge_power * dt
        full_range_filtered_hes_data.at[i, 'deficit_after_pv_w'] = defi-
cit_after_pv_w
        full_range_filtered_hes_data.at[i, 'deficit_after_battery_w'] = def-
icit_after_battery_w
        full_range_filtered_hes_data['balance_no_battery_w'] =
full_range_filtered_hes_data['pv_ac_available_w'] -
full_range_filtered_hes_data['load_power_current_w']
        full_range_filtered_hes_data['balance_with_battery_w'] =
full_range_filtered_hes_data['pv_ac_available_w'] +
full_range_filtered_hes_data['battery_ac_supply_w'] -
full_range_filtered_hes_data['load_power_current_w']
        # Добавить расчёт накопленной солнечной энергии
        full_range_filtered_hes_data['solar_energy_step_wh'] =
full_range_filtered_hes_data['solar_power_after_pv_cable_w'] *
full_range_filtered_hes_data['hour_diff']
        full_range_filtered_hes_data['solar_energy'] =
full_range_filtered_hes_data['solar_energy_step_wh'].cumsum()
        # Добавить колонку с минимальным порогом энергии АКБ (для графика)
        full_range_filtered_hes_data['battery_min_energy_wh'] = bat-
tery_energy_control_min_wh

# = ДОПОЛНИТЕЛЬНЫЕ РАСЧЁТЫ ДЛЯ ПОЛНОГО ОТЧЁТА =
# --- Проверка совместимости оборудования ---
total_inv_nominal_power = inverter_nominal_power *
n_inverters_required
inverter_power_ok = total_inv_nominal_power >= re-
quired_inverter_power
battery_voltage_ok = inverter_battery_voltage_min <= bat-
tery_bank_voltage <= inverter_battery_voltage_max
inverter_battery_current_ok = battery_current_per_inv <= invert-
er_max_battery_discharge_current

```

```

        total_battery_discharge_limit_a = battery_parallel_count * bat-
        tery_nominal_ah * kCharge
        total_required_battery_current_a = battery_current_per_inv *
        n_inverters_required
        battery_current_ok = total_required_battery_current_a <= to-
        tal_battery_discharge_limit_a
        usable_battery_energy_wh = battery_energy_max_wh - bat-
        tery_energy_control_min_wh
        battery_energy_ok = usable_battery_energy_wh >= daily_energy_wh *
1.0 # autonomy days = 1
        pv_panels_min_per_string = math.ceil(max(inverter_mppt_min, invert-
        er_pv_start_voltage) / panel_Vmp)
        pv_panels_max_per_string = math.floor(min(inverter_mppt_max, invert-
        er_pv_max_voltage) / panel_Voc)
        pv_string_voltage_ok = pv_panels_min_per_string <= panels_per_string
<= pv_panels_max_per_string
        if best_layout["electrical_ok"] and total_strings:
            strings_per_mppt_actual = math.ceil(total_strings /
(n_inverters_required * inverter_mppt_count))
            pv_current_per_mppt = strings_per_mppt_actual * panel_Isc
            pv_current_ok = (pv_current_per_mppt <= invert-
        er_pv_max_current_per_mppt and
                            pv_current_per_mppt <= invert-
        er_pv_max_isc_per_mppt)
        else:
            pv_current_ok = False
            equipment_ok = (inverter_power_ok and battery_voltage_ok and invert-
        er_battery_current_ok and
                            battery_current_ok and battery_energy_ok and
        pv_string_voltage_ok and pv_current_ok)

# --- Ориентация панелей (оптимальный угол) ---
tilt_opt, az_opt = optimal_tilt_azimuth(lat)

# --- Периоды дефицита (без учёта АКБ) ---
mask_deficit = full_range_filtered_hes_data['balance_no_battery_w']
< 0

if mask_deficit.any():
    group_ids = (mask_deficit != mask_deficit.shift()).cumsum()
    deficit_periods =
full_range_filtered_hes_data[mask_deficit].groupby(group_ids[mask_deficit])
    periods_list = []
    total_deficit_energy_wh = 0.0
    for _, period in deficit_periods:
        start_time = period['datetime'].iloc[0]
        end_time = period['datetime'].iloc[-1]
        energy_wh = (-period['balance_no_battery_w'] * peri-
        od['hour_diff']).sum()
        periods_list.append({'Начало дефицита': start_time, 'Конец
дефицита': end_time, 'Энергия дефицита, Вт·ч': round(energy_wh, 2)})
        total_deficit_energy_wh += energy_wh

```

```

deficit_df = pd.DataFrame(periods_list)
else:
    deficit_df = pd.DataFrame()
    total_deficit_energy_wh = 0.0

# --- Периоды низкого заряда АКБ ---
def find_periods_at_or_below_threshold(df, datetime_col, energy_col,
limit, eps=1e-9):
    tmp = df[[datetime_col, energy_col]].copy()
    tmp[datetime_col] = pd.to_datetime(tmp[datetime_col], er-
rors='coerce')
    tmp[energy_col] = pd.to_numeric(tmp[energy_col], er-
rors='coerce')
    tmp = tmp.dropna(subset=[datetime_col, ener-
gy_col]).sort_values(datetime_col).reset_index(drop=True)
    mask = tmp[energy_col] <= (limit + eps)
    if not mask.any():
        return []
    group_id = (mask != mask.shift()).cumsum()
    periods = []
    for _, g in tmp[mask].groupby(group_id[mask]):
        periods.append((g[datetime_col].iloc[0],
g[datetime_col].iloc[-1]))
    return periods

battery_control_min_periods = find_periods_at_or_below_threshold(
    full_range_filtered_hes_data, 'datetime', 'battery_energy', bat-
tery_energy_control_min_wh
)

# --- Подключения к электросети для зарядки АКБ ---
mask_grid = full_range_filtered_hes_data['grid_import_w'] > 0
if mask_grid.any():
    group_ids_grid = (mask_grid != mask_grid.shift()).cumsum()
    periods_grid =
full_range_filtered_hes_data[mask_grid].groupby(group_ids_grid[mask_grid])
connections = []
    for _, group in periods_grid:
        start_time = group['datetime'].iloc[0]
        end_time = group['datetime'].iloc[-1]
        connections.append((start_time, end_time))
    connect_df = pd.DataFrame(connections, col-
umns=['datetime_start', 'datetime_end'])
    connect_df['connection_number'] = range(1, len(connect_df) + 1)
    connect_df = connect_df[['connection_number', 'datetime_start',
'datetime_end']]
else:
    connect_df = pd.DataFrame()

# --- Дополнительные метрики для раздела "Дефицит и баланс" ---

```

```

    peak_pv_deficit_w =
full_range_filtered_hes_data['deficit_after_pv_w'].max()
    peak_battery_deficit_w =
full_range_filtered_hes_data['deficit_after_battery_w'].max()
    total_grid_energy_wh =
full_range_filtered_hes_data['grid_energy_step_wh'].sum()
    total_battery_charge_wh =
full_range_filtered_hes_data['battery_charge_power_w'].sum()
    total_battery_discharge_wh =
full_range_filtered_hes_data['battery_discharge_power_w'].sum()

    # Количество панелей, активная площадь и т.д.
    numberpanel = installed_panels
    panel_active_area_m2 = photoCellWidth * photoCellHigh * photoCellNum
    pv_array_active_area_m2 = panel_active_area_m2 * numberpanel
    battery_series_count = battery_series_count # уже есть
    battery_parallel_count = battery_parallel_count
    numberbattery = numberbattery_total

# ----- 7. Формирование PDF-отчёта -----
from fpdf import FPDF
from tabulate import tabulate
img_dir = '/tmp/report_images'
os.makedirs(img_dir, exist_ok=True)

# Функции сохранения графиков
def save_energy_graph(data, filename):
    fig, ax = plt.subplots(figsize=(10,4))
    ax.plot(data['datetime'], data['solar_energy'], la-
bel='Выработанная', color='blue')
    ax.plot(data['datetime'], data['load_energy_wh'], la-
bel='Необходимая', color='orange')
    ax.set_xlabel('Дата'); ax.set_ylabel('Вт·ч')
    ax.set_title('Выработанная и необходимая энергия')
    ax.grid(True); ax.legend()
    fig.tight_layout(); fig.savefig(filename); plt.close(fig)
def save_battery_balance_graph(data, min_energy, filename):
    fig, ax = plt.subplots(figsize=(10,4))
    ax.plot(data['datetime'], data['battery_energy'], color='blue')
    ax.axhline(y=min_energy, color='black', linestyle='--', label='Мин.
порог')
    ax.set_xlabel('Дата'); ax.set_ylabel('Вт·ч')
    ax.set_title('Баланс энергии в АКБ')
    ax.grid(True); ax.legend()
    fig.tight_layout(); fig.savefig(filename); plt.close(fig)
def save_charge_discharge_graph(data, filename):
    fig, ax = plt.subplots(figsize=(10,4))
    ax.plot(data['datetime'], data['battery_charge_power_w'],
label='Заряд от PV')
    ax.plot(data['datetime'], data['battery_discharge_power_w'],
label='Разряд АКБ')

```

```

ax.set_xlabel('Дата'); ax.set_ylabel('Вт')
ax.set_title('Заряд и разряд АКБ')
ax.grid(True); ax.legend()
fig.tight_layout(); fig.savefig(filename); plt.close(fig)
def save_grid_import_graph(data, filename):
fig, ax = plt.subplots(figsize=(10,4))
ax.plot(data['datetime'], data['grid_import_w'], color='purple')
ax.set_xlabel('Дата'); ax.set_ylabel('Вт')
ax.set_title('Потребление из сети')
ax.grid(True)
fig.tight_layout(); fig.savefig(filename); plt.close(fig)
def save_deficit_graph(data, filename):
fig, ax = plt.subplots(figsize=(10,4))
ax.plot(data['datetime'], data['balance_no_battery_w'], color='red',
linewidth=1, label='Без АКБ')
ax.fill_between(data['datetime'], 0, data['balance_no_battery_w'],
where=(data['balance_no_battery_w'] < 0), color='red', alpha=0.3,
label='Дефицит без АКБ')
ax.plot(data['datetime'], data['balance_with_battery_w'], col-
or='green', linestyle='--', label='С АКБ')
ax.axhline(y=0, color='black', linewidth=0.5)
ax.set_xlabel('Дата'); ax.set_ylabel('Вт')
ax.set_title('Дефицит генерации PV и покрытие АКБ')
ax.grid(True); ax.legend()
fig.tight_layout(); fig.savefig(filename); plt.close(fig)

# Графики из облачности
conn1 = sqlite3.connect(solar_flux_file1); df1 =
pd.read_sql_query("SELECT * FROM Solar_Flux WHERE datetime BETWEEN ? AND ?",
conn1, params=(start_str, end_str)); conn1.close()
df1['datetime'] = pd.to_datetime(df1['datetime'])
conn2 = sqlite3.connect(solar_flux_file2); df2 =
pd.read_sql_query("SELECT * FROM screenshots WHERE datetime BETWEEN ? AND
?", conn2, params=(start_str, end_str)); conn2.close()
df2['datetime'] = pd.to_datetime(df2['datetime'])
conn3 = sqlite3.connect(cloudiness_db_file); df3 =
pd.read_sql_query("SELECT * FROM Cloudiness WHERE datetime BETWEEN ? AND ?",
conn3, params=(start_str, end_str)); conn3.close()
df3['datetime'] = pd.to_datetime(df3['datetime'])

# График 1: Солнечная энергия
def save_plot1():
# Приводим к datetime и фильтруем по периоду
df1_filtered = df1.copy()
df1_filtered['datetime'] = pd.to_datetime(df1_filtered['datetime'])
start_dt = pd.to_datetime(start_str)
end_dt = pd.to_datetime(end_str)
df1_filtered = df1_filtered[(df1_filtered['datetime'] >= start_dt) &
(df1_filtered['datetime'] <= end_dt)]
# Удаляем дубликаты по времени и сортируем

```

```

df1_filtered =
df1_filtered.drop_duplicates(subset='datetime').sort_values('datetime').reset_index(drop=True)

fig, ax = plt.subplots(figsize=(10,4))
# Используем plot с маркерами
ax.plot(df1_filtered['datetime'], df1_filtered['solar_flux'], linestyle='-', linewidth=1)
ax.set_xlabel('Дата'); ax.set_ylabel('Solar flux (W/m²)')
ax.set_title('Интенсивность солнечного потока (Clear Sky)')
ax.grid(True)
fig.tight_layout()
fig.savefig(os.path.join(img_dir, 'solar_flux_clear.png'))
plt.close(fig)

# График 2: Облачность
def save_plot2():
df2f = df2.copy()
df2f['clouds'] = pd.to_numeric(df2f['clouds'], errors='coerce')
df2f = df2f.dropna(subset=['clouds'])
fig, ax = plt.subplots(figsize=(10, 4))
ax.plot(df2f['datetime'], df2f['clouds'], linestyle='-')
ax.set_xlabel('Дата'); ax.set_ylabel('Облачность (%)')
ax.set_title('Уровень облачности')
ax.grid(True)
fig.tight_layout(); fig.savefig(os.path.join(img_dir, 'cloudiness.png')); plt.close(fig)

# График 3: Температура
def save_plot3():
df_temp = df2[['datetime', 'temperature']].copy()
df_temp['temperature'] = pd.to_numeric(df_temp['temperature'], errors='coerce')
df_temp = df_temp.dropna()
fig, ax = plt.subplots(figsize=(10, 4))
ax.plot(df_temp['datetime'], df_temp['temperature'], linestyle='-')
ax.set_xlabel('Дата'); ax.set_ylabel('Температура (°C)')
ax.set_title('Температура')
ax.grid(True)
fig.tight_layout(); fig.savefig(os.path.join(img_dir, 'temperature.png')); plt.close(fig)

# График 4: Солнечная энергия с учётом облачности
def save_plot4():
df3_filtered = df3.copy()
df3_filtered['datetime'] = pd.to_datetime(df3_filtered['datetime'])
start_dt = pd.to_datetime(start_str)
end_dt = pd.to_datetime(end_str)
df3_filtered = df3_filtered[(df3_filtered['datetime'] >= start_dt) &
(df3_filtered['datetime'] <= end_dt)]

```

```

df3_filtered =
df3_filtered.drop_duplicates(subset='datetime').sort_values('datetime').reset_index(drop=True)

fig, ax = plt.subplots(figsize=(10,4))
ax.plot(df3_filtered['datetime'], df3_filtered['solar_flux'], linestyle='-', linewidth=1)
ax.set_xlabel('Дата'); ax.set_ylabel('Solar flux (W/m²)')
ax.set_title('Интенсивность солнечного потока с учётом облачности')
ax.grid(True)
fig.tight_layout()
fig.savefig(os.path.join(img_dir, 'solar_flux_cloudy.png'))
plt.close(fig)

save_plot1(); save_plot2(); save_plot3(); save_plot4()
save_energy_graph(full_range_filtered_hes_data, os.path.join(img_dir, 'energy.png'))
save_battery_balance_graph(full_range_filtered_hes_data, battery_energy_control_min_wh, os.path.join(img_dir, 'battery_balance.png'))
save_charge_discharge_graph(full_range_filtered_hes_data, os.path.join(img_dir, 'charge_discharge.png'))
save_grid_import_graph(full_range_filtered_hes_data, os.path.join(img_dir, 'grid_import.png'))
save_deficit_graph(full_range_filtered_hes_data, os.path.join(img_dir, 'deficit.png'))

# Создание PDF
pdf = FPDF()
pdf.add_page()
font_path = os.path.join(os.path.dirname(__file__), 'DejaVuSans.ttf')
pdf.add_font('DejaVu', '', font_path, uni=True)
pdf.set_font('DejaVu', '', 12)
pdf.cell(0, 10, 'Отчёт о моделировании солнечной электростанции', ln=True, align='C')
pdf.ln(5)
pdf.set_font('DejaVu', '', 10)
pdf.cell(0, 8, f'Период: {start_str} - {end_str} | Широта {lat}°, долгота {lon}°', ln=True)

# --- Параметры нагрузки ---
pdf.ln(3)
pdf.set_font('DejaVu', '', 12)
pdf.cell(0, 8, 'Параметры нагрузки', ln=True)
pdf.set_font('DejaVu', '', 10)
pdf.cell(0, 6, f'Максимальная нагрузка: {max_load_power_kw:.2f} кВт', ln=True)
pdf.cell(0, 6, f'Рабочие часы: {max_load_work_hours:.2f} ч', ln=True)
pdf.cell(0, 6, f'Нагрузка за 1 час: макс {load_power_max_w:.2f} Вт, средняя {load_power_normal_w:.2f} Вт, мин {load_power_min_w:.2f} Вт', ln=True)

```

```

pdf.cell(0, 6, f'Суточная потребность: итого {daily_energy_wh:.2f}
Вт·ч', ln=True)

# --- Раскладка панелей ---
pdf.ln(3)
pdf.set_font('DejaVu', '', 12)
pdf.cell(0, 8, 'Раскладка панелей на крыше', ln=True)
pdf.set_font('DejaVu', '', 10)
pdf.cell(0, 6, f'Эффективная площадь панелей: {round(photoCellSquare,
2)} м²', ln=True)
if isinstance(best_layout, dict):
    pdf.cell(0, 6, f'Ориентация: {best_layout.get("orientation",
"N/A")}', ln=True)
    pdf.cell(0, 6, f'Панелей в линии:
{best_layout.get("cols_along_length", "N/A")}', ln=True)
    pdf.cell(0, 6, f'Линий по ширине:
{best_layout.get("rows_along_width", "N/A")}', ln=True)
    pdf.cell(0, 6, f'Панелей в строке:
{best_layout.get("panels_per_string", "N/A")}', ln=True)
    pdf.cell(0, 6, f'Строк на линию: {best_layout.get("strings_per_row",
"N/A")}', ln=True)
    pdf.cell(0, 6, f'Всего строк: {best_layout.get("total_strings",
"N/A")}', ln=True)
    pdf.cell(0, 6, f'Всего панелей: {best_layout.get("total_panels",
"N/A")}', ln=True)

# --- Сбалансированное оборудование ---
pdf.ln(3)
pdf.set_font('DejaVu', '', 12)
pdf.cell(0, 8, 'Сбалансированное оборудование', ln=True)
pdf.set_font('DejaVu', '', 10)
pdf.cell(0, 6, f'Инверторов: {n_inverters_required}', ln=True)
pdf.cell(0, 6, f'Панелей всего: {installed_panels}, в строке: {pan-
els_per_string}, строк: {total_strings}', ln=True)
mppt_per_inv = inverter_mppt_count
strings_per_mppt_calc = math.ceil(total_strings / (n_inverters_required
* mppt_per_inv)) if total_strings > 0 else 0
pdf.cell(0, 6, f'Строк на МРРТ: {strings_per_mppt_calc}', ln=True)
pdf.cell(0, 6, f'Напряжение батарейного банка: {bat-
tery_bank_voltage:.2f} В', ln=True)
pdf.cell(0, 6, f'Параллельных веток на инвертор: {bat-
tery_parallel_count_per_inv}, всего в системе: {battery_parallel_count}',
ln=True)
pdf.cell(0, 6, f'Всего аккумуляторов: {numberbattery_total}', ln=True)
pdf.cell(0, 6, f'Номинальная ёмкость: {nominal_capacity_battery:.2f}
А·ч', ln=True)
pdf.cell(0, 6, f'Доступная ёмкость: {batteryCapacityLim:.2f} А·ч',
ln=True)
pdf.cell(0, 6, f'Фактическая ёмкость: {batteryCapacity:.2f} А·ч',
ln=True)

```

```

pdf.cell(0, 6, f'Минимально допустимая ёмкость: {battery_DoD:.2f} А·ч',
ln=True)
pdf.cell(0, 6, f'Энергия банка: {battery_energy_floor_wh:.2f}..{battery_energy_max_wh:.2f} Вт·ч', ln=True)

# --- Проверка совместимости ---
pdf.ln(3)
pdf.set_font('DejaVu', '', 12)
pdf.cell(0, 8, 'Проверка совместимости', ln=True)
pdf.set_font('DejaVu', '', 10)
if 'equipment_ok' in globals():
    pdf.cell(0, 6, f'ВЫВОД: {"связка рабочая" if equipment_ok else
"связка НЕ ПРОХОДИТ проверку"}', ln=True)

# --- Ориентация солнечных панелей ---
pdf.ln(3)
pdf.set_font('DejaVu', '', 12)
pdf.cell(0, 8, 'Ориентация солнечных панелей', ln=True)
pdf.set_font('DejaVu', '', 10)
pdf.cell(0, 6, f'Оптимальная ориентация для широты {lat}°:', ln=True)
pdf.cell(0, 6, f' Угол наклона: {tilt_opt:.1f}°', ln=True)
pdf.cell(0, 6, f' Азимут: {az_opt:.1f}° (0=север, 180=юг)', ln=True)

# --- Дефицит и баланс ---
pdf.add_page()
pdf.set_font('DejaVu', '', 12)
pdf.cell(0, 8, 'Дефицит и баланс', ln=True)
pdf.set_font('DejaVu', '', 10)

daily_actual = full_range_filtered_hes_data['load_energy_step_wh'].sum()
pdf.cell(0, 6, f'Количество панелей: {numberpanel}', ln=True)
pdf.cell(0, 6, f'Активная площадь массива: {pv_array_active_area_m2:.2f}
м²', ln=True)
pdf.cell(0, 6, f'Активная площадь одной панели: {panel_active_area_m2:.3f} м²', ln=True)
pdf.cell(0, 6, f'Модулей АКБ в серии: {battery_series_count}, параллель-
ных веток: {battery_parallel_count}, всего АКБ: {numberbattery}', ln=True)
pdf.cell(0, 6, f'Напряжение банка: {battery_bank_voltage:.2f} В',
ln=True)
pdf.cell(0, 6, f'Доступная ёмкость: {batteryCapacityLim:.2f} А·ч, факти-
ческая: {batteryCapacity:.2f} А·ч', ln=True)
pdf.cell(0, 6, f'Мин. допустимая (контроль): {battery_DoD:.2f} А·ч',
ln=True)
pdf.cell(0, 6, f'Контрольный минимум энергии: {battery_energy_control_min_wh:.2f} Вт·ч', ln=True)
pdf.cell(0, 6, f'Диапазон энергии АКБ: {battery_energy_floor_wh:.2f}..{battery_energy_max_wh:.2f} Вт·ч', ln=True)
pdf.cell(0, 6, f'Суточная энергия нагрузки: {daily_actual:.2f} Вт·ч',
ln=True)
pdf.cell(0, 6, f'Макс. дефицит после PV: {peak_pv_deficit_w:.2f} Вт',
ln=True)

```

```

pdf.cell(0, 6, f'Макс. дефицит после АКБ: {peak_battery_deficit_w:.2f}
Вт', ln=True)
pdf.cell(0, 6, f'Потребление из сети: {total_grid_energy_wh:.2f} Вт·ч',
ln=True)
pdf.cell(0, 6, f'Суммарный заряд АКБ: {total_battery_charge_wh:.2f} Вт,
разряд: {total_battery_discharge_wh:.2f} Вт', ln=True)
if len(full_range_filtered_hes_data) > 0:
    start_e = full_range_filtered_hes_data.iloc[0]['battery_energy']
    end_e = full_range_filtered_hes_data.iloc[-1]['battery_energy']
    pdf.cell(0, 6, f'Начальный заряд: {start_e:.2f} Вт·ч, конечный:
{end_e:.2f} Вт·ч', ln=True)

# --- Графики (все) ---
pdf.add_page()
pdf.set_font('DejaVu', '', 12)
pdf.cell(0, 8, 'Графики', ln=True)
pdf.ln(2)

graphs = [
    ('solar_flux_clear.png', 'Солнечный поток (Clear Sky)'),
    ('cloudiness.png', 'Облачность'),
    ('temperature.png', 'Температура'),
    ('solar_flux_cloudy.png', 'Солнечный поток с учётом облачности'),
    ('energy.png', 'Выработанная и необходимая энергия'),
    ('battery_balance.png', 'Баланс энергии в АКБ'),
    ('charge_discharge.png', 'Заряд и разряд АКБ'),
    ('grid_import.png', 'Потребление из сети'),
    ('deficit.png', 'Дефицит и покрытие АКБ'),
]

for fname, title in graphs:
    path = os.path.join(img_dir, fname)
    if os.path.exists(path):
        pdf.set_font('DejaVu', '', 10)
        pdf.cell(0, 6, title, ln=True)
        pdf.image(path, x=10, w=190)
        pdf.ln(4)

# --- Периоды дефицита (без АКБ) ---
if 'deficit_df' in globals() and not deficit_df.empty:
    pdf.add_page()
    pdf.set_font('DejaVu', '', 12)
    pdf.cell(0, 8, 'Периоды дефицита энергии (без учёта АКБ)', ln=True)
    pdf.set_font('DejaVu', '', 8)
    pdf.multi_cell(0, 5, tabulate(deficit_df, headers='keys', table-
fmt='grid', showindex=False))
    pdf.ln(2)
    pdf.set_font('DejaVu', '', 10)
    if 'total_deficit_energy_wh' in globals():
        pdf.cell(0, 6, f'Суммарный дефицит: {to-
tal_deficit_energy_wh:.2f} Вт·ч', ln=True)

```

```

# --- Низкий заряд АКБ ---
if 'battery_control_min_periods' in globals():
    pdf.ln(3)
    pdf.cell(0, 6, f'Периодов с энергией АКБ <= {battery_energy_control_min_wh:.2f} Вт·ч: {len(battery_control_min_periods)}',
ln=True)

# --- Подключения к сети ---
if 'connect_df' in globals() and not connect_df.empty:
    pdf.ln(3)
    pdf.set_font('DejaVu', '', 12)
    pdf.cell(0, 8, 'Подключения к электросети для зарядки АКБ', ln=True)
    pdf.set_font('DejaVu', '', 8)
    pdf.multi_cell(0, 5, tabulate(connect_df, headers='keys', tablefmt='grid', showindex=False))

pdf_output_path = os.path.join(LOCAL_BASE_PATH,
'отчёт_моделирования.pdf')
pdf.output(pdf_output_path)
return pdf_output_path, full_range_filtered_hes_data

### app.py ###
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
import os
from streamlit_folium import st_folium
import folium
from folium.plugins import Geocoder
import leafmap.foliummap as leafmap

from solar_calc import run_simulation

st.set_page_config(page_title="Солнечная электростанция", layout="wide")

st.markdown("""
<style>
    .stApp { background: linear-gradient(135deg, #2b1b4e 0%, #5b4c7a 30%,
#e0c3b0 70%, #f9e4b7 100%); background-attachment: fixed; }
    .stButton > button { background-color: rgba(28,4,123,0.2); border: 2px
solid #1c047b; border-radius: 30px; color: #1c047b; font-weight: bold; }
    .stButton > button:hover { background-color: rgba(28,4,123,0.4); }

div[data-testid="stAlert"] {
    background-color: transparent !important;
    border-left-color: #1c047b !important;
    color: #1c047b !important;
}

```

```

div[data-testid="stAlert"] .stMarkdown {
    color: #1c047b !important;
}
iframe[title="streamlit_folium.st_folium"] {
    width: 100% !important;
    border-radius: 15px;
    border: none;
}
</style>
""" , unsafe_allow_html=True)

# ----- ИНИЦИАЛИЗАЦИЯ СОСТОЯНИЯ -----
if 'lat' not in st.session_state:
    st.session_state.lat = 50.739537
if 'lon' not in st.session_state:
    st.session_state.lon = 136.567232
if 'show_map' not in st.session_state:
    st.session_state.show_map = True
if 'calculation_done' not in st.session_state:
    st.session_state.calculation_done = False

# ----- СИНХРОНИЗАЦИЯ ВРЕМЕННЫХ КООРДИНАТ С ОСНОВНЫМИ -----
if 'map_lat' in st.session_state and 'map_lon' in st.session_state:
    st.session_state.lat = st.session_state.map_lat
    st.session_state.lon = st.session_state.map_lon
    del st.session_state.map_lat
    del st.session_state.map_lon

# ----- БОКОВАЯ ПАНЕЛЬ (все параметры) -----
with st.sidebar:
    st.title("⚙️ Параметры системы")
    st.markdown("---")

    # 1. Местоположение и даты
    st.subheader("📍 Местоположение")
    # Поля ввода связаны с session_state через key
    lat = st.number_input("Широта", value=st.session_state.lat, format="%.6f")
    lon = st.number_input("Долгота", value=st.session_state.lon, format="%.6f")
    tz = st.selectbox("Часовой пояс", ["Asia/Vladivostok", "Europe/Moscow", "Asia/Yekaterinburg", "UTC"], index=0)

    st.subheader("📅 Период моделирования")
    col1, col2 = st.columns(2)
    with col1:
        start_date = st.date_input("Дата начала", datetime(2026, 1, 1))
    with col2:
        end_date = st.date_input("Дата окончания", datetime(2026, 1, 7))
    start_str = start_date.strftime("%Y-%m-%dT00:00")
    end_str = end_date.strftime("%Y-%m-%dT23:59")

```

```

# 2. Нагрузка
st.subheader("⚡ Параметры нагрузки")
max_load_power_kw = st.number_input("Максимальная нагрузка системы (кВт·ч)", value=175.0)
max_load_work_hours = st.number_input("Рабочие часы предприятия (ч)", value=9.0)
st.markdown("***Часы нагрузки (0-23)**")
c1, c2 = st.columns(2)
with c1:
    load_min_start = st.number_input("Начало мин. нагрузки", value=0, min_value=0, max_value=23, key="min_start")
    load_min_end = st.number_input("Конец мин. нагрузки", value=8, min_value=0, max_value=24, key="min_end")
with c2:
    load_max_start = st.number_input("Начало макс. нагрузки", value=8, min_value=0, max_value=23, key="max_start")
    load_max_end = st.number_input("Конец макс. нагрузки", value=13, min_value=0, max_value=24, key="max_end")
    load_normal_start = st.number_input("Начало обычной нагрузки", value=13, min_value=0, max_value=23)
    load_normal_end = st.number_input("Конец обычной нагрузки", value=24, min_value=0, max_value=24)

# 3. Панель
st.subheader("☀ Солнечная панель")
photoEfficiency = st.number_input("КПД панели (0..1)", value=0.23, format="%.3f")
k_photoEffTemp = st.number_input("Температурный коэффициент (%/°C)", value=-0.0029, format="%.4f")
photoNOCT = st.number_input("NOCT (°C)", value=43.0)
panel_Vmp = st.number_input("Рабочее напр-е (В)", value=45.40)
panel_Voc = st.number_input("Напр-е холостого хода (В)", value=53.80)
panel_Imp = st.number_input("Ток к точке макс. мощности (А)", value=10.14)
panel_Isc = st.number_input("Ток короткого замыкания (А)", value=10.81)
photoCellWidth = st.number_input("Ширина ячейки (м)", value=0.210)
photoCellHigh = st.number_input("Высота ячейки (м)", value=0.210)
photoCellNum = st.number_input("Количество ячеек", value=144, step=1)

# 4. Аккумулятор
st.subheader("🔋 Аккумулятор")
numberbattery_1 = st.number_input("Начальное кол-во АКБ", value=1, step=1)
charge_voltage = st.number_input("Напряжение заряда (В)", value=57.6)
battery_nominal_ah = st.number_input("Ёмкость одного АКБ (А·ч)", value=100.0)
battery_voltage = st.number_input("Рабочее напряжение (В)", value=51.2)
k_min_SoC = st.number_input("Мин. SoC (0..1)", value=0.1, format="%.2f")
k_mode_battery = st.number_input("Кoeff. доступной ёмкости", value=0.8)
kCharge = st.number_input("Макс. ток заряда (от ёмкости)", value=0.3)

```

```

# 5. Инвертор
st.subheader("⚡ Инвертор")
inverter_nominal_power = st.number_input("Номинальная мощность (кВт)",
value=25.0)
inverter_efficiency = st.number_input("КПД (0..1)", value=0.976, for-
mat="%.3f")
inverter_battery_voltage_nominal = st.number_input("Ном. напряжение АКБ
(B)", value=400)
inverter_battery_voltage_min = st.number_input("Мин. напряжение АКБ
(B)", value=160)
inverter_battery_voltage_max = st.number_input("Макс. напряжение АКБ
(B)", value=700)
inverter_max_battery_charge_current = st.number_input("Макс. ток заряда
(A)", value=50.0)
inverter_max_battery_discharge_current = st.number_input("Макс. ток раз-
ряда (A)", value=50.0)
inverter_mppt_min = st.number_input("Мин. МРРТ напряжение (B)", val-
ue=150)
inverter_mppt_max = st.number_input("Макс. МРРТ напряжение (B)", val-
ue=850)
inverter_pv_max_voltage = st.number_input("Макс. входное PV (B)", val-
ue=1000)
inverter_pv_start_voltage = st.number_input("Пусковое PV (B)", val-
ue=180)
inverter_mppt_count = st.number_input("Кол-во МРРТ", value=2, step=1)
inverter_pv_nominal_power = st.number_input("Ном. входная PV мощность
(Вт)", value=40000)
inverter_pv_max_current_per_mppt = st.number_input("Макс. ток на МРРТ
(A)", value=26.0)
inverter_pv_max_isc_per_mppt = st.number_input("Макс. Isc на МРРТ (A)",
value=39.0)
max_inverters_in_parallel = st.number_input("Макс. инверторов параллель-
но", value=10, step=1)

# 6. Потери и запас
st.subheader("🔌 Потери")
k_cable_pv = st.number_input("Потери DC кабель (0..1)", value=0.97, for-
mat="%.2f")
k_cable_batt = st.number_input("Потери кабель АКБ", value=0.98)
k_cable_ac = st.number_input("Потери AC сторона", value=0.99)
reserve_factor_inverter = st.number_input("Запас мощности инвертора",
value=1.2)

# 7. Размеры крыши
st.subheader("🏠 Размер крыши")
roof_length = st.number_input("Длина крыши (м)", value=30.0)
roof_width = st.number_input("Ширина крыши (м)", value=30.0)

# Начальный SOC

```

```

    initial_battery_soc = st.slider("Начальный заряд АКБ (%)", 0, 100, 60) /
100.0

# ----- ОСНОВНАЯ ОБЛАСТЬ -----
st.markdown("# 🌞 Моделирование солнечной электростанции")
st.markdown("### Заполните параметры в боковой панели, затем нажмите кнопку
ниже")

if st.button("🚀 ЗАПУСТИТЬ РАСЧЁТ", use_container_width=True):
    # Скрываем карту
    st.session_state.show_map = False
    params = {
        'lat': lat, 'lon': lon, 'timezone': tz,
        'start_date': start_str, 'end_date': end_str,
        'max_load_power_kw': max_load_power_kw, 'max_load_work_hours':
max_load_work_hours,
        'load_min_start_hour': load_min_start, 'load_min_end_hour':
load_min_end,
        'load_max_start_hour': load_max_start, 'load_max_end_hour':
load_max_end,
        'load_normal_start_hour': load_normal_start, 'load_normal_end_hour':
load_normal_end,
        'photoEfficiency': photoEfficiency, 'k_photoEffTemp':
k_photoEffTemp, 'photoNOCT': photoNOCT,
        'panel_Vmp': panel_Vmp, 'panel_Voc': panel_Voc, 'panel_Imp': pan-
el_Imp, 'panel_Isc': panel_Isc,
        'photoCellWidth': photoCellWidth, 'photoCellHigh': photoCellHigh,
        'photoCellNum': photoCellNum,
        'numberbattery_1': numberbattery_1, 'charge_voltage':
charge_voltage,
        'battery_nominal_ah': battery_nominal_ah, 'battery_voltage': bat-
tery_voltage,
        'k_min_SoC': k_min_SoC, 'k_mode_battery': k_mode_battery, 'kCharge':
kCharge,
        'inverter_nominal_power': inverter_nominal_power, 'invert-
er_efficiency': inverter_efficiency,
        'inverter_battery_voltage_nominal': invert-
er_battery_voltage_nominal,
        'inverter_battery_voltage_min': inverter_battery_voltage_min,
        'inverter_battery_voltage_max': inverter_battery_voltage_max,
        'inverter_max_battery_charge_current': invert-
er_max_battery_charge_current,
        'inverter_max_battery_discharge_current': invert-
er_max_battery_discharge_current,
        'inverter_mppt_min': inverter_mppt_min, 'inverter_mppt_max': invert-
er_mppt_max,
        'inverter_pv_max_voltage': inverter_pv_max_voltage,
        'inverter_pv_start_voltage': inverter_pv_start_voltage,
        'inverter_mppt_count': inverter_mppt_count,
        'inverter_pv_nominal_power': inverter_pv_nominal_power,

```

```

        'inverter_pv_max_current_per_mppt': invert-
er_pv_max_current_per_mppt,
        'inverter_pv_max_isc_per_mppt': inverter_pv_max_isc_per_mppt,
        'max_inverters_in_parallel': max_inverters_in_parallel,
        'k_cable_pv': k_cable_pv, 'k_cable_batt': k_cable_batt,
        'k_cable_ac': k_cable_ac,
        'reserve_factor_inverter': reserve_factor_inverter,
        'roof_length': roof_length, 'roof_width': roof_width,
        'initial_battery_soc': initial_battery_soc
    }
    with st.spinner("Идёт расчёт (может занять несколько минут)..."):
        try:
            pdf_path, df_results = run_simulation(params)
            st.session_state['pdf_path'] = pdf_path
            st.session_state['df_results'] = df_results
            st.session_state['calculation_done'] = True
            st.success("☑ Расчёт завершён!")
        except Exception as e:
            st.error(f"Ошибка при расчёте: {e}")

# Карта (показываем, только если расчёт ещё не выполнен)
if not st.session_state.calculation_done and st.session_state.show_map:
    st.subheader("📍 Кликните по карте, чтобы выбрать местоположение")

    # Создаём карту с правильным тайлом
    m = folium.Map(
        location=[st.session_state.lat, st.session_state.lon],
        zoom_start=8,
        tiles='https://{s}.basemaps.cartocdn.com/light_all/{z}/{x}/{y}{r}.pn
g',
        attr='CartoDB'
    )
    folium.Marker([st.session_state.lat, st.session_state.lon],
tooltip="Текущая точка").add_to(m)

    # Поиск по городам
    from folium.plugins import Geocoder
    m.add_child(Geocoder(position='topright', collapsed=True,
placeholder='🔍 Поиск города...'))

# Карта
map_data = st_folium(m, width='100%', height=500)

# Кнопка сброса вида
col_btn, _ = st.columns([1, 3])
with col_btn:
    if st.button("🔄 Сбросить вид карты"):
        st.session_state.lat = 50.739537
        st.session_state.lon = 136.567232
        st.rerun()

```

```

if map_data and map_data.get('last_clicked'):
    clicked_lat = map_data['last_clicked']['lat']
    clicked_lon = map_data['last_clicked']['lng']
    st.session_state.map_lat = clicked_lat
    st.session_state.map_lon = clicked_lon
    st.rerun()

st.caption("💡 Кликните по карте - координаты автоматически подставляются
в поля ввода")

# ----- ОТОБРАЖЕНИЕ РЕЗУЛЬТАТОВ -----
if st.session_state.get('calculation_done', False):
    df = st.session_state['df_results']
    st.subheader("📊 Результаты моделирования")

    col_gr1, col_gr2 = st.columns(2)
    fig1, ax = plt.subplots(figsize=(8,4))
    ax.plot(df['datetime'], df['solar_energy'], label='Выработанная', col-
or='blue')
    ax.plot(df['datetime'], df['load_energy_wh'], label='Необходимая', col-
or='orange')
    ax.set_xlabel('Дата'); ax.set_ylabel('Вт·ч')
    ax.set_title('Выработанная и необходимая энергия')
    ax.grid(True); ax.legend()
    col_gr1.pyplot(fig1)

    fig2, ax = plt.subplots(figsize=(8,4))
    ax.plot(df['datetime'], df['battery_energy'], color='blue')
    ax.axhline(y=df['battery_min_energy_wh'].iloc[0], color='black', lin-
estyle='--', label='Мин. порог')
    ax.set_xlabel('Дата'); ax.set_ylabel('Вт·ч')
    ax.set_title('Энергия в АКБ')
    ax.grid(True); ax.legend()
    col_gr2.pyplot(fig2)

    col_gr3, col_gr4 = st.columns(2)
    fig3, ax = plt.subplots(figsize=(8,4))
    ax.plot(df['datetime'], df['battery_charge_power_w'], label='Заряд от
PV')
    ax.plot(df['datetime'], df['battery_discharge_power_w'], label='Разряд
АКБ')
    ax.set_xlabel('Дата'); ax.set_ylabel('Вт')
    ax.set_title('Заряд и разряд АКБ')
    ax.grid(True); ax.legend()
    col_gr3.pyplot(fig3)

    fig4, ax = plt.subplots(figsize=(8,4))
    ax.plot(df['datetime'], df['balance_no_battery_w'], color='red',
label='Баланс без АКБ')
    ax.fill_between(df['datetime'], 0, df['balance_no_battery_w'],
where=(df['balance_no_battery_w']<0), color='red', alpha=0.3)

```

```

    ax.plot(df['datetime'], df['balance_with_battery_w'], color='green',
linestyle='--', label='С АКБ')
    ax.axhline(y=0, color='black', linewidth=0.5)
    ax.set_xlabel('Дата'); ax.set_ylabel('Вт')
    ax.set_title('Дефицит генерации')
    ax.grid(True); ax.legend()
    col_gr4.pyplot(fig4)

    st.subheader("📊 Сводная статистика")
    col_s1, col_s2, col_s3 = st.columns(3)
    col_s1.metric("Пиковый дефицит (после АКБ)",
f"{df['deficit_after_battery_w'].max():.1f} Вт")
    col_s2.metric("Суммарное потребление из сети",
f"{df['grid_energy_step_wh'].sum():.1f} Вт·ч")
    col_s3.metric("Конечный заряд АКБ", f"{df['battery_energy'].iloc[-
1]:.1f} Вт·ч")

    with open(st.session_state['pdf_path'], 'rb') as f:
        pdf_bytes = f.read()
    st.download_button(
        label="📄 Скачать PDF-отчёт",
        data=pdf_bytes,
        file_name="отчёт_моделирования.pdf",
        mime="application/pdf",
        use_container_width=True
    )
else:
    st.markdown('<p style="color: #1c047b; background-color: transparent;
margin: 10px 0;">🔗 Введите все параметры и нажмите «ЗАПУСТИТЬ РАСЧЁТ»</p>',
unsafe_allow_html=True)

```