

Алгоритм управления и динамика реабилитационного робота нижних конечностей на основе инновационного электропривода

ОГЛАВЛЕНИЕ **Error! Bookmark not defined.**

ВВЕДЕНИЕ	5
ГЛАВА 1. Введение в роботизированную систему реабилитации нижних конечностей	16
1.1 Статус применения роботов для реабилитации нижних конечностей	19
1.1.1 Применение роботов для реабилитации нижних конечностей в Китае	19
1.1.2 Применение роботов для реабилитации нижних конечностей в других странах.....	22
1.2 Основные методы управления реабилитационными роботами	27
1.3 Основные проблемы текущих исследований	29
1.4 Основное исследовательское содержание данной диссертации.	32
Выводы по первой главе	33
Глава 2 Проектирование и анализ роботизированной системы реабилитации нижних конечностей	35
2.1 Анализ движений нижних конечностей человека	35
2.1.1 Биомеханические параметры движений нижних конечностей человека	36
2.1.2 Анализ траектории пространственного движения поясницы	37

2.1.3 Моделирование и симуляция в OpenSim	38
2.2 Систематическая конструкция робота для реабилитации нижних конечностей	40
2.2.1 Проектирование механической конструкции	41
2.2.2 Проектирование системы управления	47
2.2.3 Выбор электрических компонентов	48
2.3 Анализ методом конечных элементов для робота реабилитации нижних конечностей	52
Выводы по второй главе	57
Глава 3 Математическая модель и алгоритм управления роботом для реабилитации нижних конечностей	59
3.1 Алгоритм управления опорной частью шагающего робота для реабилитации нижних конечностей	59
3.1.1 Кинематический анализ опорной части робота	59
3.1.2 Динамический анализ опорной части робота	64
3.2 Методы управления опорной частью робота	74
3.2.1 Адмитанс и PD-контроль	75
3.2.2 Скользящее управление	79
3.3 Алгоритм управления шагающим шасси робота для реабилитации нижних конечностей	84
3.3.1 Кинематический анализ шагающего шасси	85
3.3.2 Динамический анализ шагающего шасси	86

3.3.3 Процесс нечеткого управления ходовой платформой.....	87
3.3.4 Проектирование нечеткого управления ходовой платформой	92
Выводы по третьей главе	102
Глава 4. Экспериментальная проверка робота для реабилитации нижних конечностей	104
4.1 Трёхмерное динамическое отслеживание движения таза	104
4.1.1 Подготовка к эксперименту.....	104
4.1.2 Экспериментальный процесс	105
4.1.3 Обработка экспериментальных данных	108
4.2 Обработка сигналов датчиков с использованием фильтрации.....	110
4.2.1 Фильтр Калмана.....	110
4.2.2 FIR-фильтрация	113
4.2.3 IIR-фильтрация	115
4.3 Экспериментальная проверка движения робота для реабилитации нижних конечностей	119
4.3.1 Описание условий эксперимента.....	119
4.3.2 Проверка ходьбы.....	123
Выводы по четвертой главе.....	128
ЗАКЛЮЧЕНИЕ	129
ЛИТЕРАТУРА	132
Приложение 1	144

СПИСОК СОКРАЩЕНИЙ

ЭЭГ — электроэнцефалограмма

ЭМГ — электромиограмма

D-H — параметры Денавита–Хартенберга

IIR — фильтр с бесконечным импульсным откликом

FIR — фильтр с конечным импульсным откликом

ВВЕДЕНИЕ

Актуальность работы. Нарушение функций нижних конечностей является распространенным последствием таких серьезных заболеваний, как инсульт и травмы спинного мозга, что приводит к ослаблению или потере способности ходить, серьезно влияя на способность пациентов к независимой жизни и их качество жизни. Традиционные методы реабилитации, как правило, зависят от ручной помощи реабилитационных терапевтов, ограничиваются нехваткой человеческих ресурсов, нестабильностью интенсивности тренировок и недостаточной индивидуализацией, что затрудняет полное удовлетворение потребностей пациентов. Таким образом, появился робот для реабилитации нижних конечностей, объединяющий междисциплинарные технологии, такие как мехатроника, теория управления и взаимодействие человека с машиной. В настоящей работе описан робот для реабилитации нижних конечностей, который благодаря функции поддержки поясничного отдела, ходьбе и динамическому сопровождению движений поясницы значительно повышает комфорт и безопасность. Используя разгрузочную поддержку и интеллектуальную помощь, он лучше помогает пациентам в выполнении повторяющихся и целенаправленных тренировок походки, ускоряя процесс реабилитации и снижая затраты на нее, что имеет значительное социальное и экономическое значение. С учетом ускоряющегося старения населения в большинстве стран и нехватки специалистов по реабилитации проведение

систематических исследований и внедрение роботов для реабилитации нижних конечностей в клиническую практику и их коммерциализация являются крайне актуальными и важными задачами.

Целью диссертационного исследования является разработка и реализация системы реабилитационного робота для нижних конечностей, оснащённого функцией разгрузочной поддержки и интеллектуальными алгоритмами управления, которая обеспечивает гибкое сопровождение и помощь в движениях поясницы пациента в трёхмерных направлениях, создавая безопасную, комфортную и эффективную среду для реабилитации.

Для достижения поставленной цели в работе решены следующие задачи:

1. Проведены системные исследования и анализ текущего состояния и типичных приложений реабилитационных роботов для нижних конечностей в стране и за рубежом, определены требования к дизайну и технические узкие места.

2. На основе особенностей трёхмерного движения тазобедренного сустава человека разработана конструкция механического корпуса, включая модули подъёма, горизонтального перемещения и двустороннего противоположного движения, а также выполнена верификация прочности методом конечных элементов.

3. Построены кинематическая и динамическая модели робота (с использованием метода Денавіта-Хартенберга и уравнений Лагранжа), обеспечивая теоретическую основу для дальнейшей разработки алгоритмов управления.

4. Исследованы и сравнены различные стратегии управления: в модуле разгрузки (подъём) реализованы комбинации адаптивного управления с PID-регулятором и адаптивного управления с методом скользящего режима, а в модуле ходьбы применён метод нечеткого управления.

5. Проведен сравнительный анализ фильтрации сигналов датчиков (силового и крутящего момента) с использованием фильтров Калмана, FIR и IIR, выбрана оптимальная схема фильтрации.

6. На экспериментальном прототипе проверена реализуемость и эффективность предложенной системы, включая испытания по захвату трёхмерных движений поясницы, тесты помощи при ходьбе и оценку эффективности фильтрации сигналов.

Объект исследования.

Оказывать помощь в реабилитации пациентов с нарушениями двигательных функций нижних конечностей позволяет реабилитационный робот, оснащённый механизмом поддержки поясницы, обеспечивающим её вертикальное и горизонтальное сопровождение, а также ходовым шасси, следящим за движениями пациента. Оба компонента приводятся в действие сервоприводами, а определение двигательных намерений

пациента осуществляется на основе данных с датчиков давления и крутящего момента, регистрирующих усилия в области поясницы.

Методы исследований.

1. Теоретический анализ и численное моделирование

Построение модели кинематических характеристик движения тазобедренного сустава человека с использованием программного обеспечения OpenSim и знаний биомеханики.

Разработка кинематической и динамической модели робота с применением метода Денавіта-Хартенберга (D-H) и уравнений Лагранжа.

Проведение анализа методом конечных элементов (Finite Element Analysis, FEA) основных конструктивных элементов с использованием ANSYS Workbench.

2. Исследование алгоритмов управления

Применение алгоритмов управления типа "адмитанс + PID" и "адмитанс + скользящий режим" для модулей подъёма и разгрузки.

Использование нечеткого управления для регулирования скорости движения и поворотов ходового шасси.

Сравнительный анализ методов фильтрации сигналов датчиков (фильтр Калмана, FIR и IIR) для предварительной обработки данных.

3. Экспериментальные исследования

Измерение трёхмерных траекторий движения поясницы с помощью оптической системы захвата движений Optitrack.

Комплексное тестирование предложенных методов управления на экспериментальном прототипе, включая оценку плавности движений, взаимодействия "человек-машина" и эффективности разгрузочной поддержки.

Научная новизна работы

1.Интегрированная структура "разгрузочная поддержка + многостепенная поддержка поясницы":

Реализация модулей подъёма, горизонтального перемещения и двустороннего противоположного движения в одном роботе. Эти модули обеспечивают гибкое сопровождение трёхмерных движений тазобедренного сустава пациента, улучшая естественность и безопасность реабилитационной тренировки.

2.Предложение стратегии подъёма на основе комбинации адмитанса с традиционным/скользящим управлением:

Алгоритм "адмитанс + PID" демонстрирует хорошую стабильность и комфорт при сложных человеко-машинных взаимодействиях, в то время как "адмитанс + скользящий режим" обладает высокой робастностью, но сопряжён с риском появления дрожания.

3.Метод нечеткого управления для ходового шасси:

Определение намерений движения вперёд и поворота в реальном времени осуществляется на основе суммарной силы и разницы сил с левого и правого датчиков. Метод не требует точной математической модели и адаптируется к индивидуальным различиям пациентов.

4. Сравнение фильтров и выбор фильтра Калмана:

Использование фильтра Калмана в дополнение к цифровым преобразователям давления значительно повышает плавность сигнала, снижает дрожание сервоприводов и улучшает пользовательский опыт пациентов.

Научные результаты, выносимые на защиту:

1. В области механической структуры:

Завершён общий дизайн конструкции робота и проведён анализ прочности методом конечных элементов. Результаты показали, что при нагрузке 120 кг система остаётся безопасной и надёжной.

2. В области стратегий управления:

В процессе подъёма (разгрузки) комбинация адмитанса с PID-регулятором демонстрирует более плавный контроль по сравнению с адмитансом и скользящим режимом.

Для ходового шасси использован метод нечеткого управления, который регулирует скорость и направление на основе суммарной силы и разницы показаний датчиков, что обеспечивает простоту реализации и хорошую адаптивность.

3.В области обработки сигналов:

После сравнения трёх методов фильтрации (Калмана, FIR и IIR) выяснилось, что фильтр Калмана позволяет эффективно подавлять шум и дрожание, сохраняя высокую скорость отклика.

4.В области экспериментальной проверки:

Проведён захват трёхмерных движений поясницы с использованием системы Optitrack, а также многочисленные испытания прототипа на помощь в ходьбе и разгрузке. Результаты подтвердили, что система точно понимает и следует намерениям пациента, обеспечивая стабильную, безопасную и эффективную поддержку реабилитации.

Научная новизна работы заключается в следующем:

Математическая модель и её применение в ниже-конечностной роботизированной системе реабилитации

Построение математической модели движения многозвенного робота на основе метода Денавита-Хартенберга (D-H) и уравнений Лагранжа позволяет проводить детальный кинематический и динамический анализ системы. Это обеспечивает теоретическую поддержку для разработки и оптимизации сложных роботизированных систем.

Адаптивное управление движением робота для реабилитации

Комбинированный алгоритм управления, основанный на методах импедансного управления и PID-регулирования, обеспечивает гибкость

взаимодействия с пациентом, улучшает точность управления и снижает риск дискомфорта или травм пациента.

Система управления на основе данных от множества сенсоров

Интеграция давления и крутящего момента с использованием встроенных датчиков позволяет точно идентифицировать намерения пациента. Реализация модифицированного нечеткого управления обеспечивает плавность и естественность движений робота.

Технологии фильтрации сигналов

Сравнительный анализ различных методов фильтрации сигналов выявил, что использование алгоритма фильтрации Калмана значительно повышает точность измерений и снижает уровень шумов, что критически важно для работы систем управления в реальном времени.

Основные правила нечеткой логики

Разработан многоцелевой нечеткий контроллер с использованием многовходовых и многовыходовых подходов. Для обеспечения гибкости управления диапазон переменных разделён на множество подмножеств. Применение метода основных правил эффективно устраняет избыточность данных, предоставляя теоретическую поддержку для работы с большим количеством управляющих правил в нечеткой системе.

Достоверность научных и практических результатов. Была разработана и реализована механическая структура робота для реабилитации нижних конечностей. Проведено моделирование

динамического сопровождения подъёмного модуля с использованием алгоритмов управления на основе адмитанса и PID. Реализовано динамическое сопровождение ходового шасси с применением нечеткого управления. Эффективность системы подтверждена экспериментальной проверкой на прототипе.

Структура диссертации.

Диссертационная работа состоит из введения, четырех глав с выводами, заключения, списка литературы и приложения. Общий объем диссертации 242 с., в том числе 128 с. основного текста, 50 рисунка, 11 таблиц, список литературы из 104 наименований и приложение на 100 страницах.

Глава 1: Представляет исследовательский фон и значимость разработки реабилитационного робота для нижних конечностей, подчёркивая необходимость изучения данной темы. Рассматривается текущее состояние исследований в области реабилитационных роботов, обобщаются существующие методы управления, их преимущества и недостатки, а также излагается содержание настоящего исследования.

Глава 2: Анализируются пространственные траектории движения тазобедренного сустава человека, а также проводятся симуляции движений тазобедренного сустава с использованием OpenSim. Результаты показывают, что тазобедренный сустав выполняет сложные движения в трёхмерном пространстве. На основе этого анализа разработана и

реализована общая конструкция реабилитационного робота для нижних конечностей. Для повышения комфорта разработан механизм с регулируемой шириной в области поясницы, адаптирующийся к различным типам телосложения пользователей. В соответствии с требованиями к управлению реабилитационной ходьбой была построена структура системы управления и выбраны подходящие электрические компоненты, обеспечивающие стабильность и надёжность системы. Для проверки рациональности и безопасности конструкции робота проведён статический и динамический анализ с использованием Workbench, что подтвердило достаточную прочность и устойчивость робота.

Глава 3: С помощью метода Денавіта-Хартенберга (D-H) построена кинематическая модель реабилитационного робота для нижних конечностей, проведён кинематический анализ. Для дальнейшего изучения динамического поведения робота проведён динамический анализ с использованием метода Лагранжа. Для реализации мягкого следящего управления в модуле подъёма использованы алгоритмы адмитансного управления в сочетании с PID-регулятором, что обеспечивает плавный отклик робота на изменения внешних сил. В ходе экспериментов определена связь между переднезадними движениями поясницы испытуемого и расстоянием ходьбы, на основе чего разработана соответствующая стратегия управления. Применён алгоритм нечеткого управления, что позволило реализовать динамическое сопровождение

между испытуемым и роботом, обеспечив стабильность и адаптивность робота в процессе ходьбы, а также повысив комфорт и эффективность реабилитации испытуемого.

Глава 4: Проведены эксперименты по изучению движений человеческого тела при ходьбе с использованием оборудования для трёхмерного захвата движений, собраны данные о движении тазобедренного сустава. На основании анализа получены траектории положения тазобедренного сустава, подтверждающие пространственные закономерности движения сустава и диапазон изменений его высоты во время ходьбы, что стало важной основой для конструктивного проектирования робота. Для повышения точности и стабильности сигналов датчиков применены три распространённых метода фильтрации, проверена их эффективность обработки данных. В ходе экспериментов протестированы модуль ходового шасси и модуль подъёма в разгрузочной системе, что позволило успешно реализовать мягкое управление роботом со стороны испытуемого, обеспечив динамическую адаптивность и комфортность системы.

ГЛАВА 1. Введение в роботизированную систему реабилитации нижних конечностей

Робот для реабилитации нижних конечностей, как новый тип интеллектуального носимого устройства, объединяет в себе достижения таких дисциплин, как механика, электроника, автоматическое управление и вычислительная техника. Это одна из наиболее передовых областей научных исследований в мире, которая уже доказала свою эффективность в оказании помощи пациентам с ослабленной или утраченной функцией нижних конечностей [1]. Усиление старения населения и сокращение численности трудоспособного населения неизбежно приведут к росту нагрузки на медицинскую отрасль. Кроме того, с возрастом физические способности ухудшаются, а риск нарушений двигательной функции нижних конечностей, вызванных такими заболеваниями, как инсульт и сердечно-сосудистые заболевания, значительно увеличивается [2]. Это серьёзно влияет на качество жизни как пациентов, так и их семей.

Для снижения бремени пациентов и их семей реабилитационная тренировка нижних конечностей, направленная на восстановление двигательных функций, имеет практическое значение [3]. Выбор безопасного и эффективного метода тренировок имеет первостепенное значение для пациентов. Однако традиционные методы реабилитации в основном зависят от непосредственного участия реабилитационных терапевтов. В Китае существует значительный разрыв между спросом на реабилитационных терапевтов и их наличием, особенно по сравнению с развитыми странами

Запада. Кроме того, обучение реабилитационных специалистов требует много времени, что означает, что в краткосрочной перспективе будет сохраняться серьёзная нехватка специалистов [4], что затрудняет удовлетворение потребностей большинства пациентов в Китае.

Чтобы решить проблему нехватки медицинских ресурсов, исследователи по всему миру активно занимаются разработкой реабилитационных роботов для нижних конечностей [5,6]. Такие роботы представляют собой новую форму интеллектуальных медицинских устройств и объединяют знания из робототехники, взаимодействия человека с машиной, реабилитационной медицины и теории управления [7]. В сфере помощи пожилым людям реабилитационные роботы для нижних конечностей способны значительно улучшить силу конечностей, эффективно предотвращать случайные падения и повышать уровень самостоятельности и качество жизни [8]. Постоянные тренировки помогают пожилым людям избежать проблем, вызванных недостаточной активностью, таких как мышечная атрофия и остеопороз.

В сфере помощи людям с ограниченными возможностями реабилитационные роботы предоставляют пациентам с утратой двигательных функций или нарушениями ходьбы возможность вновь встать и начать тренировки по ходьбе. Эта технология не только предотвращает осложнения, такие как пролежни, атрофия мышц и контрактуры суставов, вызванные длительным постельным режимом, но и улучшает кровообращение, снижает риски, связанные с остеопорозом [9].

С быстрым развитием мировой экономики и технологий инновации становятся основным драйвером обеспечения здоровья населения в разных странах, оказывая глубокое влияние на развитие медицинской реабилитации по всему миру. Эффективность тренировок с использованием реабилитационных роботов для нижних конечностей уже получила предварительное подтверждение в ходе клинических испытаний [10,11]. В процессе реабилитации безопасность и стабильность тренировок играют ключевую роль в обеспечении успешного восстановления [12].

В зависимости от стадии реабилитации пациента целевые методы управления тренировками могут повысить мотивацию пациента к продолжению реабилитации. Таким образом, правильно спроектированная механическая структура и система управления имеют решающее значение для всего процесса восстановления.

На глобальном уровне реабилитационные роботы для нижних конечностей стали одним из наиболее эффективных инструментов в области реабилитационного медицинского оборудования [13,14]. Их основная ценность заключается в реализации точных реабилитационных тренировок пациентов за счёт научно обоснованного проектирования конструкции робота и передовых стратегий управления. Рациональная конструкция робота позволяет эффективно адаптироваться к физиологическим особенностям пациента, а современные стратегии управления обеспечивают точность и безопасность тренировок, что помогает пациентам быстрее восстанавливать способность к ходьбе.

1.1 Статус применения роботов для реабилитации нижних конечностей

Робот для реабилитации нижних конечностей представляет собой инновационный продукт, объединяющий в себе достижения множества научных дисциплин, который вызывает большой интерес у университетов и научно-исследовательских учреждений по всему миру [15]. В развитых странах, которые раньше других столкнулись с проблемой старения населения, учёные начали исследования оборудования для вспомогательной реабилитации ещё в 1980-х годах и разработали различные типы роботов для реабилитации нижних конечностей. С развитием экономики в начале XXI века исследования в этой области начали проводиться и в Китае. Несмотря на то, что Китай начал изучение реабилитационных роботов для нижних конечностей позже, в последние годы этому направлению уделяется всё больше внимания. Многие университеты, научно-исследовательские учреждения и технологические компании вложили значительные усилия в исследования, добившись определённых успехов. Однако в настоящее время большинство разработок остаётся на стадии исследований, и лишь немногие продукты вышли на рынок [16].

1.1.1 Применение роботов для реабилитации нижних конечностей в Китае

В области исследований реабилитационных роботов для нижних конечностей Китай находится на начальной стадии развития. Растущие потребности населения в улучшении здоровья предъявляют всё более высокие требования к эффективности и результативности

реабилитационного лечения. На основе этих потребностей университеты и исследовательские учреждения в сотрудничестве с предприятиями уже разработали несколько поколений реабилитационных роботов для нижних конечностей, которые начали внедряться в практику. Способности к самостоятельным разработкам постепенно усиливаются, однако по сравнению с передовыми зарубежными роботами высокого уровня в этой области остаётся значительный разрыв [17,18]. Ниже приведён краткий обзор реабилитационных роботов для нижних конечностей, разработанных китайскими университетами и предприятиями.

(1) Реабилитационный робот GR-A1 для нижних конечностей

GR-A1 — это реабилитационный робот для тренировки походки с разгрузкой веса, разработанный компанией "Manbuzhe" (Tianjin, Китай) в 2018 году [19], как показано на рисунке 1.1. На верхней части робота установлено подвесное устройство для разгрузки веса, оснащённое уникальной системой рельсового перемещения. Реабилитационный терапевт с помощью дистанционного пульта управления управляет устройством, поднимая пациента из инвалидного кресла в положение стоя и перемещая его к месту тренировки. Это значительно снижает нагрузку на реабилитационного терапевта. Система тренировок оснащена устройством коррекции походки нижних конечностей, которое обеспечивает поддержку движений и эффективно улучшает результаты реабилитации.

университетом, ускоряет внедрение технологий экзоскелетов в различных областях. Эта платформа позволяет университетам, научно-исследовательским институтам и клиническим центрам выполнять вторичную разработку экзоскелетов на её основе.

В 2023 году компания Guangzhou Yikang Medical Equipment Industrial Co., Ltd. провела исследование подвешенного робота для реабилитации походки. На рисунке 1.3 представлена их разработка — система тренировки и оценки походки серии АЗ, которая поддерживает переключение между активным и пассивным режимами тренировки в любое время. Система анализирует структуру тренировки, выполняет отслеживание и оценку процесса реабилитации, а затем на основе результатов оценки изменяет режим тренировки, что значительно повышает эффективность реабилитационного лечения пациентов [23].



Рисунок 1.3: Система тренировки и оценки походки АЗ



Рисунок 1.4: MotionMaker — робот для реабилитации нижних конечностей в сидячем положении

1.1.2 Применение роботов для реабилитации нижних конечностей в других странах

С 1980-х годов научно-исследовательские учреждения других стран последовательно разработали системы тренировок с использованием реабилитационных роботов для нижних конечностей, которые в зависимости от положения пациента при тренировке делятся на сидячие, стоячие и автономные носимые роботы. На сегодняшний день среди наиболее развитых реабилитационных роботов для нижних конечностей можно выделить швейцарские MotionMaker — сидячий робот для реабилитации нижних конечностей и Lokomat — робот для тренировки походки, турецкий Physiotherabot — сидячий робот для реабилитации одной конечности, немецкий LokoHelp — робот для тренировки нижних конечностей в стоячем положении, американский экзоскелет Ekso NR, японский экзоскелет HAL и израильский экзоскелет ReWalk [24]. Ниже представлен краткий обзор некоторых из этих роботов.

Робот для реабилитации нижних конечностей MotionMaker, разработанный Швейцарской федеральной политехнической школой Лозанны, представляет собой стационарную программируемую систему для тестирования и тренировки нижних конечностей, как показано на рисунке 1.4. Система состоит из двух экзоскелетных ортезов для нижних конечностей и блока управления для регулировки электрической стимуляции в реальном времени. Экзоскелетные ортезы оснащены встроенными двигателями и датчиками, которые позволяют пациенту выполнять движения ног в соответствии с заданными системой параметрами, такими как положение, скорость и угол. Цель устройства заключается в укреплении мышечной силы

нижних конечностей, повышении выносливости и улучшении координации движений суставов. Клинические испытания показали, что использование MotionMaker ускоряет восстановление функции походки у пациентов. Однако у каждого сустава экзоскелета имеется только одна степень свободы, что ограничивает движения ног только в сагиттальной плоскости [25-26].

Lokomat, разработанный швейцарской компанией Hocoma совместно с Швейцарской федеральной политехнической школой Цюриха, стал первым в мире коммерческим реабилитационным роботом для нижних конечностей, как показано на рисунке 1.5.



Рисунок 1.5: Lokomat — робот для реабилитации нижних конечностей



Рисунок 1.6: ReWalk — робот для реабилитации нижних конечностей

Lokomat — это робот для тренировки походки, разработанный швейцарской компанией Hocoma для пациентов с травмами спинного мозга и после инсульта. Робот включает в себя три основных компонента: систему разгрузки веса, механические экзоскелетные ноги и беговую дорожку [28-29]. Нижние конечности пациента фиксируются ремнями на механических ногах, каждая из которых имеет одну степень свободы в тазобедренном и коленном

суставах. Движение в сагиттальной плоскости осуществляется с помощью двигателей и шариковинтовой передачи. Система разгрузки веса эффективно снижает нагрузку на нижние конечности пациента, а скорость ленты беговой дорожки синхронизируется со скоростью механических ног, что помогает пациенту выполнять реабилитационные тренировки [30-31].

Кроме того, чтобы улучшить опыт и повысить интерес пациентов к повторяющимся и монотонным реабилитационным тренировкам, робот оснащён разнообразными виртуальными сценариями, которые усиливают вовлечённость пациентов. Система управления робота сложная и включает стратегии управления, такие как PD-контроль, управление импедансом и адаптивное управление. Эксперименты подтвердили эффективность этих стратегий [32].

ReWalk, разработанный израильской компанией ReWalk Robotics, показан на рисунке 1.6. Этот робот имеет простую и удобную в использовании операционную систему, позволяющую пользователям самостоятельно управлять устройством. Для снижения веса устройства применяются лёгкие материалы, а в голеностопном суставе установлены пружины для амортизации движений системы робота. Кроме того, высота и ширина робота могут регулироваться в зависимости от индивидуальных особенностей пользователей [33]. В системе управления используются две группы двигателей, которые независимо управляют тазобедренным и коленным суставами. Робот также способен распознавать изменения центра тяжести пользователя и на основе этих данных корректировать режим

движения, чтобы синхронизировать тело пользователя с роботом [34]. Клинические испытания показали, что использование робота ReWalk позволяет пациентам выполнять реабилитационные движения с минимальными энергозатратами, что значительно облегчает процесс восстановления [35].

В 2004 году японская компания **Cyberdyne** была основана и представила технологию гибридной вспомогательной конечности (HAL) [36]. Это первая в мире технология, которая улучшает, поддерживает и усиливает физические функции носителя на основе его намерений, также известная как «носимый робот». Самая новая модель экзоскелета компании называется **HAL-5** [37], как показано на рисунке 1.7(a). Высота этого экзоскелета составляет 1600 мм, а вес — 23 кг. Устройство работает на батарейном питании и оснащено датчиками, которые прикрепляются к поверхности кожи. Эти датчики обнаруживают «биоэлектрические сигналы» носителя и выполняют необходимые движения в соответствии с его намерениями. С точки зрения применения, данный экзоскелет помогает носителю выполнять основные движения в повседневной жизни. HAL-5 оснащён гибридной системой управления, которая демонстрирует отличные характеристики как в помещении, так и на улице [38,39].

На рисунке 1.7(b) представлена схема ношения экзоскелета робота компании Cyberdyne.



(a) Внешний вид **HAL-5**



(b) Применение **HAL-5**

Рисунок 1.7: Экзоскелетный робот компании Cyberdyne

1.2 Основные методы управления реабилитационными роботами

При использовании реабилитационных роботов для помощи пациентам в реабилитационных тренировках ключевым фактором является обеспечение безопасности и эффективности. Это не только влияет на результаты восстановления пациентов, но и напрямую связано с продвижением роботов в клинической практике. В этой связи исследователи разработали различные системы управления. Основные из них представлены ниже:

1. Системы управления, основанные на моделях

Системы этого типа используют математические модели, описывающие динамику робота или закономерности движений пациента, чтобы предсказывать и генерировать оптимальные управляющие команды, обеспечивая точную помощь в движениях. Типичные подходы включают:

Динамическая модель: описывает кинематические и динамические характеристики роботизированной системы. С помощью физического

моделирования или алгоритмов искусственного интеллекта достигается повышение точности управления [40].

Модель мышц: имитирует характеристики движений мышц человека, используя электромиографические сигналы (ЭМГ) для предсказания силы сокращения мышц и предоставления индивидуализированных реабилитационных решений. Однако из-за нестабильности сигналов такие системы редко применяются на практике [41-42].

2. Системы управления на основе физических параметров

Эти системы зависят от параметров, измеряемых в реальном времени (например, силы, скорости, положения), для корректировки состояния работы робота. Наиболее распространённые методы включают:

PD-контроль: используется для точного управления положением, особенно подходит для задач фиксации суставов или следования заданной траектории [43].

PID-контроль: сочетание с другими методами управления позволяет улучшить координацию взаимодействия человека и машины. Часто применяется в роботах для тренировки походки, таких как ARMin III и RUPERT IV [44].

3. Системы управления на основе сигналов

Эти системы используют сигналы электроэнцефалограммы (ЭЭГ) или электромиограммы (ЭМГ) для считывания намерений пациента и управления устройствами. Примеры включают:

Использование ЭМГ-сигналов для распознавания моделей движений, что повышает точность управления протезами или экзоскелетами [45-46].

Применение технологий интеграции многомерных сигналов (например, углов суставов, давления на подошву) для повышения точности распознавания намерений и точности управления [47-49].

Несмотря на значительные достижения в улучшении характеристик реабилитационных роботов, остаются проблемы, такие как адаптивность моделей, стабильность сигналов и их обработка в режиме реального времени [50,51]. В будущих исследованиях необходимо оптимизировать алгоритмы и технологии обработки сигналов для предоставления более интеллектуальных, безопасных и эффективных услуг реабилитационного тренинга.

1.3 Основные проблемы текущих исследований

В настоящее время реабилитационные роботы с разгрузочной поддержкой уже нашли определённое применение в реабилитационных центрах и больницах многих стран, а также постепенно начинают использоваться в домашних условиях. Однако реабилитационные роботы, основанные на управлении с использованием поверхностных электромиографических (ЭМГ) сигналов, всё ещё находятся на стадии исследований и не применяются в широкомасштабной клинической практике. Анализ текущих систем реабилитационных роботов с разгрузочной поддержкой выявил следующие проблемы:

1. Высокая стоимость:

Дизайн, производство и тестирование реабилитационных роботов с разгрузочной поддержкой предполагают использование высокоточных многомерных датчиков силы, передовых композитных материалов, сложных систем управления и оборудования для сбора ЭМГ-сигналов, что значительно увеличивает затраты [52]. Высокая стоимость делает эти устройства недоступными для многих медицинских учреждений и индивидуальных пользователей, ограничивая их распространение и доступность для пациентов.

2. Недостаток персонализированной адаптации:

Роботы с разгрузочной поддержкой должны адаптироваться к пользователям разного возраста, пола, физического состояния и этапов реабилитации [53]. Текущие конструкции и алгоритмы часто не могут полностью учитывать эти различия, что приводит к снижению эффективности реабилитации для отдельных групп пользователей.

3. Проблемы безопасности и комфорта:

Движущиеся механические части могут представлять опасность для пациента, особенно при отсутствии достаточного контроля или в случае неисправности [54]. Обеспечение безопасности робота при любых условиях эксплуатации остаётся серьёзной задачей. Кроме того, длительное использование или ношение робота может вызывать дискомфорт, особенно в местах контакта с телом, что может снижать желание пользователя продолжать реабилитацию и замедлять процесс восстановления.

4. Проблемы задержки управления из-за скорости сбора сигналов и вычислений:

Задержки в управлении, вызванные скоростью сбора сигналов и их обработки, являются распространённой проблемой в управлении роботами. Это проявляется в замедленной реакции и снижении точности. Во-первых, данные, собранные датчиками, требуют обработки (например, фильтрации или преобразования). Если скорость обработки низкая или алгоритмы слишком сложные, система управления может не успевать своевременно реагировать, что вызывает задержки [55]. Во-вторых, сложные алгоритмы управления (например, адаптивное управление или управление с прогнозированием модели) требуют больших вычислительных ресурсов. При недостаточной производительности оборудования или ограниченных вычислительных возможностях команды управления могут выводиться с опозданием, что снижает точность и стабильность управления движениями робота. Кроме того, задержки могут возникать на этапе передачи сигналов, увеличивая общую задержку. Эти проблемы приводят к тому, что робот становится менее отзывчивым в быстро меняющейся среде, что снижает точность и стабильность управления электродвигателями.

В рамках данного исследования:

Используются одноосевые датчики силы, а корпус робота выполнен из алюминиевого сплава, что значительно снижает стоимость устройства, делая его более доступным для использования в домашних условиях.

Предусмотрены регулируемые механизмы ширины пояса и высоты подъёма, что позволяет адаптировать устройство к пользователям разного возраста и пола.

Для повышения безопасности встроена функция мониторинга скорости снижения пояса, которая позволяет определить падение пользователя. Для пользователей с серьёзными нарушениями двигательных функций нижних конечностей предусмотрен ручной режим управления, предотвращающий возникновение аварийных ситуаций.

Для решения проблемы задержки данных используется цифровой передатчик с скоростью передачи 11500 бод, а верхнеуровневая система управления работает на промышленном компьютере с процессором Intel i7, что обеспечивает высокую скорость вычислений.

1.4 Основное исследовательское содержание данной диссертации.

Роботы для реабилитации нижних конечностей являются одной из актуальных тем исследований как в области робототехники, так и в медицинской сфере. В данной работе изучены предпосылки и актуальность этой темы, проанализировано текущее состояние исследований в области реабилитационных роботов для нижних конечностей, выявлены существующие трудности и предложены решения этих проблем.

Во-первых, на основе структуры костей и движений нижних конечностей человека был разработан робот для реабилитации, способствующий восстановлению функции ходьбы. Конструкция робота позволила снизить затраты на производство за счёт использования

механических регулировочных механизмов, которые повышают уровень персонализации устройства. Проведён эксперимент по пространственному движению поясницы человека с использованием специального оборудования для динамического захвата движения, что подтвердило соответствие конструкции робота требованиям использования.

С помощью метода Денавіта-Хартенберга (D-H) проведён кинематический анализ, а методом Лагранжа выполнен динамический анализ. В управлении роботом реализованы адмитансный контроль и PID-регулятор, обеспечивающие динамическое следование движениям соединительных частей робота и пациента в реальном времени. Для реализации движения вперёд или поворотов использовались теория нечеткого управления и логика определения намерений движения.

Во-вторых, проведён анализ прочности конструкции реабилитационного робота, включая статические и динамические состояния, а также предусмотрены меры безопасности, такие как устройства экстренной остановки, что повышает безопасность реабилитационных тренировок.

Наконец, эксперименты с ходьбой человека подтвердили соответствие робота требованиям по плавности и комфорту его следования за движениями пациента.

Выводы по первой главе

1. Реабилитационные роботы для нижних конечностей, объединяющие механические, электронные и управленческие технологии, предоставляют помощь в движении и реабилитационные тренировки пациентам с

ограниченными двигательными функциями. Эти устройства помогают решить проблему нехватки реабилитационных терапевтов, улучшая качество жизни пациентов и являясь важным направлением в области медицинской робототехники.

2. Развитые страны начали исследования в области реабилитационных роботов значительно раньше, достигнув множества технологических прорывов. Китай, несмотря на более поздний старт, в последние годы демонстрирует быстрый прогресс и частичное внедрение продукции в клиническую практику.

3. Передовые реабилитационные роботы уже применяются в клинической практике. Например, швейцарский робот Lokomat улучшает функции нижних конечностей пациента через тренировки походки, а израильский экзоскелет ReWalk помогает пациентам в восстановлении ходьбы, значительно повышая эффективность реабилитации.

4. Системы управления роботами в основном основаны на моделях, физических параметрах и сигналах. Контроль, основанный на физических параметрах, в настоящее время считается наиболее надёжной и широко применяемой системой управления.

5. Реабилитационные роботы сталкиваются с рядом проблем, таких как высокая стоимость, недостаточная персонализация, вопросы безопасности и комфорта, а также задержки управления. Высокая стоимость обусловлена использованием высокоточных датчиков и сложных систем. Текущие конструкции недостаточно адаптированы к разнообразию пользователей.

Глава 2 Проектирование и анализ роботизированной системы реабилитации нижних конечностей

2.1 Анализ движений нижних конечностей человека

Система реабилитационного робота для нижних конечностей представляет собой автоматизированное оборудование и систему, разработанную специально для пациентов с нарушениями функций нижних конечностей. Её основной принцип основан на технологиях разгрузочной поддержки и динамического сопровождения. Система включает в себя три основных компонента: динамическую систему разгрузочной поддержки, платформу для ходьбы и систему поддержки с демпфированием боковых колебаний.

Дизайн динамической системы разгрузочной поддержки является ключевым элементом, подобным «сердцу» всей системы. Он напрямую влияет на эффективность реализации реабилитационных функций и конечные результаты восстановления.

В данной главе сначала проводится углублённый анализ механизмов движения нижних конечностей человека, что позволяет точно определить степени свободы и диапазоны движений для динамической разгрузочной поддержки. На основе этих данных тщательно разработан её приводной модуль.

Кроме того, с учётом требований к управлению выполнены комплексные работы по разработке программного и аппаратного обеспечения системы контроля и датчиков.

Этот последовательный и системный процесс проектирования позволил успешно завершить разработку полной системы реабилитационного робота для нижних конечностей.

2.1.1 Биомеханические параметры движений нижних конечностей человека

В стандартной стоячей позе человека, как показано на рисунке 2.1, анатомически определены основные плоскости и оси тела [56]. Основные плоскости тела включают: сагиттальную (срединную) плоскость, фронтальную (корональную) плоскость и горизонтальную (трансверсальную) плоскость. Эти три плоскости взаимно перпендикулярны и делят тело на симметричные правую и левую части, переднюю и заднюю части, а также верхнюю и нижнюю части.

Основные оси тела включают: продольную ось, перпендикулярную к сагиттальной плоскости; поперечную ось, перпендикулярную к фронтальной плоскости; и вертикальную ось, перпендикулярную к горизонтальной плоскости. Эти три оси также взаимно перпендикулярны и пересекаются в центре тяжести тела.

Все движения нижних конечностей человека можно описать как движения в пределах основных плоскостей и вращения вокруг основных осей.

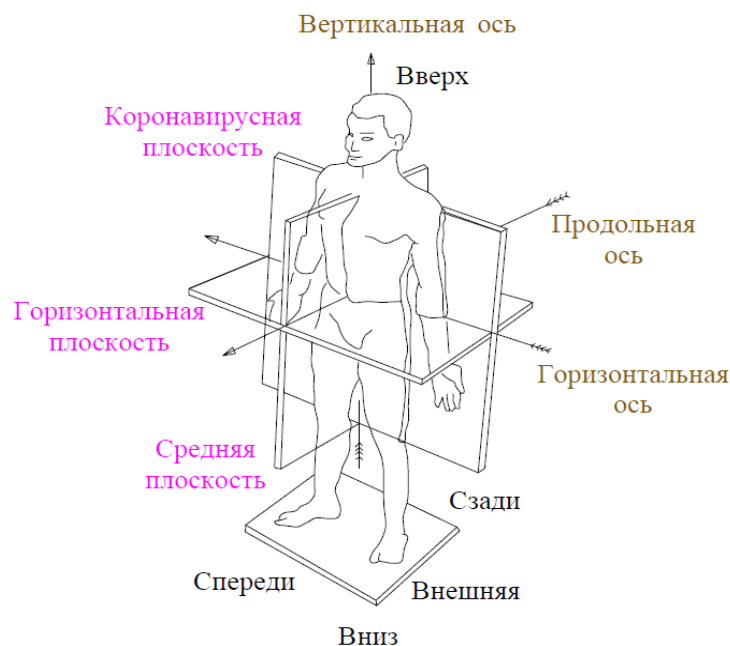


Рисунок 2.1: Основные плоскости тела человека

Наиболее важные суставы нижних конечностей включают тазобедренный, коленный и голеностопный суставы [57]. Их структура и формы движения являются основными ориентирами при разработке антропоморфного дизайна экзоскелетов.

Поскольку в данной работе разгрузочная поддержка осуществляется на уровне поясницы, движения поясницы соответствуют траектории движения тазобедренного сустава. Поэтому ниже подробно рассматриваются структура и механизмы движения тазобедренного сустава человека.

2.1.2 Анализ траектории пространственного движения поясницы

Как показано на рисунке 2.2, тазобедренный сустав расположен в верхней части бедра, соединяя туловище и нижние конечности. Он выполняет функции опоры и поддержания равновесия во время движения. По своей структуре это трёхосный шарнирный сустав, состоящий из

вертлужной впадины тазовой кости, головки бедренной кости и других вспомогательных структур [58,59].

Вертлужная впадина охватывает 2/3 головки бедренной кости, что аналогично полусферическому шарниру в механике. Тазобедренный сустав имеет три степени свободы: сгибание/разгибание в сагиттальной плоскости вокруг продольной оси; приведение/отведение во фронтальной плоскости вокруг поперечной оси; внутреннюю/наружную ротацию в горизонтальной плоскости вокруг вертикальной оси. Во время ходьбы сгибание/разгибание является основным движением, приведение/отведение поддерживает баланс тела, а внутренняя/наружная ротация изменяет направление движения.

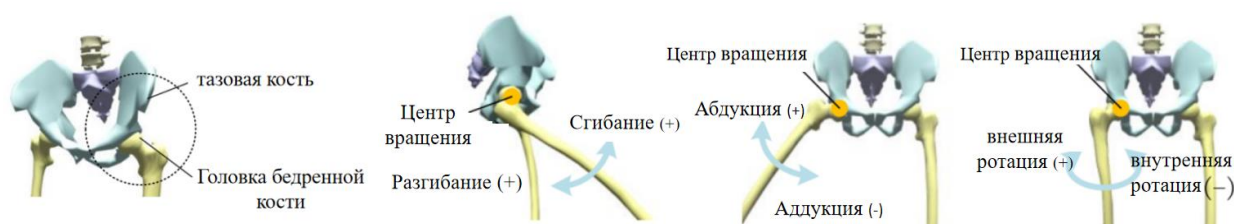


Рисунок 2.2: Структура и механизмы движения тазобедренного сустава

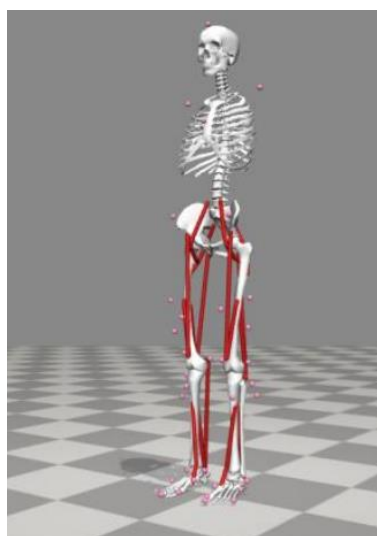
2.1.3 Моделирование и симуляция в OpenSim

OpenSim — это бесплатное программное обеспечение с открытым исходным кодом, предназначенное для разработки, анализа и визуализации систем опорно-двигательного аппарата. Оно широко используется для анализа динамики ходьбы, изучения двигательных функций, разработки медицинских устройств и в других областях. В настоящее время сотни биомеханических лабораторий по всему миру используют его мощные функции для исследований в области движения. Пользователи OpenSim создали активное сообщество разработчиков, которое постоянно

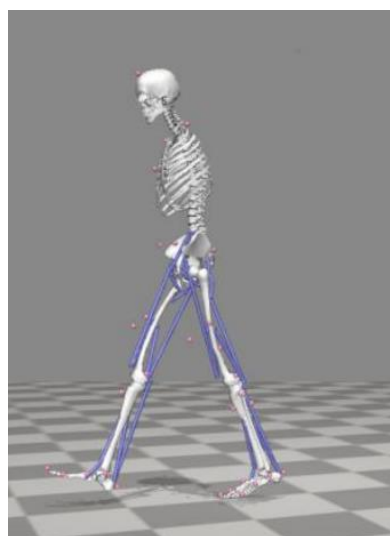
совершенствует функционал программы, что значительно упрощает доступ к ресурсам [60,61].

Для выполнения симуляции анализа походки человека с помощью OpenSim необходимо создать высокоточный кинематический мышечно-скелетный модель нижних конечностей человека. Эта модель состоит из суставов, костей и упрощённой модели мышц, соответствующих структуре человеческого тела. Программное обеспечение может симулировать движения суставов, приводимых в действие мышечной силой.

Биомеханическая лаборатория Стэнфордского университета уже создала полную мышечно-скелетную модель нижних конечностей человека [62,63]. Поэтому можно использовать готовую модель, которая включена в установочный пакет OpenSim, как показано на рисунке 2.3 (а). После открытия модели необходимо загрузить файлы, требуемые для кинематического и динамического анализа, и провести симуляцию анализа походки, как показано на рисунке 2.3 (b).



(а) Мышечно-скелетная модель нижних конечностей



(b) Симуляция ходьбы

Рисунок 2.3: Моделирование и симуляция человека в OpenSim

На основе графика угла тазобедренного сустава, сгенерированного в OpenSim, как показано на рисунке 2.4, было определено, что в нормальной походке угол поворота тазобедренного сустава человека находится в диапазоне от -19° до 26° . Данные об угле тазобедренного сустава, полученные из симуляции OpenSim для нормальной походки, будут использоваться в качестве теоретической основы для планирования траектории помощи в ходьбе для экзоскелета поясницы [64].

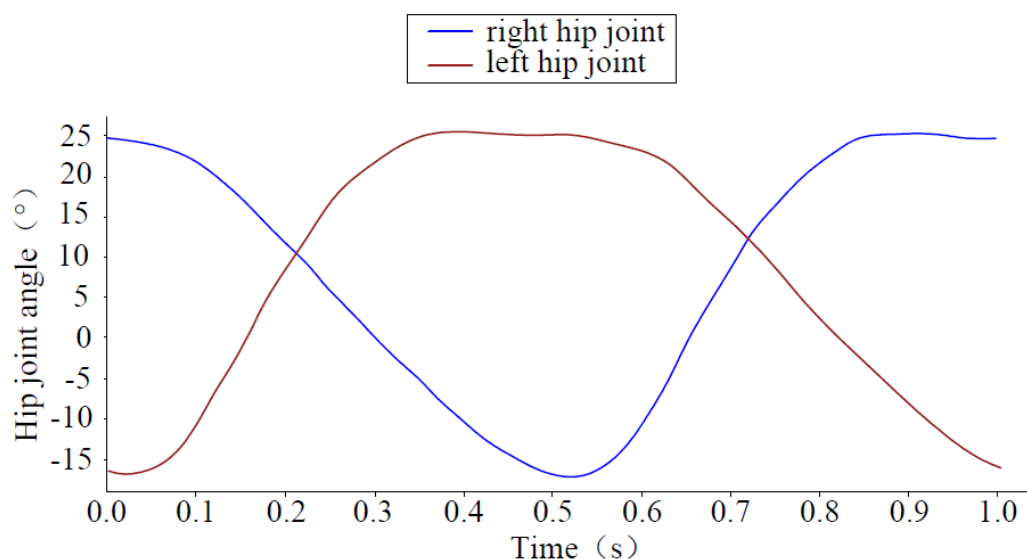


Рисунок 2.4: График изменения угла тазобедренного сустава при ходьбе человека

2.2 Систематическая конструкция робота для реабилитации нижних конечностей

Общая цель проектирования роботизированного устройства для реабилитации нижних конечностей определяется на основе функциональных требований и особенностей ношения. Устройство должно одновременно предоставлять помощь в выполнении определённых реабилитационных

движений и обеспечивать безопасность и комфорт пациента в процессе использования. При разработке конструкции следует придерживаться следующих принципов проектирования [65,66]:

Безопасность: Робот для реабилитации нижних конечностей непосредственно воздействует на пациентов с нарушенной или утраченной двигательной функцией нижних конечностей. Его основная задача — помочь пациентам выполнять повседневные действия и проходить реабилитацию. Чтобы избежать вторичных травм пациента из-за возможных неисправностей экзоскелета, конструкция каждого сустава должна быть оснащена механическими ограничителями для обеспечения безопасности пациента.

Совместимость: Конструкция реабилитационного робота должна основываться на принципах эргономики и быть ориентированной на человека. В частности, движения тазобедренного сустава устройства должны соответствовать форме, диапазону движения и степени свободы тазобедренного сустава пациента, обеспечивая совместимость движений.

Универсальность: Размеры основных частей реабилитационного робота должны регулироваться в определённых пределах, чтобы устройство подходило для пользователей с различным телосложением. Это повысит универсальность экзоскелета и снизит затраты на его использование.

2.2.1 Проектирование механической конструкции

Данная часть посвящена пациентам с недостаточной мышечной силой. У таких пациентов мышечная сила недостаточна для поддержания их тела во время длительных стоячих реабилитационных тренировок. Согласно

теории реабилитации, для таких пациентов эффективным методом тренировки является разгрузочная реабилитация. На рисунке 2.5 представлен разработанный робот для реабилитации нижних конечностей, включающий следующие модули: модуль ходовой платформы, модуль подъёма, модуль бокового перемещения и модуль двустороннего противоположного движения. Общая масса устройства составляет 180 кг.

Устройство для разгрузочной ходьбы представляет собой передовую технологию реабилитации, разработанную для помощи пациентам с инсультом и повреждениями нервной системы. Оно уменьшает нагрузку на нижние конечности пациента во время ходьбы, что позволяет безопасно и эффективно проводить реабилитационные тренировки.

Данная технология использует регулируемую систему разгрузочной поддержки, которая снижает давление на суставы и мышцы нижних конечностей во время лечения, способствуя восстановлению двигательной функции. Пациентам обычно требуется надевать специально разработанный разгрузочный жилет или пояс для таза, которые соединяются с системой поддержки. Это обеспечивает стабильную и точную разгрузочную поддержку тела пациента в процессе тренировки ходьбы [67,68].

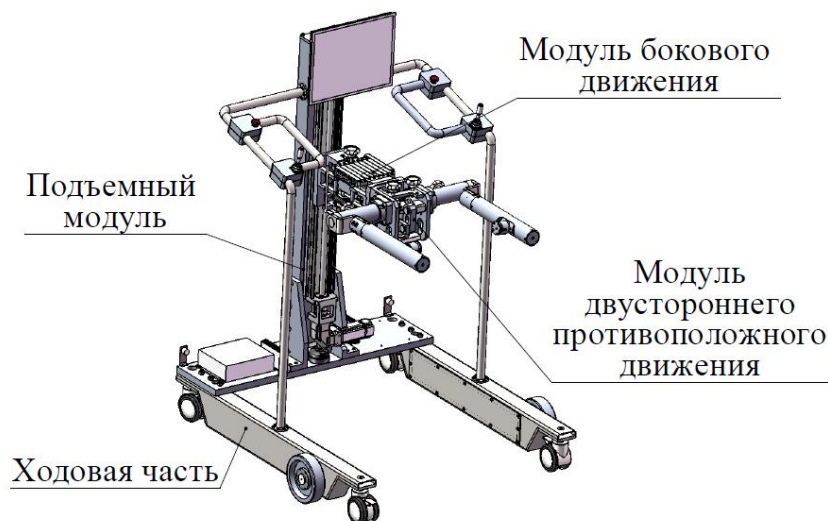


Рисунок 2.5. Структурная схема робота для реабилитационных тренировок

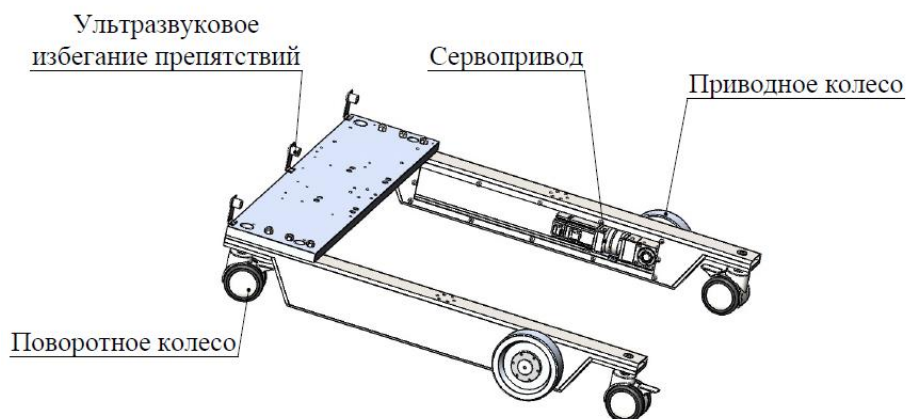


Рисунок 2.6. Модуль ходовой платформы

Модуль ходовой платформы является базовым компонентом реабилитационного робота. По обеим сторонам ходовой платформы установлены два сервопривода. Управление скоростью вращения этих сервоприводов позволяет осуществлять прямолинейное движение вперед или повороты.

Большая часть веса ходовой платформы распределяется на четыре шариковых колеса, в то время как ведущие колёса отвечают за движение модуля. На передней части модуля ходовой платформы установлены три

ультразвуковых датчика расстояния, которые предназначены для обнаружения людей или препятствий перед роботом во время его движения.

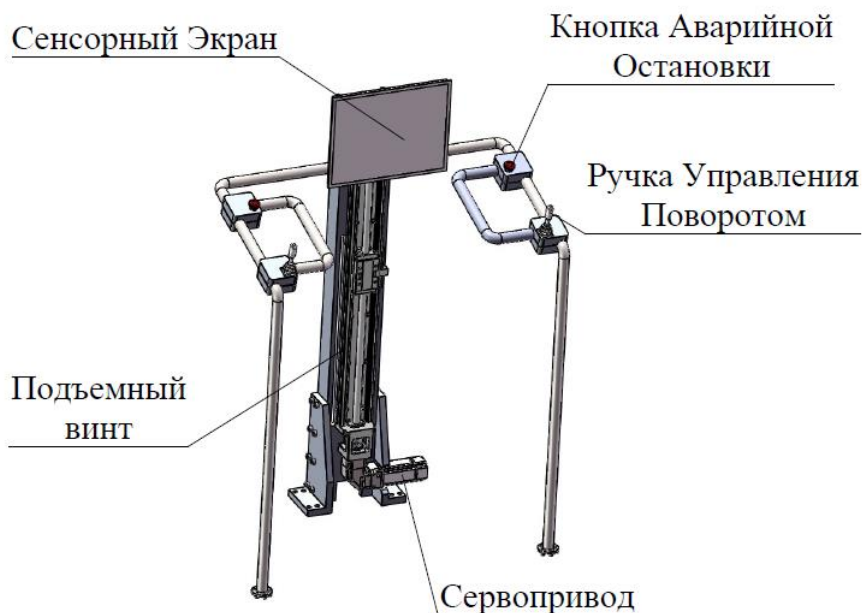


Рисунок 2.7. Модуль подъёма и поручни

Модуль подъёма осуществляет вертикальное движение с помощью шариково-винтовой передачи, приводимой в действие сервоприводом. На верхней части модуля закреплён сенсорный экран, который облегчает управление во время реабилитационного процесса.

На поручнях с обеих сторон установлены рычаги управления поворотом и кнопки аварийной остановки. Пациенты могут выбрать автоматический режим поворота, при котором робот для реабилитации поворачивает в соответствии с разницей давления на поясничной области с обеих сторон.

В случаях, когда пациент не может стоять прямо и его верхняя часть тела наклонена в одну сторону, необходимо использовать режим управления с помощью рычагов. В этом режиме пациент управляет движением вперёд или поворотах робота, используя рычаги.

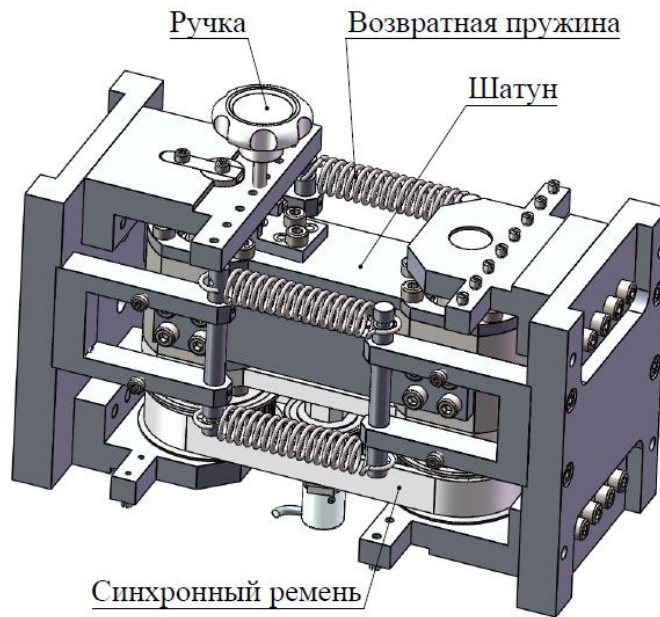


Рисунок 2.8. Модуль бокового перемещения

Модуль бокового перемещения разработан для адаптации к периодическим боковым движениям тазобедренного сустава пациента при ходьбе. Передняя ось фиксирована и неподвижна, соединяется с задней осью с помощью системы рычагов. Между передней и задней осями установлена система ремённой передачи.

Между соединительными пластинами передней и задней оси установлены пружины, которые обеспечивают демпфирование и позволяют задней соединительной пластине перемещаться влево и вправо.

Если пациенту необходимо зафиксировать модуль бокового перемещения, это можно сделать с помощью запорного механизма на рычаге. В этом случае модуль не будет выполнять боковые движения. Эта функция особенно важна для пациентов, которые при стоянии наклоняются в одну сторону, их центр тяжести смещён, и они не могут стоять прямо.

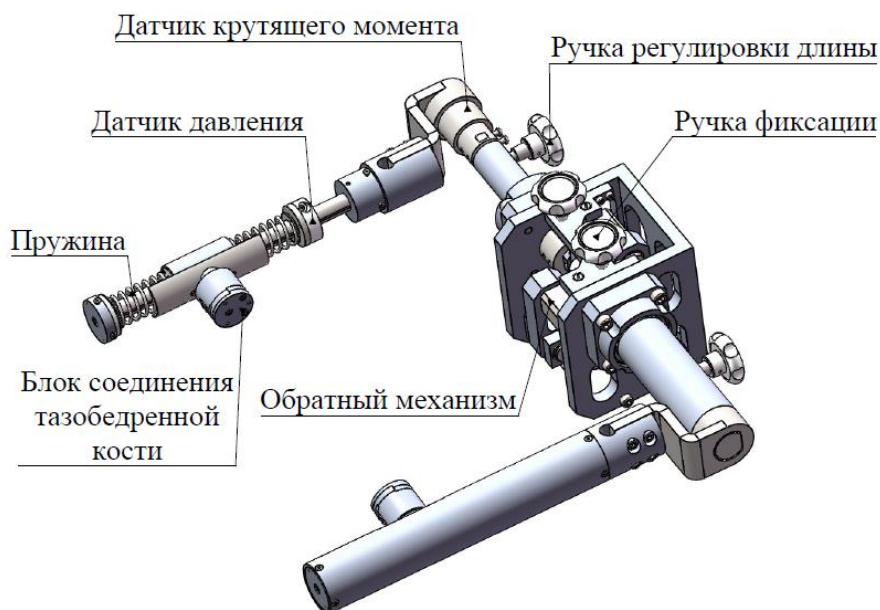


Рисунок 2.9. Модуль двустороннего противоположного движения

Модуль двустороннего противоположного движения соединяется с ремнями через защёлкивающийся механизм. Структуры с обеих сторон связаны между собой через центральный обратный механизм, который обеспечивает противоположные движения, адаптируясь к изменениям высоты тазобедренных суставов во время ходьбы.

На передней части обеих сторон модуля установлены датчики крутящего момента, которые фиксируют силу подъёма тазобедренных суставов пациента. Кроме того, датчики давления регистрируют изменения давления на тазобедренные суставы в продольном направлении.

Длина конструкции регулируется с помощью специальной рукоятки, которая изменяет ширину между двумя сторонами. При необходимости модуль можно зафиксировать с помощью запорной рукоятки, что предотвращает противоположные движения. Эта функция предназначена для пациентов, которые во время стояния наклоняются в одну сторону.

2.2.2 Проектирование системы управления

Как показано на рисунке 2.10, структура системы управления роботом для реабилитации нижних конечностей состоит из нескольких компонентов. Система использует три различных подхода к передаче данных и управлению, которые противопоставляются друг другу, чтобы предотвратить потерю данных или взаимное влияние в процессе передачи.

Передача данных от всех сервоприводов осуществляется через шину CAN, которая подключена к CAN-карте. Далее CAN-карта передаёт данные в основную управляющую систему через USB-порт. Серводрайверы сервоприводов поддерживают три режима управления: режим позиционирования, режим управления скоростью и режим управления моментом.

Левые и правые рукоятки передают данные в основную управляющую систему через шину 485. Основная управляющая система выполнена на базе промышленного компьютера (ПК), который оснащён встроенным портом связи по протоколу 485.

Датчики давления и крутящего момента с обеих сторон подключаются к промышленному компьютеру через цифровые преобразователи, которые передают данные через USB-порт.

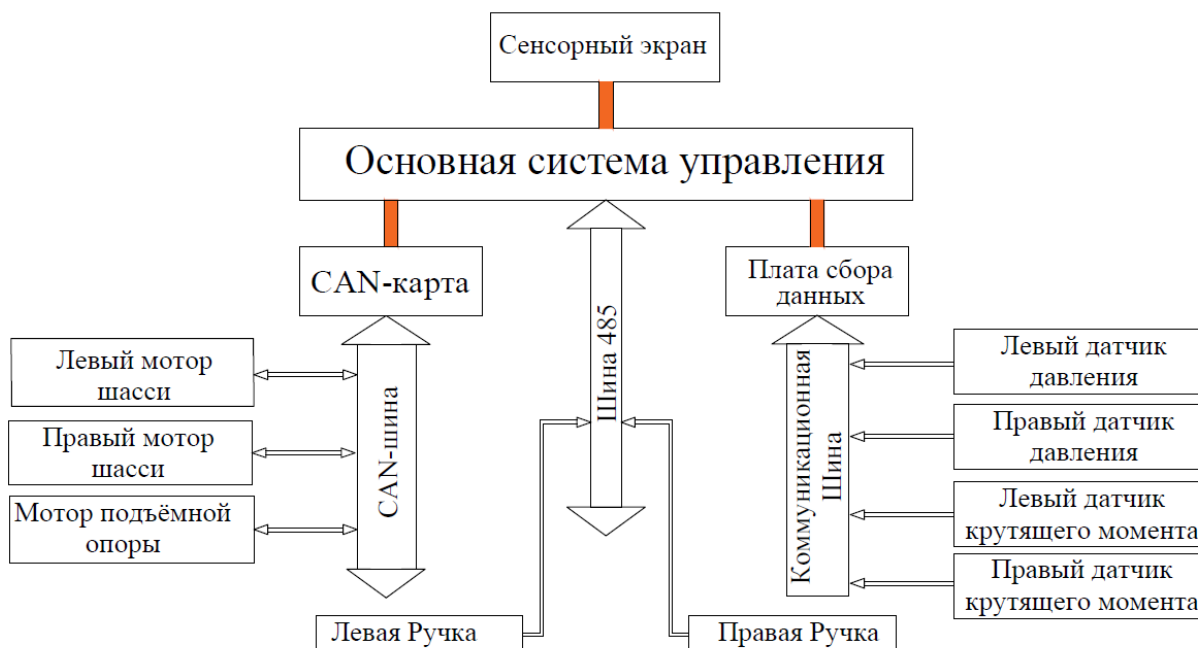


Рисунок 2.10. Структура системы управления

2.2.3 Выбор электрических компонентов

Основная управляющая система, с точки зрения экономичности, использует промышленный компьютер модели QY-B5700, произведённый компанией Zhengzhou Qiyang Industrial Control Technology Co., Ltd. (Чжэнчжоу, Китай).



Рисунок 2.11. Промышленный компьютер

Таблица 2.1. Параметры промышленного компьютера QY-B5700

Наименование параметра	Значение параметра
Процессор	Intel Core i7-10610U
Интерфейс	8×USB, 6×RS232или4×RS232/2×422/485

Операционная система	WIN10
I/O порты	HDMI/VGA
Питание	9-36VDC
Потребляемая мощность	15W
Управляющее программное обеспечение	C++

Для привода ходовой платформы и модуля подъёма используются сервоприводы, произведённые компанией Shanghai STEP Electric Corporation (Китай).

Приводной двигатель ходовой платформы: модель SMC60S-0020-30AAK-5DKH. С номинальной скоростью 3000 об/мин он требует установки планетарного редуктора с передаточным числом 1:30 для уменьшения скорости.

Приводной двигатель модуля подъёма: модель SMC60S-0040-30ABK-5DKH. При той же номинальной скорости 3000 об/мин для снижения скорости используется угловой редуктор с передаточным числом 1:5, который соединён с модулем шарикового винта.

Модуль шарикового винта имеет шаг винта 20 мм. Модель модуля шарикового винта: KK10020P-980A1-F0S3.

Таблица 2.2. Параметры электродвигателей

Название	SMC60S-0020-30AAK-5DKH	SMC60S-0040-30ABK-5DKH
Номинальное напряжение	48VDC	48VDC
Номинальный	0.64Nm	1.27Nm

крутящий момент		
Вес	0.9Kg	1.6Kg
Момент инерции	$1.4 \times 10^{-4} \text{Kg/m}^2$	$2.6 \times 10^{-4} \text{Kg/m}^2$
Номинальная скорость вращения	3000rpm	3000rpm
Мощность	200W	400W
Привод	FD124S-LB-000	

Для измерения взаимодействия пациента в сагиттальной плоскости на двух выдвижных рычагах опоры для таза установлены четыре независимых датчика давления. Каждый датчик давления соединён с ползуном через пружину, образуя простую систему для измерения относительных сил между пациентом и устройством.

Когда пациент пытается двигаться, его относительное положение по отношению к роботу изменяется. Поясной ремень, который носит пациент, тянет ползунок, вызывая сжатие или растяжение пружины. Согласно закону Гука, упругая деформация пружины создаёт силу, которая измеряется датчиком давления. Параметры датчиков давления приведены в Таблице 2.3.

Для измерения взаимодействия пациента в вертикальной плоскости на передних концах двух выдвижных рычагов опоры для таза установлены два датчика крутящего момента. Эти датчики предоставляют простой способ измерения относительных сил между пациентом и устройством в вертикальной плоскости.

Когда таз пациента находится выше уровня опоры для таза, пояс пациента создаёт восходящую силу на выдвижные рычаги. Эта сила

вызывает вращение рычагов против часовой стрелки, и возникающий крутящий момент регистрируется датчиком как положительный сигнал.

Наоборот, если таз пациента находится ниже уровня опоры для таза, пояс создаёт нисходящую силу на выдвижные рычаги. Эта сила вызывает вращение рычагов по часовой стрелке, и соответствующий крутящий момент фиксируется датчиком как отрицательный сигнал.

Параметры датчиков крутящего момента приведены в Таблице 2.3.

Таблица 2.3. Параметры датчиков давления и крутящего момента

Тип	датчик давления	датчик крутящего момента
Название	LZ-HM49	LZ-NJD-500NM
Напряжение	5~12VDC	5~12VDC
Диапазон измерений	40kg	500N.m
Частота отклика	60HZ	60HZ
Повторяемость	$\leq \pm 0.5\%F.S$	$\leq \pm 0.5\%F.S$
Чувствительность	1.0~1.5mV/V	0.8-1.0mV/V
Безопасная перегрузка	120%F • S	120%F • S

Для обеспечения высокой точности и стабильности сбора данных аналоговые сигналы от двух датчиков крутящего момента и двух датчиков давления обрабатываются и преобразуются с помощью четырёхканального цифрового преобразователя давления.

Этот преобразователь способен преобразовывать аналоговые сигналы, поступающие от датчиков, в цифровую форму и передавать их на промышленный компьютер через протокол связи RS485 для дальнейшей обработки и мониторинга данных. Устройство оснащено встроенным

фильтрующим алгоритмом, что обеспечивает высокую стабильность и точность сигнала.

На следующем рисунке представлено изображение цифрового преобразователя давления и его параметры.



Рисунок 2.12. Цифровой преобразователь давления

Таблица 2.4. Таблица параметров цифрового передатчика

Наименование параметра	Значение параметра
Модель продукта	TDA-04D4
Название продукта	4-канальный цифровой передатчик
Общая точность	1/10000
Чувствительность	0.4mV/V–6mV/V
Напряжение возбуждения	5VDC
Интерфейс связи	RS485
Скорость передачи данных (TDA-04B)	1200、2400、4800、9600、19200、38400、57600、115200

2.3

Анализ методом конечных элементов для робота реабилитации нижних конечностей

В процессе реабилитационных тренировок робот для восстановления нижних конечностей несёт вес тела человека и, будучи консольной конструкцией, подвержен деформациям. Чтобы убедиться, что основная поддерживающая структура не будет подвергаться избыточным

деформациям или разрушению из-за нагрузки, в данной работе проводится проверка безопасности нагрузок с использованием стандартов безопасности медицинских реабилитационных устройств. Для этого выполняется статический и динамический анализ конструкции.

Метод конечных элементов (МКЭ) является одной из наиболее широко применяемых численных методик в инженерной практике. Благодаря своим уникальным преимуществам в расчётах, этот метод способствовал появлению целого ряда специализированного программного обеспечения для анализа методом конечных элементов [69].

Workbench — это программное обеспечение для интегрированного моделирования, разработанное компанией ANSYS, которое помогает пользователям решать проблемы гетерогенности, возникающие при использовании других САЕ-программ в процессе разработки продукта. Эта платформа моделирования может легко выполнять анализ методом конечных элементов (МКЭ) для сложных механических систем, включая статический и динамический анализ конструкций, кинематику жёстких тел и гидродинамический анализ [70].

По сравнению с классическим интерфейсом ANSYS, Workbench предлагает более интуитивно понятный и ориентированный на инженерные задачи подход, что делает его удобным даже для пользователей без опыта работы с программным обеспечением для анализа методом конечных элементов [70].

Для моделирования конструкции был использован SolidWorks, в котором была создана трёхмерная модель. После удаления элементов, не влияющих на общую прочность конструкции (таких как поручни), модель была импортирована в программное обеспечение ANSYS Workbench для анализа. Материалы конструкции были определены следующим образом: Валы и подшипники: сталь 45; Поддерживающие разгрузочные элементы: алюминиевый сплав 6061-T5; Остальные детали и крепёжные элементы: углеродистая сталь Q235. Таблица 2.4 представляет параметры обычной углеродистой стали.

Таблица 2.5. Параметры обычной углеродистой стали

Properties of Outline Row 3: Structural Steel			
	A	B	C
1	Property	Value	Unit
2	Material Field Variables	Table	
3	Density	7850	kg m ⁻³
4	Isotropic Secant Coefficient of Thermal Expansion		
5	Coefficient of Thermal Expansion	1.2E-05	C ⁻¹
6	Isotropic Elasticity		
7	Derive from	Young's Modulus and Poisson's Ratio	
8	Young's Modulus	2E+11	Pa
9	Poisson's Ratio	0.3	
10	Bulk Modulus	1.6667E+11	Pa
11	Shear Modulus	7.6923E+10	Pa
12	Strain-Life Parameters		
13	Display Curve Type	Strain-Life	
14	Strength Coefficient	9.2E+08	Pa
15	Strength Exponent	-0.106	
16	Ductility Coefficient	0.213	
17	Ductility Exponent	-0.47	
18	Cyclic Strength Coefficient	1E+09	Pa
19	Cyclic Strain Hardening Exponent	0.2	
20	S-N Curve	Tabular	
21	Interpolation	Log-Log	
22	Scale	1	
23	Offset	0	Pa
24	Tensile Yield Strength	2.5E+08	Pa
25	Compressive Yield Strength	2.5E+08	Pa
26	Tensile Ultimate Strength	4.6E+08	Pa
27	Compressive Ultimate Strength	0	Pa

При проведении анализа методом конечных элементов (МКЭ) для работа реабилитации нижних конечностей упрощение конструкции является ключевым шагом для повышения эффективности и надёжности результатов. Для упрощения процесса обработки деталей конструкция была разделена на отдельные элементы, соединённые болтами. Если анализировать полную

модель без упрощения, это приведёт к избыточным вычислениям, потребляющим значительные временные и ресурсные затраты.

Путём упрощения и оптимизации конструкции были удалены избыточные соединительные элементы, обшивка и другие компоненты, которые минимально влияют на прочность. Оставив только ключевые детали, удалось значительно снизить вычислительную сложность и повысить эффективность решения задачи [71–73].

Для сеточной разбивки в работе реабилитации нижних конечностей использовался автоматический метод разбиения сетки, с размером элементов сетки, установленным на 6 мм. Для областей с повышенной концентрацией напряжений, таких как штифты, была выполнена дополнительная детализация сетки. Качество сетки было оценено как хорошее. Модель после разбиения сетки показана на Рисунке 2.11.

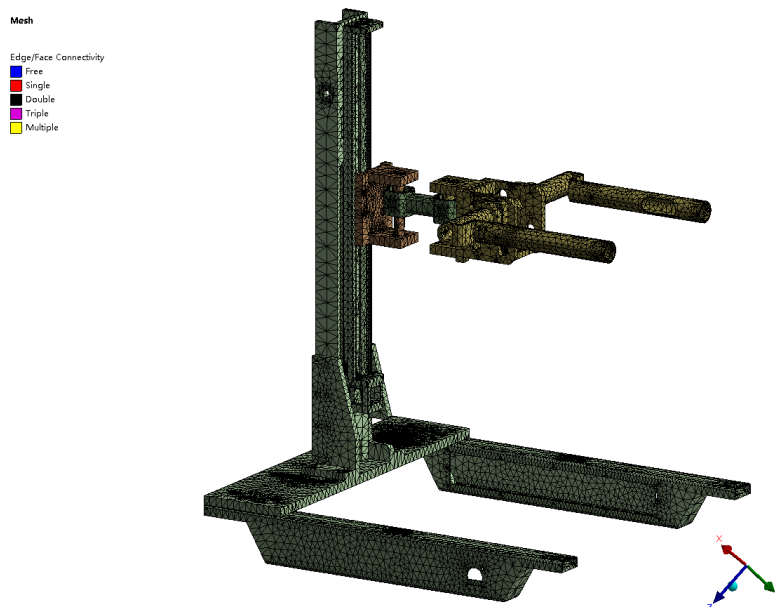
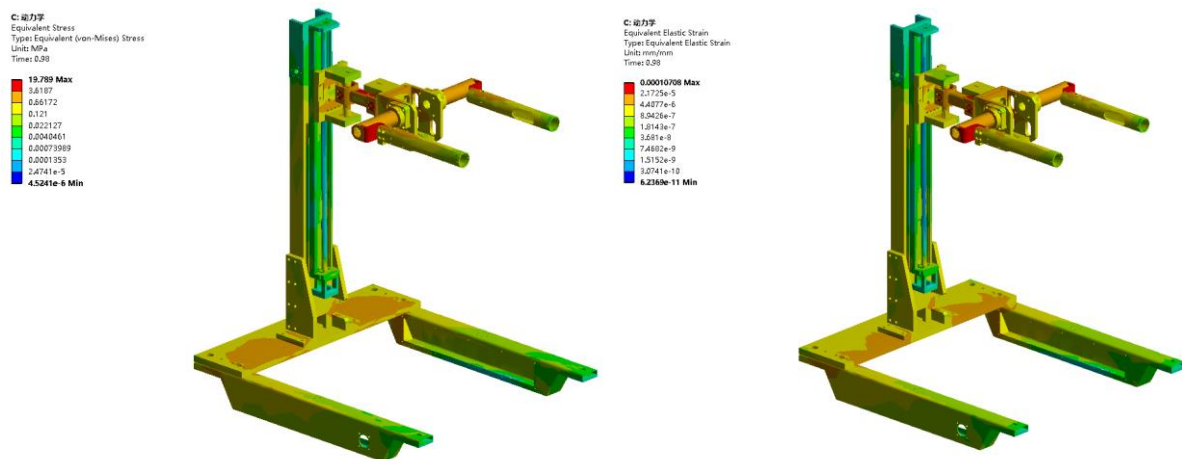


Рисунок 2.11. Разбиение сетки робота для реабилитации нижних конечностей

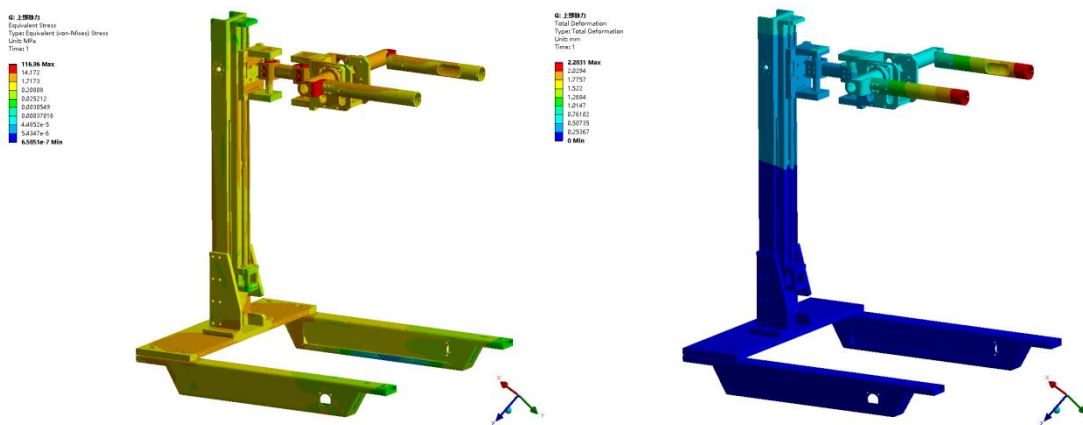
Робот для реабилитации нижних конечностей способен обеспечивать максимальную разгрузочную силу в 30 кг. Согласно китайским стандартам приёмки реабилитационного оборудования, устройство должно выдерживать статическую нагрузку, превышающую рабочую в 4 раза, то есть 120 кг, без появления трещин или повреждений.

Результаты расчётов при динамической нагрузке в 30 кг представлены на Рисунке 2.12. Максимальные значения напряжения и деформации наблюдаются в области, отмеченной красным цветом. Максимальное напряжение: 19.789 МПа. Максимальная деформация: 0.000107.



(a) Карта распределения напряжений (b) Карта распределения деформаций

Рисунок 2.12. Результаты динамического анализа



(a) Карта распределения напряжений (b) Карта распределения деформаций
Рисунок 2.13. Результаты статического анализа

При статической нагрузке в 120 кг результаты расчётов представлены на Рисунке 2.13. Максимальные значения напряжения и деформации наблюдаются в области, отмеченной красным цветом: Максимальное напряжение: 116.96 МПа, Максимальная деформация: 2.2831 мм, Предел текучести алюминиевого сплава 6061-T5 составляет 270 МПа, что свидетельствует о соответствии конструкции требованиям безопасности.

Выводы по второй главе

1. Описание физиологической структуры тазобедренного сустава, определение его степеней свободы и диапазона движений для создания теоретической основы проектирования робота для реабилитации нижних конечностей.

2. Построение мышечно-скелетной модели нижних конечностей с использованием OpenSim, выполнение симуляции анализа походки, извлечение данных о углах движения тазобедренного сустава для планирования траектории движения робота.

3. Робот состоит из ходовой платформы, модуля подъёма, модуля бокового перемещения и модуля двустороннего противоположного движения, что позволяет соответствовать трём степеням свободы движения тазобедренного сустава в пространстве.

4. Использование схемы управления с многошинной передачей данных, интеграция сервоприводов, датчиков и управляющего терминала для

обеспечения передачи данных в реальном времени и управления устройством.

5.Выбор высокопроизводительного промышленного компьютера, сервоприводов, а также датчиков давления и крутящего момента для обеспечения высокой точности управления и стабильности системы.

6.Проведение статического и динамического анализа конструкции робота с использованием ANSYS Workbench для проверки прочности и безопасности под нагрузкой.

Глава 3 Математическая модель и алгоритм управления роботом для реабилитации нижних конечностей

3.1 Алгоритм управления опорной частью шагающего робота для реабилитации нижних конечностей

В реабилитационных тренировках ходьбы пациенты часто не могут самостоятельно выдерживать весь вес своего тела, поэтому им требуется система поддержки веса для снижения нагрузки силы тяжести и уменьшения давления на организм[74-76]. Для повышения стабильности и естественности походки такая система должна не только эффективно уменьшать гравитационную нагрузку, но и точно направлять траекторию движения таза пациента, оптимизируя его походку. Исходя из этого, в данной работе предлагается динамическая модель системы поддержки веса, направленная на научную и техническую поддержку реабилитации ходьбы пациентов.

3.1.1 Кинематический анализ опорной части робота

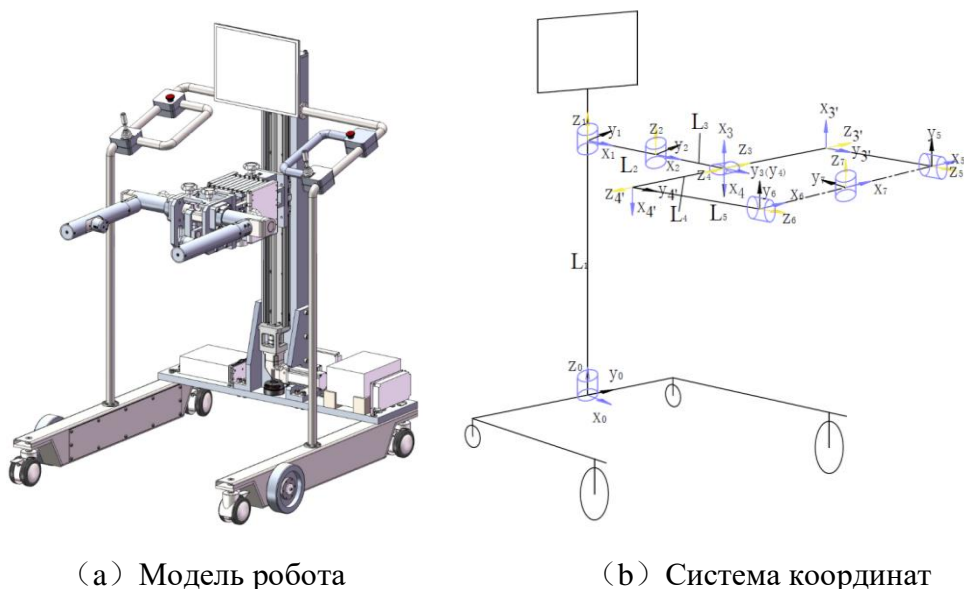
Так как пациенты часто не могут поддерживать весь свой вес самостоятельно, им нужна система поддержки веса, чтобы обеспечить им гравитационную разгрузку. Во время ходьбы система поддержки веса, помимо обеспечения гравитационной разгрузки, также должна направлять таз пациента. Исходя из этого, предлагается динамическая модель системы поддержки веса.

Робот для реабилитации нижних конечностей, рассматриваемый в данной работе, представляет собой типичный последовательный робот с открытой цепью. Его кинематический анализ направлен на изучение взаимосвязи между позицией и ориентацией таза пациента, параметрами звеньев и углами сочленений. Точное математическое моделирование и корректный анализ движения позволяют более точно оценить характеристики движения робота, его совместимость с движениями человека, а также создать теоретическую основу для управления движением робота.

В 1955 году Denavit и Hartenberg предложили универсальный стандартный метод D-H для кинематического анализа последовательных роботов [77]. Этот метод предусматривает создание локальных систем координат на каждом сочленении робота по унифицированным правилам. С помощью матрицы преобразования 4×4 , содержащей D-H параметры, описываются пространственные отношения между соседними звеньями. Этот подход значительно сокращает количество переменных в кинематических уравнениях и упрощает процесс решения, что позволяет удобно описывать позицию и ориентацию робота. Кинематическое моделирование методом D-H включает три этапа

[78,79]: Установление D-Н систем координат, Определение D-Н параметров, Решение матрицы преобразования D-Н координат.

На рисунке 3.1 представлен трёхмерный цифровой модель робота для реабилитации нижних конечностей. Положение таза (поясницы) пациента может изменяться в трёхмерном пространстве. Горизонтальные движения таза влево и вправо обеспечиваются демпфированием с помощью пружины, расположенной в передней части. Изменения высоты таза с обеих сторон достигаются благодаря механической конструкции, а подъём и опускание таза осуществляется с помощью винтовой передачи, приводимой в движение подъёмным электродвигателем. Для удобства расчётов было разработано 7 систем координат, как показано на рисунке 3.2.



(а) Модель робота

(б) Система координат

Рисунок 3.1 Модель робота для реабилитации нижних конечностей и система координат

Таблица 3.1 параметров D-Н робота для реабилитации нижних конечностей:

<i>Rodi-l</i>	<i>Rodi</i>	Variable θ_i	α_{i-1}	a_{i-1}	d_i
0	1	0	0°	0	L_1
1	2	0°	0°	L_2	0
2	3	-90°	-90°	L_3	0
3	3'	θ_3	0	0	L_4
3'	5	-90°	-90°	0	L_5
5	7	0°	-90°	L_4	0
2	4	-90°	90°	L_3	0
4	4'	θ_2	0	0	L_4
4'	6	90°	-90°	0	L_5
6	7	0°	-90°	L_4	0

Используя метод Д-Н, преобразование координат из системы координат {i} в систему координат {i-1} записывается следующим образом:

$${}^{i-1}_i T = Rot(x, \alpha_{i-1}) Trans(a_{i-1}, 0, 0) Rot(z, \theta_i) Trans(0, 0, d_i)$$

(3-1)

$${}^{i-1}_i T = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -d_i \sin \alpha_{i-1} \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(3-2)

Подставляя данные из таблицы 3.1 последовательно в формулу, можно получить:

$${}^0_1T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-3)$$

Между системой координат 1 и системой координат 2 соединены посредством параллельного четырехшарнирного механизма, реализующего перемещение по координатам X и Y. Угол параллельного четырехшарнирного механизма составляет θ_1 , и вращения не существует.

$${}^1_2T = \begin{bmatrix} 1 & 0 & 0 & L_2 \cos \theta_1 \\ 0 & 1 & 0 & L_2 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-4)$$

$${}^2_3T = \begin{bmatrix} 0 & 1 & 0 & L_3 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-5)$$

$${}^3_3'T = \begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 & 0 \\ -\sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-6)$$

$${}^5_3T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-7)$$

$${}^7_5T = \begin{bmatrix} 1 & 0 & 0 & L_4 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-8)$$

$${}^2_4T = \begin{bmatrix} 0 & 1 & 0 & L_3 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-9)$$

$${}^4_4T = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-10)$$

$${}^6_4T = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & L_5 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-11)$$

$${}^7_6T = \begin{bmatrix} 1 & 0 & 0 & L_4 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-12)$$

$${}^0_7T = {}^0_1T (\theta_1)_2^1 T (\theta_2)_3^2 T (\theta_3)_3^3 T (\theta_3)_3^5 T (\theta_5)_5^7 T (\theta_7) \quad (3-13)$$

$${}^0_7T = \begin{bmatrix} 0 & -L_5 \cos \theta_2 & -\sin \theta_2 & L_3 + L_2 \cos \theta_1 \\ 1 & 0 & 0 & 2 * L_4 + L_2 \sin \theta_1 \\ 0 & -L_5 \sin \theta_2 & \cos \theta_2 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.1.2 Динамический анализ опорной части робота

Методы анализа динамики системы манипулятора разнообразны. В данной системе для вывода динамических уравнений робота для реабилитации нижних конечностей выбирается уравнение Лагранжа. Основным принципом метода Лагранжа заключается в объединении энергетических уравнений системы с дифференцированием различных состояний системы и времени. Необходимо лишь проанализировать кинетическую и потенциальную энергии системы, не нужно учитывать другие

вопросы[80,81]. Во многих случаях он очень прост в применении, легко понятен и маловероятно приводит к ошибкам.

Для получения динамических уравнений системы сначала необходимо решить кинетическую и потенциальную энергии системы, затем построить функцию Лагранжа, а затем, взяв частные производные функции Лагранжа по системным переменным и времени соответственно, можно получить их[82].

$$L = K - P \quad (3-14)$$

$$\begin{cases} \tau_i = \frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} \\ F_i = \frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{r}_i} \right) - \frac{\partial L}{\partial r_i} \end{cases} \quad (3-15)$$

Здесь L — функция Лагранжа, K и P — соответственно кинетическая и потенциальная энергии системы, τ_i Сумма внешних моментов, создающих вращение, F_i Сумма внешних сил, создающих вращение, системная переменная.

1. Анализ кинетической энергии системы.

Создаем прямоугольную систему координат в пространстве от нижней части основания, обозначаем ее как x_0 , y_0 , z_0 . Берём центр каждой соединительной тяги как положение центра масс. Согласно созданной пространственной декартовой системе координат можно получить координаты центра масс каждой соединительной тяги. Полученные координаты центра масс каждой соединительной тяги дифференцируем, чтобы получить компоненты скорости этой

соединительной тяги по трем осям системы координат, обозначаем их как x_i , y_i , z_i , $i=1,2,\dots,6$, По формуле $v_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2}$ Можно вычислить скорость i -й соединительной тяги, а затем получить трансляционную кинетическую энергию i -й соединительной тяги. Добавив вращательную кинетическую энергию i -й соединительной тяги, можно получить общую кинетическую энергию i -й соединительной тяги по следующей формуле[83-85].

$$k_i = \frac{1}{2} m_i v_i^2 + \frac{1}{2} I_i \dot{\theta}_i^2 \quad (3-16)$$

Соединительная тяга 1: существует только движение вверх и вниз:

$$E_{K1} = \frac{1}{2} m_1 v_1^2$$

Соединительная тяга 2:

$$E_{K2} = \frac{1}{2} m_2 v_2^2 + \frac{1}{2} I_2 \dot{\theta}_1^2$$

Соединительная тяга 3:

$$E_{K3} = \frac{1}{2} m_3 v_3^2$$

Левая соединительная тяга 4:

$$E_{K41} = \frac{1}{2} I_{41} \dot{\theta}_2^2$$

Левая соединительная тяга 5:

$$E_{K51} = \frac{1}{2} m_5 v_{51}^2 + \frac{1}{2} I_5 \dot{\theta}_{51}^2$$

$$\theta_{51} = 90^\circ, \quad \dot{\theta}_{51} = 0$$

Правая соединительная тяга 4:

$$E_{K42} = \frac{1}{2} I_{42} \dot{\theta}_2^2$$

Правая соединительная тяга 5:

$$E_{K52} = \frac{1}{2} m_{52} v_{52}^2 + \frac{1}{2} I_{52} \dot{\theta}_{52}^2$$

$$\theta_{52} = 90^\circ, \quad \dot{\theta}_{52} = 0$$

Решение координат центра масс каждой соединительной тяги:

Соединительная тяга 1:

$$\begin{cases} x_1 = 0 \\ y_1 = 0 \\ z_1 = L_1 \end{cases}$$

Соединительная тяга 2:

$$\begin{cases} x_2 = \frac{1}{2} L_2 C_1 \\ y_2 = \frac{1}{2} L_2 S_1 \\ z_2 = L_1 \end{cases}$$

Соединительная тяга 3:

$$\begin{cases} x_3 = L_2 C_1 + \frac{1}{2} L_3 \\ y_3 = L_2 S_1 \\ z_3 = L_1 \end{cases}$$

Левая соединительная тяга 4:

$$\begin{cases} x_{41} = L_3 + L_2 C_1 \\ y_{41} = -\frac{1}{2} L_4 + L_2 S_1 \\ z_{41} = L_1 \end{cases}$$

Левая соединительная тяга 5:

$$\begin{cases} x_{51} = L_3 + L_2 C_1 + \frac{1}{2} L_5 C_2 \\ y_{51} = -L_4 + L_2 S_1 \\ z_{51} = L_1 + \frac{1}{2} L_5 S_2 \end{cases}$$

Правая соединительная тяга 4:

$$\begin{cases} x_{42} = L_3 + L_2 C_1 \\ y_{42} = \frac{1}{2} L_4 + L_2 S_1 \\ z_{42} = L_1 \end{cases}$$

Правая соединительная тяга 5:

$$\begin{cases} x_{52} = L_3 + L_2 C_1 + \frac{1}{2} L_5 C_2 \\ y_{52} = L_4 + L_2 S_1 \\ z_{52} = L_1 - \frac{1}{2} L_5 S_2 \end{cases}$$

Теперь найдем скорость центра масс каждой соединительной тяги. Дифференцируя координаты центра масс каждой соединительной тяги, можно получить компоненты скорости центра масс каждой соединительной тяги в трех направлениях системы координат:

Соединительная тяга 1:

$$\begin{cases} \dot{x}_1 = 0 \\ \dot{y}_1 = 0 \\ \dot{z}_1 = \dot{L}_1 \end{cases}$$

Соединительная тяга 2:

$$\begin{cases} \dot{x}_2 = -\frac{1}{2}L_2S_1\dot{\theta}_1 \\ \dot{y}_2 = \frac{1}{2}L_2C_1\dot{\theta}_1 \\ \dot{z}_2 = \dot{L}_1 \end{cases}$$

Соединительная тяга 3:

$$\begin{cases} \dot{x}_3 = -\frac{1}{2}L_2S_1\dot{\theta}_1 \\ \dot{y}_3 = L_2C_1\dot{\theta}_1 \\ \dot{z}_3 = \dot{L}_1 \end{cases}$$

Левая соединительная тяга 4:

$$\begin{cases} \dot{x}_{41} = -L_2S_1\dot{\theta}_1 \\ \dot{y}_{41} = L_2C_1\dot{\theta}_1 \\ \dot{z}_{41} = \dot{L}_1 \end{cases}$$

Левая соединительная тяга 5:

$$\begin{cases} \dot{x}_{51} = -L_2S_1\dot{\theta}_1 - \frac{1}{2}L_5S_2\dot{\theta}_2 \\ \dot{y}_{51} = L_2C_1\dot{\theta}_1 \\ \dot{z}_{51} = \dot{L}_1 + \frac{1}{2}L_5C_2\dot{\theta}_2 \end{cases}$$

Правая соединительная тяга 4:

$$\begin{cases} \dot{x}_{42} = -L_2S_1\dot{\theta}_1 \\ \dot{y}_{42} = L_2C_1\dot{\theta}_1 \\ \dot{z}_{42} = \dot{L}_1 \end{cases}$$

Правая соединительная тяга 5:

$$\begin{cases} \dot{x}_{52} = -L_2S_1\dot{\theta}_1 - \frac{1}{2}L_5S_2\dot{\theta}_2 \\ \dot{y}_{52} = L_2C_1\dot{\theta}_1 \\ \dot{z}_{52} = \dot{L}_1 - \frac{1}{2}L_5C_2\dot{\theta}_2 \end{cases}$$

Согласно вышеупомянутым найденным скоростям центра масс, теперь можно получить конкретные выражения кинетической энергии каждой соединительной тяги:

Кинетическая энергия соединительной тяги 1:

$$E_{K1} = \frac{1}{2} m_1 v_1^2 = \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2) = \frac{1}{2} m_1 \dot{L}_1^2$$

Кинетическая энергия соединительной тяги 2:

$$E_{K2} = \frac{1}{2} m_2 v_2^2 + \frac{1}{2} I_2 \dot{\theta}_1^2 = \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2) + \frac{1}{2} I_2 \dot{\theta}_1^2 = \frac{1}{2} m_2 (\dot{L}_1^2 + \frac{1}{4} L_2^2 \dot{\theta}_1^2) + \frac{1}{2} I_2 \dot{\theta}_1^2$$

Кинетическая энергия соединительной тяги 3:

$$E_{K3} = \frac{1}{2} m_3 v_3^2 = \frac{1}{2} m_3 (\dot{x}_3^2 + \dot{y}_3^2 + \dot{z}_3^2) = \frac{1}{2} m_3 (\frac{1}{4} L_2^2 S_1^2 \dot{\theta}_1^2 + L_2^2 C_1^2 \dot{\theta}_1^2 + \dot{L}_1^2)$$

Кинетическая энергия левой соединительной тяги 4:

$$E_{K41} = \frac{1}{2} I_{41} \dot{\theta}_2^2 + \frac{1}{2} m_{41} v_{41}^2 = \frac{1}{2} I_{41} \dot{\theta}_2^2 + \frac{1}{2} m_{41} (L_2^2 \dot{\theta}_1^2 + \dot{L}_1^2)$$

Кинетическая энергия левой соединительной тяги 5:

$$\begin{aligned} E_{K51} &= \frac{1}{2} m_{51} v_{51}^2 = \frac{1}{2} m_{51} (\dot{x}_{51}^2 + \dot{y}_{51}^2 + \dot{z}_{51}^2) \\ &= \frac{1}{2} m_{51} (\dot{L}_1^2 + L_2^2 \dot{\theta}_1^2 C_1^2 + \frac{1}{4} L_5^2 \dot{\theta}_2^2 + L_2^2 \dot{\theta}_1^2 S_1^2 + L_5 \dot{L}_1 \dot{\theta}_2 C_2 + L_2 L_5 \dot{\theta}_1 \dot{\theta}_2 S_1 S_2) \end{aligned}$$

Кинетическая энергия правой соединительной тяги 4:

$$E_{K42} = \frac{1}{2} I_{42} \dot{\theta}_2^2 + \frac{1}{2} m_{42} v_{42}^2 = \frac{1}{2} I_{42} \dot{\theta}_2^2 + \frac{1}{2} m_{42} (L_2^2 \dot{\theta}_1^2 + \dot{L}_1^2)$$

Кинетическая энергия правой соединительной тяги 5:

$$\begin{aligned} E_{K52} &= \frac{1}{2} m_{52} v_{52}^2 = \frac{1}{2} m_{52} (\dot{x}_{52}^2 + \dot{y}_{52}^2 + \dot{z}_{52}^2) \\ &= \frac{1}{2} m_{52} (\dot{L}_1^2 + L_2^2 \dot{\theta}_1^2 C_1^2 + \frac{1}{4} L_5^2 \dot{\theta}_2^2 + L_2^2 \dot{\theta}_1^2 S_1^2 - L_5 \dot{L}_1 \dot{\theta}_2 C_2 + L_2 L_5 \dot{\theta}_1 \dot{\theta}_2 S_1 S_2) \end{aligned}$$

(2) Решение потенциальной энергии системы

Согласно координатам центра масс соединительных тяг, мы можем вычислить потенциальную энергию силы тяжести каждой соединительной тяги:

Соединительная тяга 1:

$$E_{p1} = m_1 g L_1$$

Соединительная тяга 2:

$$E_{p2} = m_2 g L_1$$

Соединительная тяга 3:

$$E_{p3} = m_3 g L_1$$

Левая соединительная тяга 4:

$$E_{p41} = m_{41} g L_1$$

Левая соединительная тяга 5:

$$E_{p51} = m_{51} g \left(L_1 + \frac{1}{2} L_5 S_2 \right)$$

Правая соединительная тяга 4:

$$E_{p42} = m_{42} g L_1$$

Правая соединительная тяга 5:

$$E_{p52} = m_{52} g \left(L_1 - \frac{1}{2} L_5 S_2 \right)$$

(3) Решение уравнений динамики системы

Согласно вычисленным кинетической и потенциальной энергиям каждой соединительной тяги теперь можно получить

подробную функцию Лагранжа для работа для реабилитации нижних конечностей:

$$L = \sum_{i=1}^5 E_{ki} - \sum_{i=1}^5 E_{pi} \quad (3-17)$$

$$\begin{aligned} \sum_{i=1}^5 E_{ki} &= \frac{1}{2} m_1 \dot{L}_1^2 + \frac{1}{2} m_2 (\dot{L}_1^2 + \frac{1}{4} L_2^2 \dot{\theta}_1^2) + \frac{1}{2} I_2 \dot{\theta}_1^2 + \frac{1}{2} m_3 (\frac{1}{4} L_2^2 S_1^2 \dot{\theta}_1^2 + L_2^2 C_1^2 \dot{\theta}_1^2 + \dot{L}_1^2) \\ &+ \frac{1}{2} I_{41} \dot{\theta}_2^2 + \frac{1}{2} m_{41} (L_2^2 \dot{\theta}_1^2 + \dot{L}_1^2) + \frac{1}{2} m_{51} (\dot{L}_1^2 + L_2^2 \dot{\theta}_1^2 C_1^2 + \frac{1}{4} L_5^2 \dot{\theta}_2^2 + L_2^2 \dot{\theta}_1^2 S_1^2 + L_5 \dot{L}_1 \dot{\theta}_2 C_2 + L_2 L_5 \dot{\theta}_1 \dot{\theta}_2 S_1 S_2) \\ &+ \frac{1}{2} I_{42} \dot{\theta}_2^2 + \frac{1}{2} m_{42} (L_2^2 \dot{\theta}_1^2 + \dot{L}_1^2) + \frac{1}{2} m_{52} (\dot{L}_1^2 + L_2^2 \dot{\theta}_1^2 C_1^2 + \frac{1}{4} L_5^2 \dot{\theta}_2^2 + L_2^2 \dot{\theta}_1^2 S_1^2 - L_5 \dot{L}_1 \dot{\theta}_2 C_2 + L_2 L_5 \dot{\theta}_1 \dot{\theta}_2 S_1 S_2) \\ \sum_{i=1}^5 E_{pi} &= m_1 g L_1 + m_2 g L_1 + m_3 g L_1 + m_{41} g L_1 + m_{51} g (L_1 + \frac{1}{2} L_5 S_2) + m_{42} g L_1 \\ &+ m_{52} g (L_1 - \frac{1}{2} L_5 S_2) \end{aligned}$$

Производная полученной функции Лагранжа по состоянию системы и времени позволяет получить уравнения динамики подъемно-опорной части. При решении первого уравнения динамики, Теперь взятие частной производной по \dot{L}_1 из функции Лагранжа дает:

$$\begin{aligned} \frac{\partial L}{\partial \dot{L}_1} &= m_1 \dot{L}_1 + m_2 \dot{L}_1 + m_3 \dot{L}_1 + m_{41} \dot{L}_1 + \frac{1}{2} m_{51} (2\dot{L}_1 + L_5 \dot{\theta}_2 C_2) \\ &+ m_{42} \dot{L}_1 + \frac{1}{2} m_{52} (2\dot{L}_1 - L_5 \dot{\theta}_2 C_2) \end{aligned}$$

Далее, взятие производной от вышеуказанного выражения по времени t дает:

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{L}_1} &= m_1 \ddot{L}_1 + m_2 \ddot{L}_1 + m_3 \ddot{L}_1 + m_{41} \ddot{L}_1 + \frac{1}{2} m_{51} (2\ddot{L}_1 + L_5 \ddot{\theta}_2 C_2 + L_5 \dot{\theta}_2 \dot{C}_2) \\ &+ m_{42} \ddot{L}_1 + \frac{1}{2} m_{52} (2\ddot{L}_1 - L_5 \ddot{\theta}_2 C_2 - L_5 \dot{\theta}_2 \dot{C}_2) \end{aligned}$$

$$\frac{\partial L}{\partial L_1} = -m_1g - m_2g - m_3g - m_{41}g - m_{51}g - m_{42}g - m_{52}g$$

Получено первое уравнение динамики:

$$F_1 = \frac{d}{dt} \frac{\partial L}{\partial \dot{L}_1} - \frac{\partial L}{\partial L_1} \quad (3-18)$$

Получено:

$$F_1 = (m_1 + m_2 + m_3 + 2m_{41} + 2m_{51})(\ddot{L}_1 + g) + F_p$$

Поскольку подъемный механизм использует электродвигатель для привода линейной направляющей системы, и шаг винта $s = 20$ мм, F_p - чистый вес пациента (от 0 до 30 кг).

Крутящий момент винта - это вращающий момент, приложенный к винту. Его формула расчета:

$$\tau = \frac{F \times S}{2\pi\eta} \quad (3-19)$$

Здесь τ - крутящий момент винта, F - сила тяги, приложенная к винту, S - шаг винта, η - коэффициент трения. Поэтому выходной крутящий момент подъемного электродвигателя:

$$\tau_1 = \frac{F_1 \times S}{2\pi\eta} + J_1 \ddot{\theta}_1 \quad (3-20)$$

Здесь J_1 - момент инерции винта, $\ddot{\theta}_1$ - угловая скорость вращения винта.

Динамическое уравнение робота для реабилитации нижних конечностей в стандартной форме записывается как:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = T \quad (3-21)$$

В которых $T = \tau_1$, τ_1 это приводные крутящие моменты подъемных двигателей, $D(q) \in R^{1 \times 1}$ общая масса подъемной части, являющаяся крутящим моментом, вызванным ускорением, $C(q, \dot{q}) \in R^{1 \times 1}$ сила Кориолиса и центробежная сила. В данном реабилитационном роботе подъемная часть движется вверх и вниз поступательно без вращения, поэтому данный член равен 0, $G(q) \in R^{1 \times 1}$ — сила тяжести.

3.2 Методы управления опорной частью робота

Согласно приведенному выше уравнению динамики, после завершения проектирования механической структуры реабилитационного робота можно определить массу экзоскелета, и уравнение динамики может быть упрощено до уравнения момента силы, связанного с ускорением. Построение динамической модели реабилитационного робота позволяет установить взаимосвязь между движением подъема и приводным моментом, что создаёт основу для управления движением подъемного модуля реабилитационного робота нижних конечностей.

Для обеспечения свободного движения таза пациента необходимо, чтобы соединение в области таза следовало за движением пациента. В данной работе используются два подхода для управления динамическим следованием подъемной части соединения между тазом пациента и роботом:

1.Метод, основанный на импедансном управлении (адмитанс) и PID-регулировании.

2.Метод, основанный на импедансном управлении (адмитанс) и скользящем управлении.

Результаты управления анализируются и сравниваются для выбора наиболее эффективного подхода.

3.2.1 Адмитанс и PD-контроль

1. Уравнение динамики

Динамическое уравнение объекта можно записать в следующей форме:

$$F_{motor} = M \cdot (g + a) + F_P + u \quad (3-22)$$

F_{motor} :приводная сила двигателя:

M :масса механической структуры подъемной части ,

$$M = m_1 + m_2 + m_3 + 2m_{41} + 2m_{51}$$

g :ускорение свободного падения

a :ускорение механической структуры подъемной части, $a = \ddot{L}_1$

F_P :разгрузочная сила, 0-30kg

u :управляющая сила, вычисляемая системой управления

Для каждого временного шага ускорение объекта a определяется следующим соотношением:

$$a = \frac{F_{motor} - F_P}{M} - g \quad (3-23)$$

Обновление скорости:

$$v(t + dt) = v(t) + a(t)dt \quad (3-24)$$

Обновление положения:

$$x(t + dt) = x(t) + v(t)dt \quad (3-25)$$

2. Управление адмиттансом – расчет желаемого положения:

Желаемое положение $x_d(t)$ вычисляется через адмиттанс $Y(s)$ на основе внешней силы $F_e(t)$, и их взаимосвязь выражается следующим образом:

$$x_d(t) = Y(s) \cdot F_e(t) \quad (3-27)$$

Адмиттанс $Y(s)$ определяется как соотношение между внешней силой и желаемым положением.

3. PID-регулятор:

Цель PID-регулятора — с помощью обратной связи скорректировать приводную силу двигателя таким образом, чтобы фактическое положение объекта максимально приблизилось к желаемому. PID-регулятор состоит из следующих трёх компонентов [86-87]:

Пропорциональная составляющая K_p : управление на основе текущей ошибки;

Интегральная составляющая K_i : управление на основе накопленной исторической ошибки;

Дифференциальная составляющая K_d : управление на основе скорости изменения ошибки.

Формула расчёта выхода PID-регулятора (управляющего воздействия u) выглядит следующим образом:

$$u(t) = K_p e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{d}{dt} e(t) \quad (3-26)$$

$e(t) = x_d(t) - x(t)$: Ошибка положения

$e(t)$: Это разница между желаемым положением и фактическим положением.

Интегральная и дифференциальная составляющие в дискретной форме вычисляются следующим образом:

Интегральная составляющая: $\int_0^t e(\tau) d\tau$ Приблизительно равно $e_{sum}(t) = \sum_{i=1}^t e(i) \cdot dt$

Дифференциальная составляющая: $\frac{d}{dt} e(t)$ Приблизительно равно $\frac{e(i) - e(i-1)}{dt}$

4. Расчёт приводной силы двигателя

В дискретном времени приводная сила вычисляется как:

$$F(i) = M \left(g + \frac{v(i) - v(i-1)}{dt} \right) + F_p + u(i) \quad (3-28)$$

5. Обновление скорости и положения:

$$v(i) = v(i-1) + \left(\frac{F_{motor}(i) - F_p}{M} - g \right) dt \quad (3-29)$$

$$x(i) = x(i-1) + v(i) dt \quad (3-30)$$

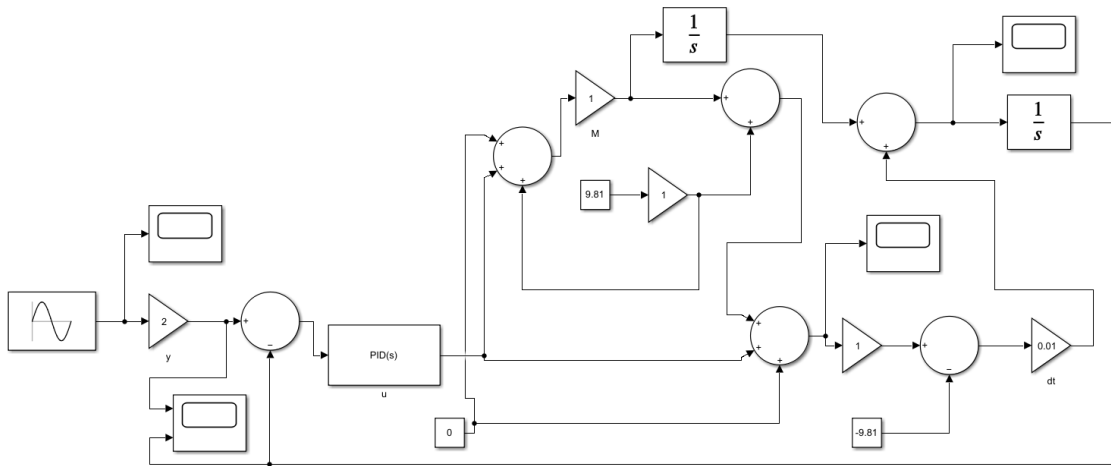


Рисунок 3.2 Блок-схема PID-контроллера в Simulink

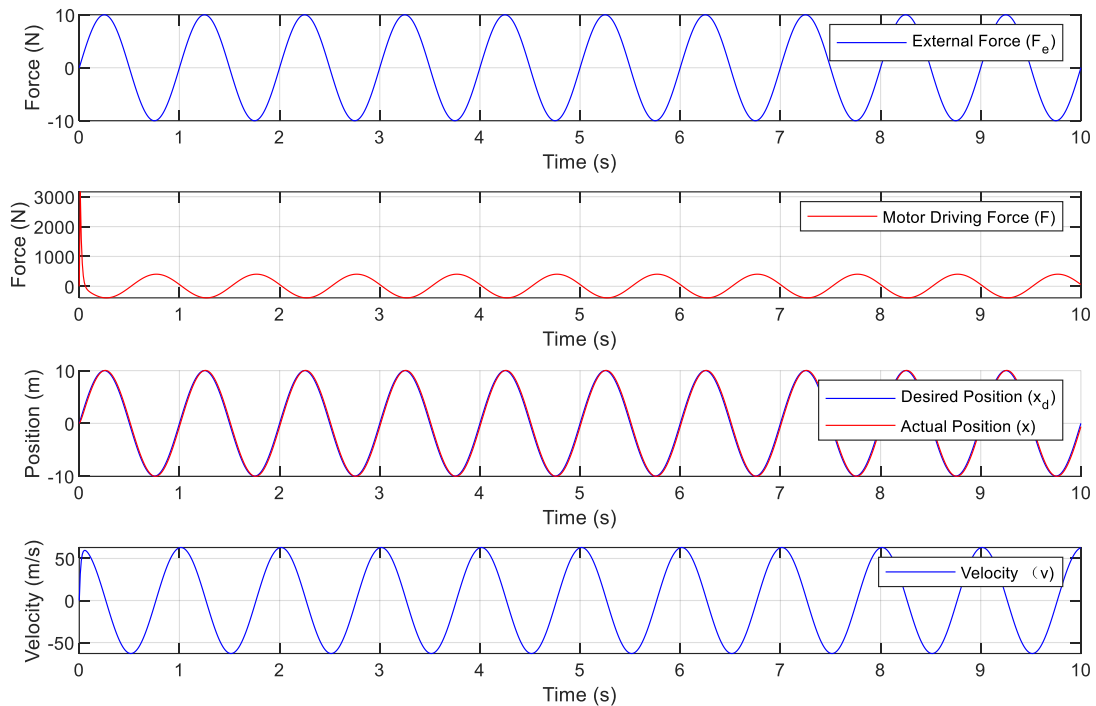


Рисунок 3.3. Результаты симуляции PID-контроллера

Как видно из приведённого выше рисунка, когда испытываемый прикладывает к соединению в области талии внешнюю силу, изменяющуюся по синусоидальному закону, система с помощью стратегии адмитанса достигает прямой пропорциональной

зависимости между внешней силой и желаемым положением. В процессе этого, с помощью точной настройки параметров PID-регулятора, удаётся точно контролировать выходную силу двигателя, что, в свою очередь, обеспечивает точное совпадение желаемого положения поддерживающей части с реальными изменениями высоты талии пациента, эффективно помогая в движении талии и улучшая пользовательский опыт.

3.2.2 Скользящее управление

1. Построение динамического уравнения системы

Заданное динамическое уравнение:

$$F_{motor} = M \cdot (g + a) + F_p \quad (3-31)$$

F_{motor} : Приводная сила двигателя

M : Масса механической конструкции подъемной части ,

$$M = m_1 + m_2 + m_3 + 2m_{41} + 2m_{51}$$

g : ускорение свободного падения

a : ускорение механической конструкции подъемной части ,

$$a = \ddot{L}_1$$

F_p : весоподъемность (не включая внешнюю силу F_e) , 0-30kg

Положение подъемной части x , скорость $v = \dot{x}$,
ускорение $a = \ddot{x}$.

2. Управление адмиттансом

Управление адмиттансом — это метод управления, который

реализуется путем установления зависимости между силой и движением. Ожидаемая зависимость между силой и положением выражается следующим образом: $F_d = k_e x_d$

Где F_d — это ожидаемая сила, k_e — пропорциональный коэффициент, x_d — ожидаемое положение. Поскольку ожидаемое положение пропорционально внешней силе, обозначим пропорциональный коэффициент как k_p , тогда $x_d = k_p F_e$.

3. Управление по скользящей поверхности

Определение скользящей поверхности: s

$$s = \dot{e} + \lambda e \quad (3-32)$$

Где $e = x_d - x$ ошибка положения, а λ — положительная константа, используемая для настройки динамических характеристик скользящей поверхности.

Взятие производной от скользящей поверхности:

$$\dot{s} = \ddot{x}_d - \dot{x} + \lambda \dot{e} \quad (3-33)$$

Согласно уравнению динамики: $F_{motor} = M \cdot (g + a) + F_p$, можно получить: \ddot{x} :

$$\ddot{x} = \frac{F - F_0 - Mg}{M} \quad (3-34)$$

Подставив \ddot{x} в выражение для \dot{s} и приравняв $\dot{s} = 0$, можно получить эквивалентный закон управления: F_{eq} :

$$F_{eq} = M(\ddot{x}_d + \lambda \dot{e}) + F_p + Mg \quad (3-35)$$

Чтобы система достигла скользящей поверхности за конечное время и оставалась на ней, вводится переключающий элемент управления F_{sw} :

$$F_{sw} = -k \operatorname{sgn}(s) \quad (3-36)$$

Где k — положительная константа, а $\operatorname{sgn}(s)$ — знак-функция.

Окончательный закон управления F_c выражается как:

$$F_c = F_{eq} + F_{sw} \quad (3-35)$$

4 Анализ устойчивости

Устойчивость управления по скользящей поверхности можно проанализировать с использованием теории устойчивости Ляпунова. Определим функцию Ляпунова $V = \frac{1}{2}s^2$, а затем возьмем её производную. Согласно правилу дифференцирования сложных функций: $\dot{V} = s\dot{s}$.

Взяв производную от $s = \dot{e} + \lambda e$, $\dot{s} = \ddot{e} + \lambda \dot{e}$, Согласно уравнению динамики: $F_{motor} = M \cdot (g + a) + F_p$, Можно

получить: $\ddot{x} = \frac{F - F_0 - Mg}{M}$, Одновременно $\ddot{e} = \ddot{x}_d - \ddot{x}$, Подставив

$\ddot{x} = \frac{F - F_0 - Mg}{M}$ в указанное выше уравнение, можно выразить:

$$\ddot{e} = \ddot{x}_d - \frac{F - F_0 - Mg}{M}, \quad \text{И} \quad \ddot{e} = \ddot{x}_d - \frac{F - F_0 - Mg}{M} \quad \text{Подставляя в}$$

$\dot{s} = \ddot{e} + \lambda \dot{e}$ выражается:

$$\dot{s} = \ddot{x}_d - \frac{F - F_0 - Mg}{M} + \lambda (\dot{x}_d - \dot{x}) \quad (3-36)$$

Закон управления: $F_c = F_{eq} + F_{sw}$, Где $F_{eq} = M(\ddot{x}_d + \lambda \dot{e}) + F_p + Mg$, $F_{sw} = -k \operatorname{sgn}(s)$, $k > 0$ является постоянной величиной , $\operatorname{sgn}(s)$ является знак-функцией , Подставляя $F_c = F_{eq} + F_{sw}$ в функцию \dot{s} ,выраженную через F , а затем подставляя в функцию Ляпунова $\dot{V} = s\dot{s}$:

$$\begin{aligned} \dot{V} &= s \left(\ddot{x}_d - \frac{M(\ddot{x}_d + \lambda \dot{e}) + F_p + Mg - F_p - Mg}{M} + \lambda (\dot{x}_d - \dot{x}) \right) - s \frac{k \operatorname{sgn}(s)}{M} \\ &= s(\ddot{x}_d - (\ddot{x}_d + \lambda \dot{e}) + \lambda (\dot{x}_d - \dot{x})) - s \frac{k \operatorname{sgn}(s)}{M} \\ &= -s \frac{k \operatorname{sgn}(s)}{M} \end{aligned}$$

Поскольку $s \frac{k \operatorname{sgn}(s)}{M} = \frac{k}{M} |s|$, Следовательно $\dot{V} = -\frac{k}{M} |s|$.

Потому что $V = \frac{1}{2} s^2$, Для любого $s \neq 0$, $V > 0$, Следовательно, V является положительно определённой функцией. Кроме того, поскольку $\dot{V} = -\frac{k}{M} |s|$, Для любого $s \neq 0$, $\dot{V} < 0$, Следовательно, \dot{V} является отрицательно определённой функцией. Согласно теореме устойчивости Ляпунова, равновесие системы является асимптотически устойчивым, то есть подъемная часть способна стабильно следовать изменениям внешней силы.

В соответствии с процессом управления нижним конечным реабилитационным роботом мы разработали блок-схему управления по скользящей поверхности в Simulink, как показано на рисунке 3.4,

и получили результаты моделирования, представленные на рисунке 3.5. Несмотря на то, что управление по скользящей поверхности позволяет эффективно реализовать адмиттанс-контроль в зависимости от внешней силы, прикладываемой испытуемым, и достигать отслеживания заданного положения, результаты моделирования приводной силы двигателя показывают, что при резком приложении внешней силы возникает значительное резкое изменение приводной силы двигателя. Это указывает на то, что, хотя управление по скользящей поверхности обеспечивает соответствие опорной части и поясничной области, иногда наблюдается эффект дрожания, что снижает комфорт испытуемого. Поэтому в процессе реального управления в данном реабилитационном роботе для нижних конечностей было принято решение использовать PID-контроль, чтобы обеспечить плавность работы и комфортное ощущение для испытуемого.

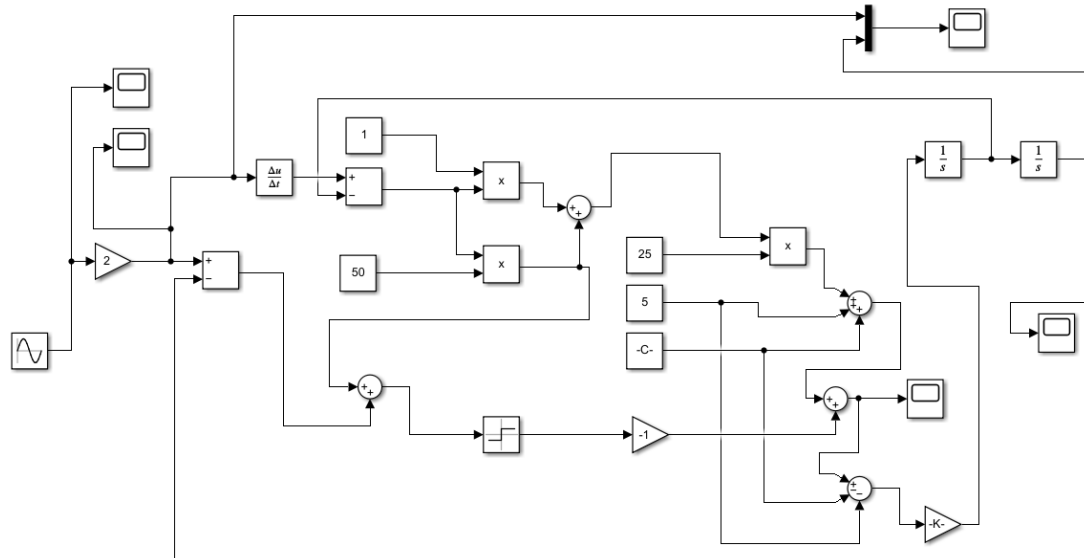


Рисунок 3.4. Блок-схема управления по скользящей поверхности в Simulink.

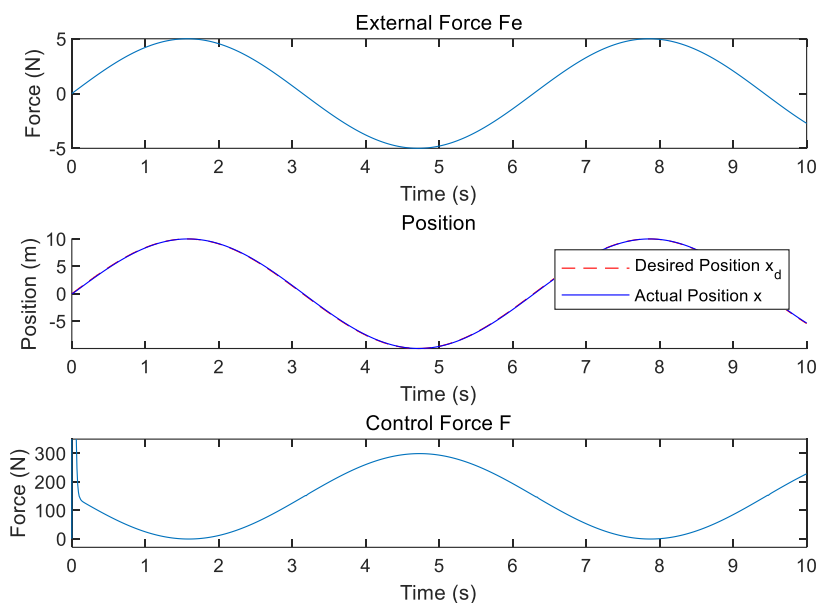


Рисунок 3.5. Результаты моделирования управления по скользящей поверхности.

3.3 Алгоритм управления шагающим шасси робота для реабилитации нижних конечностей

Пациенты с инсультом или спинальной травмой часто не могут ходить самостоятельно. Передвижение - это базовая способность нормальных людей для осуществления повседневной деятельности.

Поэтому, предоставление способности передвижения пациентам во время реабилитации является ключевым содержанием данного робота-помощника для реабилитации и ходьбы. Поэтому в этой главе на основе мобильного шасси робота для реабилитации, описанного во второй главе, предлагается алгоритм управления мобильным шасси на основе распознавания намерений.

3.3.1 Кинематический анализ шагающего шасси

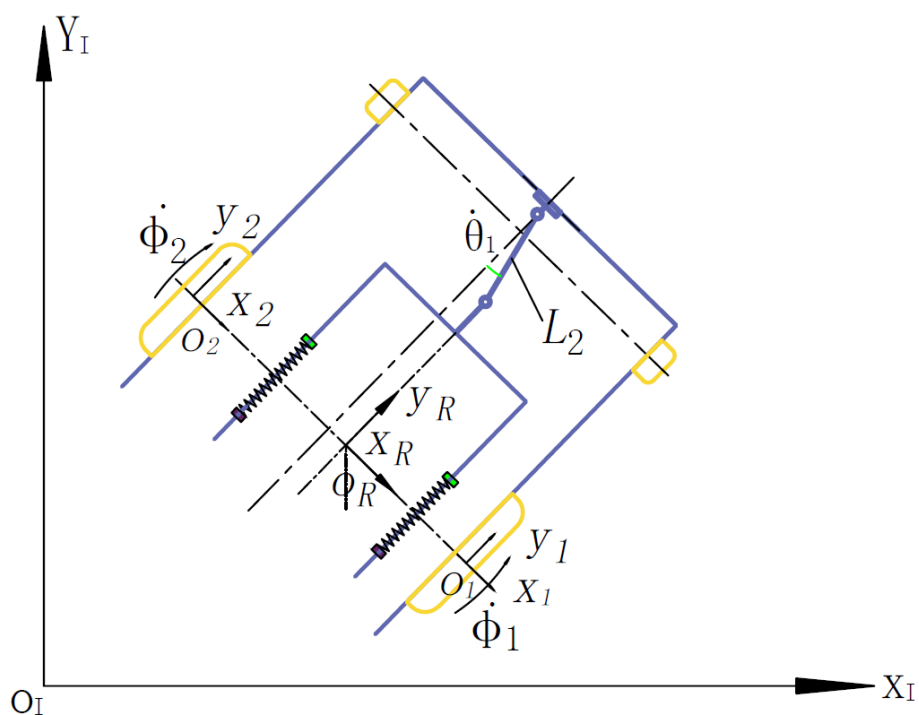


Рисунок 3.6 Упрощенная модель ходовой части робота для реабилитации нижних конечностей

Глобальная (мировая) система координат: поза робота выражается как:

$$\mathbf{X}_I = (x_I, y_I, \theta_I)^T \quad (3-37)$$

Система координат робота (локальная):

$$\mathbf{X}_R = (x_R, y_R, \theta_R)^T \quad (3-38)$$

3.3.2 Динамический анализ шагающего шасси

Соотношение скоростей между ними:

$$\dot{\mathbf{X}}_R = R(\theta)\dot{\mathbf{X}}_I \quad (3-39)$$

То есть

$$\begin{pmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{pmatrix} = R(\theta) \begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{pmatrix} \quad (3-40)$$

Поскольку робот для реабилитации нижних конечностей вращается вокруг оси Z , поэтому:

$$R(\theta) = \begin{pmatrix} \cos \theta_R & \sin \theta_R & 0 \\ -\sin \theta_R & \cos \theta_R & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3-41)$$

Поскольку у робота для реабилитации нижних конечностей два движущих колеса, а остальные два колеса являются опорными колесами, поэтому скорость перемещения пациента равна:

$$\dot{y}_R = \frac{1}{2} r \dot{\phi}_1 + \frac{1}{2} r \dot{\phi}_2 + L_2 \sin \theta_1 \dot{\theta}_R$$

$$\dot{x}_R = L_2 \dot{\theta}_1$$

$$\dot{\theta}_R = \frac{r \dot{\phi}_1}{2L_4} - \frac{r \dot{\phi}_2}{2L_4}$$

Запись в матричной форме может быть представлена как:

$$\begin{pmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & L_2 \\ \frac{r}{2} & L_2 \sin \theta_1 & \frac{r}{2} & 0 \\ \frac{r}{2L_4} & 0 & -\frac{r}{2L_4} & 0 \end{pmatrix} \begin{pmatrix} \dot{\phi}_1 \\ \dot{\theta}_R \\ \dot{\phi}_2 \\ \dot{\theta}_1 \end{pmatrix} \quad (3-42)$$

Отображение скорости робота в системе координат R на мировую систему координат XOY дает:

$$\begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{pmatrix} = R(\theta)^{-1} \begin{pmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{pmatrix} \quad (3-43)$$

$$\begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \sin \theta_1 (r\dot{\phi}_1 + r\dot{\phi}_2 - 2L_2 \cos \theta_1 + 2L_2 \dot{\theta}_R \sin \theta_1) \\ L_2 - L_2 \cos \theta_1^2 + \frac{1}{2} r(\dot{\phi}_1 + \dot{\phi}_2) \cos \theta_1 + \frac{1}{2} L_2 \sin 2\theta_1 \dot{\theta}_R \\ \dot{\theta}_R \end{pmatrix}$$

Необходимо вычислить соотношение между \dot{x}_I , \dot{y}_I , $\dot{\theta}_I$ в мировой системе координат x_I, y_I из-за изменения скорости вращения двух колес ($\dot{\phi}_1$, $\dot{\phi}_2$) и $\dot{\theta}_R$ в начале системы координат x_R, y_R :

То есть:

$$\begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{pmatrix} = J_{DP} \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\theta}_R \end{pmatrix} \quad (3-43)$$

Якобиан ходовой части составляет:

$$J_{DP} = \begin{pmatrix} \frac{-r \sin \theta_R (L_4 + L_2 \sin \theta_1)}{2L_4} & \frac{-r \sin \theta_R (L_4 - L_2 \sin \theta_1)}{2L_4} & L_2 \cos \theta_R \\ \frac{r \cos \theta_R (L_4 + L_2 \sin \theta_1)}{2L_4} & \frac{r \cos \theta_R (L_4 - L_2 \sin \theta_1)}{2L_4} & L_2 \sin \theta_R \\ \frac{r}{2L_4} & -\frac{r}{2L_4} & 0 \end{pmatrix} \quad (3-44)$$

3.3.3 Процесс нечеткого управления ходовой платформой

Как метод интеллектуального управления, не зависящий от точной математической модели, нечеткое управление может

обрабатывать неопределенность и сложные системы. Путем преобразования человеческого опыта и знаний в нечеткие правила нечеткое управление может реализовать интеллектуальное управление ходьбой робота, динамически регулировать скорость вращения колес в соответствии с различными входными силами и моментами для повышения производительности и адаптивности ходьбы робота.

С учетом фактической реабилитации и безопасности, проектируются шесть режимов движения, которые могут быть фактически реализованы пациентом: 1) прямо вперед; 2) повернуть влево при движении вперед; 3) повернуть вправо при движении вперед; 4) оставаться на месте.

Человеческий организм изменяет направление движения, скручивая тело, и получает скорость движения, толкая туловище. А таз, как ключевая часть, соединяющая ноги и туловище, играет роль соединения между верхним и нижним уровнями при ходьбе. Конкретно, при ходьбе человек делает таз скручиваться путем различия размера шагов левой и правой ног, тем самым приводя туловище к изменению направления. Человек толкает таз вперед путем шага, а таз, сохраняя баланс центра тяжести, приводит туловище в движение вперед. Поэтому сигналы взаимодействия силы F_1 и F_2 между роботом и тазом, полученные двумя

датчиками давления, могут использоваться для распознавания намерений движения пациента.

$$\begin{bmatrix} \Delta F \\ F \end{bmatrix} = \begin{bmatrix} F_1 - F_2 \\ F_1 + F_2 \end{bmatrix} \quad (3-45)$$

Здесь F_1 и F_2 - давления, собираемые правым и левым датчиками давления соответственно. Флюу-диаграмма для определения намерений пациента выглядит следующим образом:

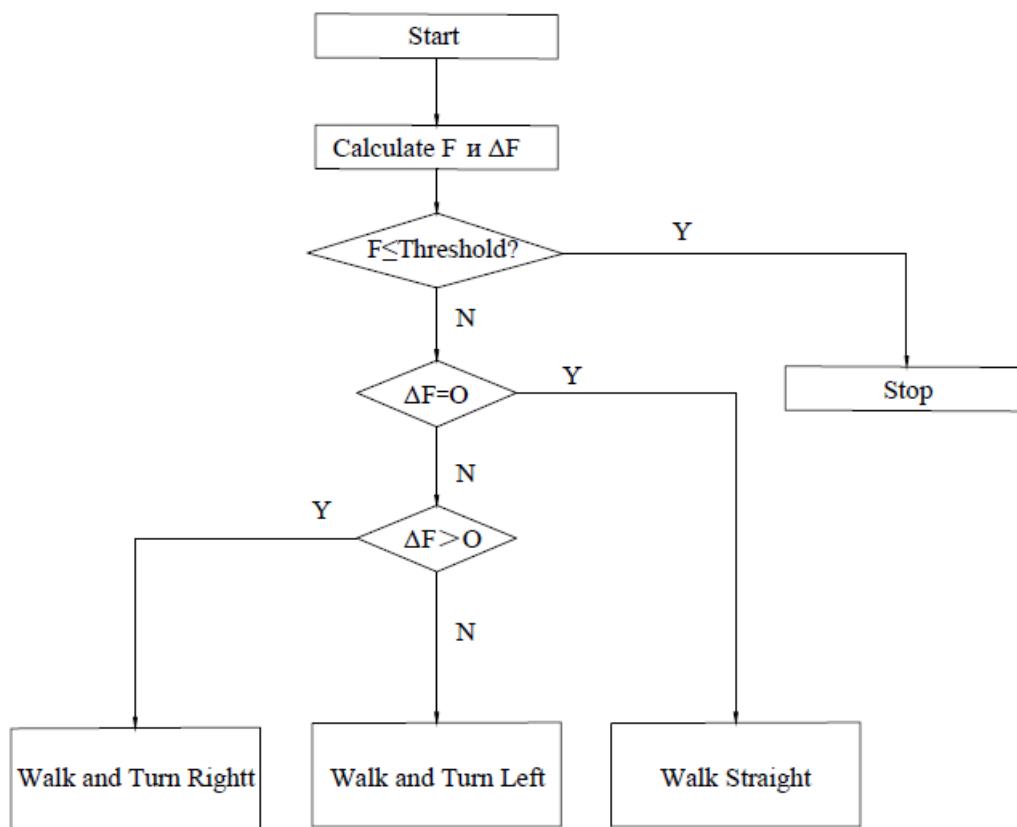


Рисунок 3.7. Определение намерения ходьбы

Нечеткая система управления в основном состоит из четырех частей: ① база знаний, ② нечеткий преобразователь, ③ механизм вывода, ④ дефаззификатор.

1. Область определения входных величин

Как правило, в нечетком контроллере необходимо предусмотреть коэффициент преобразования входных величин, который служит для масштабного преобразования входных величин и отображения их в заданную область определения входных величин. Поскольку максимальные значения F и ΔF пациента для данного нечеткого контроллера могут быть определены экспериментально.

2. Нечеткое преобразование

Функция нечеткого преобразования заключается в том, чтобы с помощью функции принадлежности отображать входные величины из вещественного пространства в нечеткое пространство, то есть переводить конкретные входные величины в нечеткие множества для последующего нечеткого вывода. Для обеспечения безопасности пациента в процессе реабилитационной тренировки движение назад запрещено, и измеряются только положительные значения силы. В данной статье для положительного направления силы установлено 12 нечетких множеств: $\{F_1, F_2, \dots, F_{12}\}$, которые последовательно увеличиваются. Для разности сил также установлено 12 нечетких множеств. Для удобства вычислений в статье используется треугольная модель в качестве функции принадлежности.

3. База знаний

База знаний состоит из базы данных и базы правил нечеткого управления. База данных используется для определения областей

определения каждого нечеткого множества входных и выходных величин, то есть областей определения нечетких подмножеств. База правил нечеткого управления является основной частью базы знаний и определяется опытным путем, в ней содержится информация о том, как выводить выходные действия на основе входных величин. Согласно опыту, когда намеренная сила F или разность сил ΔF становятся больше, согласно закону Гука, деформация пружины также увеличивается. Это указывает на то, что относительное положение человека и робота становится более заметным. Поэтому за одно и то же время роботу необходимо выполнить больший объем перемещения и поворота.

4. Нечеткий вывод

Нечеткий вывод является ключевой частью нечеткого контроллера. Это механизм вывода, который осуществляет процесс вывода на основе библиотеки правил нечеткого управления. В этой статье для входных данных суммарной силы F и разницы сил ΔF предусмотрено по 12 подмножеств, поэтому обычно необходимо установить 144 правила.

5. Дефаззификация

Поскольку результат, полученный в результате нечеткого вывода, по-прежнему является нечетким значением, а значение двигателя, являющееся фактическим выходным, должно быть

точным числовым значением, поэтому после получения результата нечеткого вывода необходимо провести дефаззификацию, чтобы отобразить нечеткое значение из нечеткого пространства в вещественное пространство. Поскольку метод взвешенного среднего дает более плавный выход.Его формула следующая:

$$v = \sum_{i=1}^{12} \frac{\mu_i^S v_i}{\mu_i^S} = \frac{\mu_1^S v_1 + \mu_2^S v_2 + \dots + \mu_{12}^S v_{12}}{\mu_1^S + \mu_2^S + \dots + \mu_{12}^S} \quad (3-46)$$

$$\Delta v = \sum_{i=1}^{12} \frac{\mu_i^R \Delta v_i}{\mu_i^R} = \frac{\mu_1^R \Delta v_1 + \mu_2^R \Delta v_2 + \dots + \mu_{12}^R \Delta v_{12}}{\mu_1^R + \mu_2^R + \dots + \mu_{12}^R} \quad (3-47)$$

3.3.4 Проектирование нечеткого управления ходовой платформой

1.Проектирование системы нечеткого управления ходовой платформой включает два основных входных переменных и два выходных переменных. Диапазоны входных и выходных переменных были определены на основе комфортности ходьбы пациента.

Входные переменные:

Сила F : Диапазон от 0 до 20, отражает общую силу, приложенную пациентом для движения платформы.

Разность сил ΔF : Диапазон от -20 до 20, указывает на разницу сил между левой и правой сторонами, используется для определения намерения поворота.

Выходные переменные:

Скорость движения v : Диапазон от 0 до 1200 мм/с, управляет скоростью движения платформы вперёд.

Приращение скорости поворота Δv : Диапазон от 0 до 500 мм/с, регулирует скорость колес с учётом разницы сил для реализации поворота.

Для каждой из входных и выходных переменных было спроектировано 12 нечетких подмножеств, использующих треугольные функции принадлежности для упрощения вычислений и обеспечения покрытия всего диапазона операций.

2. Дизайн функции принадлежности

Для каждой входной и выходной переменной было спроектировано 12 нечетких подмножеств, использующих треугольные функции принадлежности для упрощения вычислений и обеспечения покрытия всего диапазона операций.

Функция принадлежности для силы F варьируется от "очень маленькой" до "максимальной". Конкретное разбиение следующее:

$F1, F2, \dots, F12$: равномерно распределённые сегменты, каждый из которых представляет собой уровень силы.

Таблица 3.2 Функция принадлежности для силы F

Название подмножества	Центральное значение	Левый конец	Правый конец	Описание
F1	0	0	1.8	Very Very Small
F2	1.8	0	3.6	Very Small
F3	3.6	1.8	5.4	Small
F4	5.4	3.6	7.2	S-M
F5	7.2	5.4	9	Medium
F6	9	7.2	10.8	M-L
F7	10.8	9	12.6	Large
F8	12.6	10.8	14.4	Very Large
F9	14.4	12.6	16.2	Extra Large
F10	16.2	14.4	18	Huge
F11	18	16.2	20	V. Huge
F12	20	18	20	Extremely Huge

Функция принадлежности для разности сил ΔF варьируется от "крайне левого" до "крайне правого", отражая намерение и амплитуду поворота.

$\Delta F1, \Delta F2, \dots, \Delta F12$: от отрицательного значения до положительного, представляющие процесс от левого поворота к правому повороту.

Таблица 3.3 Функция принадлежности для разности сил ΔF

Название	Центральное	Левый	Правый	Описание
----------	-------------	-------	--------	----------

подмножества	значение	конец	конец	
$\Delta F1$	-20	-20	-16.36	Very Very Left
$\Delta F2$	-16.36	-20	-12.73	Very Left
$\Delta F3$	-12.73	-16.36	-9.09	Big Left
$\Delta F4$	-9.09	-12.73	-5.45	Medium-Left
$\Delta F5$	-5.45	-9.09	-1.82	Small-Left
$\Delta F6$	-1.82	-5.45	1.82	Near Zero
$\Delta F7$	1.82	-1.82	5.45	Small-Right
$\Delta F8$	5.45	1.82	9.09	Medium-Right
$\Delta F9$	9.09	5.45	12.73	Big Right
$\Delta F10$	12.73	9.09	16.36	Very Right
$\Delta F11$	16.36	12.73	20	Very Very Right
$\Delta F12$	20	16.36	20	Max Right

Скорость движения v : Диапазон от 0 до 1200, контролирует скорость движения платформы вперёд. 12 подмножеств равномерно распределены по диапазону, скорость увеличивается по мере перехода от одного подмножества к другому.

Таблица 3.4 Функция принадлежности для скорости движения v

Название подмножества	Центральное значение	Левый конец	Правый конец	Описание
V1	0	0	100	Very Very Slow
V2	100	0	200	Very Slow
V3	200	100	300	Slow
V4	300	200	400	Slightly Slow
V5	400	300	500	Medium
V6	500	400	600	Med-Fast

V7	600	500	700	Slightly Fast
V8	700	600	800	快 (Fast)
V9	800	700	900	Very Fast
V10	900	800	1000	Extremely Fast
V11	1000	900	1100	Super Fast
V12	1200	1100	1200	Max Speed

Приращение скорости поворота Δv : Диапазон от 0 до 500, регулирует скорость колес с учётом разницы сил для реализации поворота.

Таблица 3.5. Функция принадлежности для приращения скорости поворота Δv

Название подмножества	Центральное значение	Левый конец	Правый конец	Описание
$\Delta V1$	0	0	45	Very small increment
$\Delta V2$	45	45	90	Small increment
$\Delta V3$	90	90	135	Moderate increment
$\Delta V4$	135	135	180	Medium increment
$\Delta V5$	180	180	225	Relatively large increment
$\Delta V6$	225	225	270	Large increment
$\Delta V7$	270	270	315	Larger increment
$\Delta V8$	315	315	360	Much larger increment
$\Delta V9$	360	360	405	Very large increment
$\Delta V10$	405	405	450	Extremely large increment
$\Delta V11$	450	450	495	Maximum increment
$\Delta V12$	495	495	500	Ultimate increment

Согласно областям определения входных и выходных переменных, в модуле Fuzzy в Matlab были получены функции

принадлежности для входных переменных силы F и разности сил ΔF , как показано на Рисунках 3.8 и 3.9. Функции принадлежности для выходной скорости v и приращения угловой скорости Δv представлены на Рисунках 3.10 и 3.11.

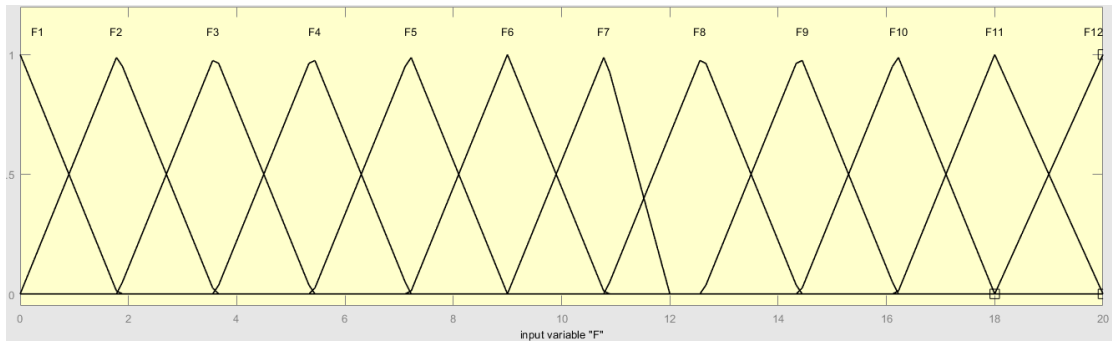


Рисунок 3.8. Функция принадлежности для входной переменной силы F

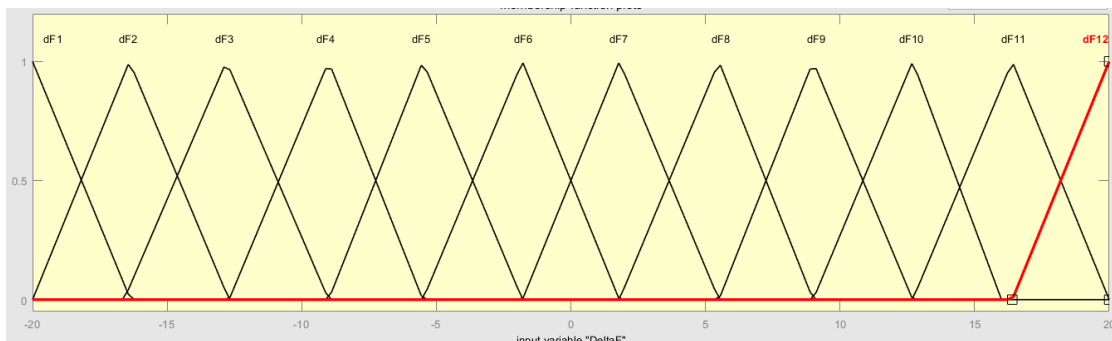


Рисунок 3.9. Функция принадлежности для входной переменной разности сил

ΔF

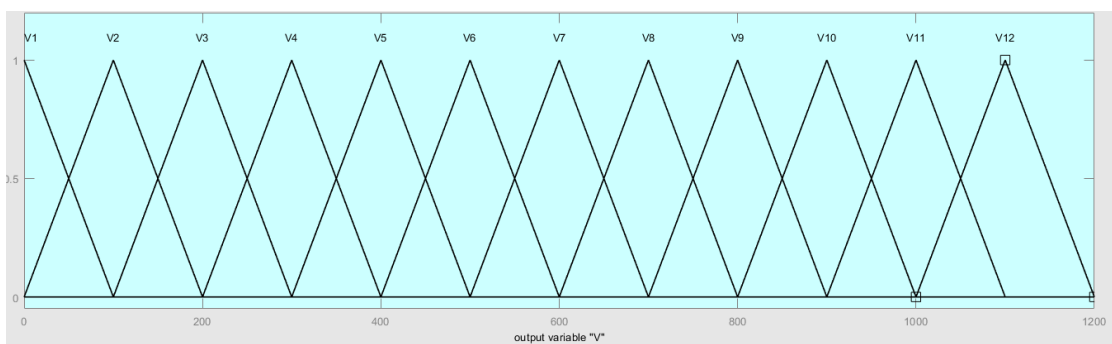


Рисунок 3.10. Функция принадлежности для выходной скорости v

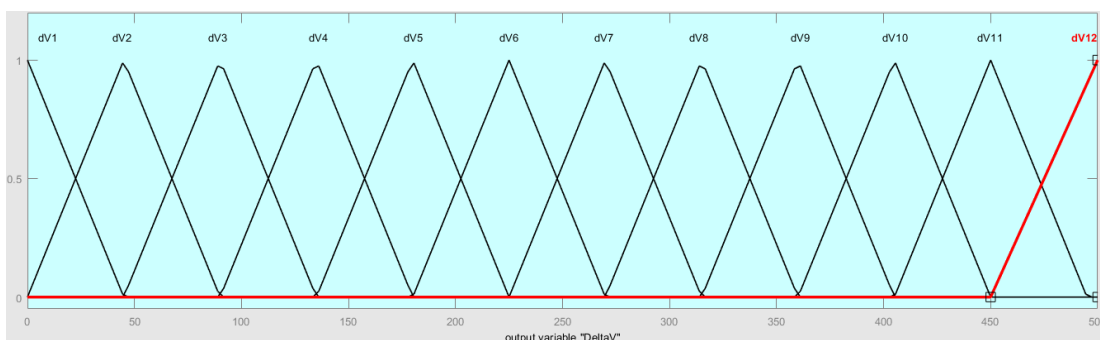


Рисунок 3.11. Функция принадлежности для приращения угловой скорости

$$\Delta v$$

3. Разработка нечетких правил

Нечеткие правила разрабатываются на основе динамической реакции ходовой платформы и привычек пациента. Дизайн правил основывается на принципах безопасности и интуитивности, чтобы гарантировать стабильный отклик платформы в любых операционных ситуациях:

Основные правила: Когда сила мала, а разница сил близка к нулю, платформа должна поддерживать низкую скорость или оставаться в покое.

Продвинутое правило: Когда сила велика и имеется значительная разница сил, платформа должна быстро реагировать, ускоряться вперед и корректировать поворот в зависимости от разницы сил.

В данной работе для входных переменных F и ΔF определено по 12 подмножеств для каждой, что позволяет создать $12 \times 12 = 144$

нечетких управляющих правил. Нечеткие правила приведены в

Таблице 4.6.

Таблица 3.6. Таблица нечетких правил

F	$\Delta F1$	$\Delta F2$	$\Delta F3$	$\Delta F4$	$\Delta F5$	$\Delta F6$
F1	V1, $\Delta v12$	V1, $\Delta v10$	V1, $\Delta v7$	V1, $\Delta v5$	V1, $\Delta v3$	V1, $\Delta v1$
F2	V2, $\Delta v12$	V2, $\Delta v10$	V2, $\Delta v7$	V2, $\Delta v5$	V2, $\Delta v3$	V2, $\Delta v1$
F3	V3, $\Delta v12$	V3, $\Delta v10$	V3, $\Delta v7$	V3, $\Delta v5$	V3, $\Delta v3$	V3, $\Delta v1$
F4	V4, $\Delta v12$	V4, $\Delta v10$	V4, $\Delta v7$	V4, $\Delta v5$	V4, $\Delta v3$	V4, $\Delta v1$
F5	V4, $\Delta v11$	V5, $\Delta v10$	V5, $\Delta v7$	V5, $\Delta v5$	V5, $\Delta v3$	V5, $\Delta v1$
F6	V5, $\Delta v10$	V6, $\Delta v10$	V6, $\Delta v7$	V6, $\Delta v5$	V6, $\Delta v3$	V6, $\Delta v1$
F7	V7, $\Delta v10$	V7, $\Delta v10$	V7, $\Delta v7$	V7, $\Delta v5$	V7, $\Delta v3$	V7, $\Delta v1$
F8	V8, $\Delta v9$	V8, $\Delta v9$	V8, $\Delta v7$	V8, $\Delta v5$	V8, $\Delta v3$	V8, $\Delta v1$
F9	V8, $\Delta v8$	V9, $\Delta v9$	V9, $\Delta v7$	V9, $\Delta v5$	V9, $\Delta v3$	V8, $\Delta v1$
F10	V10, $\Delta v8$	V10, $\Delta v9$	V10, $\Delta v7$	V10, $\Delta v5$	V9, $\Delta v3$	V9, $\Delta v1$
F11	V10, $\Delta v7$	V10, $\Delta v8$	V10, $\Delta v7$	V10, $\Delta v5$	V9, $\Delta v3$	V9, $\Delta v1$
F12	V11, $\Delta v6$	V10, $\Delta v7$	V10, $\Delta v7$	V10, $\Delta v5$	V11, $\Delta v3$	V12, $\Delta v1$
F	$\Delta F7$	$\Delta F8$	$\Delta F9$	$\Delta F10$	$\Delta F11$	$\Delta F12$
F1	V1, $\Delta v1$	V1, $\Delta v3$	V1, $\Delta v5$	V1, $\Delta v7$	V1, $\Delta v10$	V1, $\Delta v12$
F2	V2, $\Delta v1$	V2, $\Delta v3$	V2, $\Delta v5$	V2, $\Delta v7$	V2, $\Delta v10$	V2, $\Delta v12$
F3	V3, $\Delta v1$	V3, $\Delta v3$	V3, $\Delta v5$	V3, $\Delta v7$	V3, $\Delta v10$	V3, $\Delta v12$
F4	V4, $\Delta v1$	V4, $\Delta v3$	V4, $\Delta v5$	V4, $\Delta v7$	V4, $\Delta v10$	V4, $\Delta v12$
F5	V5, $\Delta v1$	V5, $\Delta v3$	V5, $\Delta v5$	V5, $\Delta v7$	V5, $\Delta v10$	V4, $\Delta v11$
F6	V6, $\Delta v1$	V6, $\Delta v3$	V6, $\Delta v5$	V6, $\Delta v7$	V6, $\Delta v10$	V5, $\Delta v10$
F7	V7, $\Delta v1$	V7, $\Delta v3$	V7, $\Delta v5$	V7, $\Delta v7$	V7, $\Delta v10$	V7, $\Delta v10$
F8	V8, $\Delta v1$	V8, $\Delta v3$	V8, $\Delta v5$	V8, $\Delta v7$	V8, $\Delta v9$	V8, $\Delta v9$
F9	V8, $\Delta v1$	V9, $\Delta v3$	V9, $\Delta v5$	V9, $\Delta v7$	V9, $\Delta v9$	V9, $\Delta v9$
F10	V9, $\Delta v1$	V9, $\Delta v3$	V10, $\Delta v5$	V10, $\Delta v7$	V10, $\Delta v9$	V10, $\Delta v8$
F11	V9, $\Delta v1$	V9, $\Delta v3$	V10, $\Delta v5$	V10, $\Delta v7$	V10, $\Delta v8$	V10, $\Delta v7$
F12	V12, $\Delta v1$	V11, $\Delta v3$	V10, $\Delta v5$	V10, $\Delta v7$	V10, $\Delta v7$	V11, $\Delta v6$

В практическом применении вышеуказанной системы линейного нечеткого управления, хотя теоретически возможно создать отдельное правило для каждой комбинации входных данных, на практике это не всегда реализуется [88,89]. Основным недостатком данного подхода является то, что он недостаточно учитывает особые ситуации, которые могут возникнуть в процессе реабилитационной тренировки. Например, когда результирующая сила велика и разница между двумя силами также значительна, необходимо снизить скорость поворота, чтобы предотвратить падение пользователя. Поэтому исходные 144 нечетких правила были скорректированы в соответствии с требованиями безопасности, чтобы лучше соответствовать реальным условиям реабилитации и повысить безопасность и практичность системы.

Согласно нечетким правилам, зависимость между входными переменными — силой F и разностью сил ΔF — и выходными переменными v и Δv изображена на графиках на Рисунках 3.12 и 3.13. Из графиков видно, что входные и выходные переменные не перекрываются, что подтверждает, что управляющие правила соответствуют проектным требованиям.

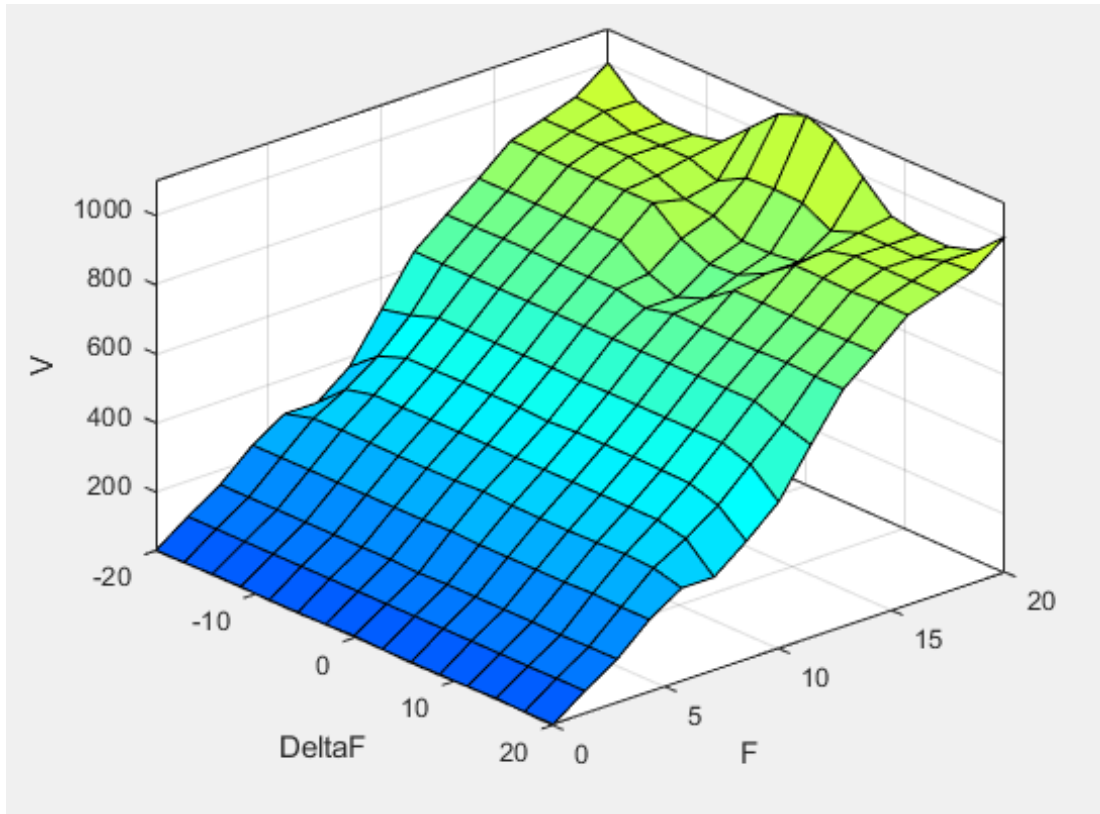


Рисунок 3.12. Схематическое изображение поверхности зависимости силы F и разности сил ΔF от скорости v

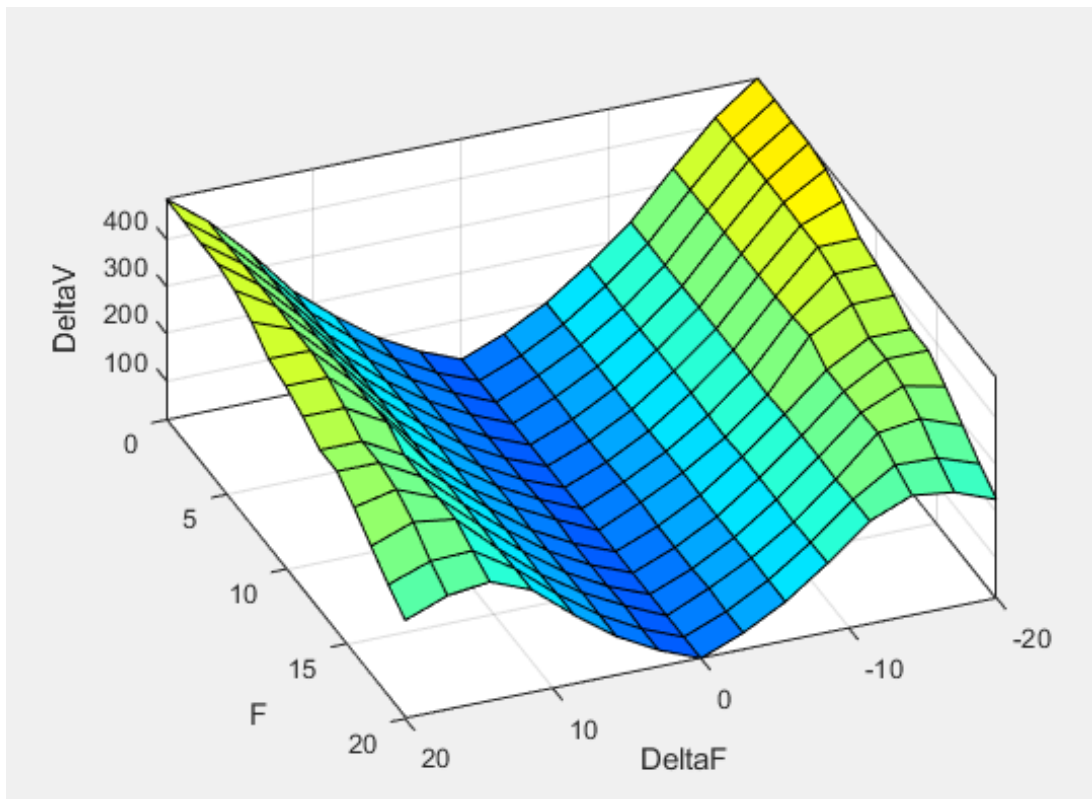


Рисунок 3.13. Схематическое изображение поверхности зависимости силы F

и разности сил ΔF от приращения угловой скорости Δv

После вычисления скорости Δv , в зависимости от знака ΔF , распределяется скорость между левым и правым колесом следующим образом:

Если $\Delta F > 0$, то скорость правого колеса увеличивается на Δv , то время как скорость левого колеса остается постоянной (v_L).

Если $\Delta F < 0$, то скорость левого колеса увеличивается на Δv , в то время как скорость правого колеса остается постоянной (v_R).

Скорости левой и правой сторон ходовой платформы могут быть вычислены по следующей формуле:

$$v = \begin{bmatrix} v_L \\ v_R \end{bmatrix} \quad (3-48)$$

$$V_{new} = v + C \quad (3-49)$$

Значение C зависит от значения ΔF :

$$C = \begin{cases} 0 & \text{if } \Delta F > 0 \\ \Delta v & \\ \Delta v & \text{if } \Delta F < 0 \\ 0 & \end{cases} \quad (3-50)$$

Выводы по третьей главе

1. С помощью метода Д-Н был построен кинематический модель робота, и в сочетании с Лагранжевой динамикой было чётко описано

взаимное движение частей робота, что обеспечило теоретическую поддержку для разработки последующих стратегий управления.

2. На основе динамического уравнения подъемной части был предложен комбинированный алгоритм управления с использованием адмитанса и PID-регулятора, а также комбинированный алгоритм с использованием адмитанса и скользящего управления. После сравнения было установлено, что PID-регулятор имеет лучшие характеристики по мягкости и стабильности.

3. Были разработаны кинематическая и динамическая модели ходовой платформы, проанализирован процесс моделирования нечеткого управления для реабилитационного робота нижних конечностей. В ходе эксперимента были установлены области определения входных и выходных параметров, а также соответствующие подмножества для каждой области.

4. На основе силы и разницы силы с обеих сторон датчиков поддерживающей части был спроектирован нечеткий регулятор. С помощью нечетких правил, функции принадлежности и алгоритма интерполяции была реализована гибкая настройка скорости и направления платформы.

Глава 4. Экспериментальная проверка робота для реабилитации нижних конечностей

4.1 Трёхмерное динамическое отслеживание движения таза

4.1.1 Подготовка к эксперименту

Для сбора данных о движении использовалась система захвата движения Optitrack, разработанная американским производителем оптического оборудования Natural Point. Как показано на рисунке 4.1, система состоит из 8 фиксированных камер для захвата движения. Разрешение камеры: 1.3 МП, Частота кадров: 200 FPS, Частота дискретизации данных (в эксперименте с фиксированными камерами): 120 Гц



Рисунок 4.1. Камера захвата движения Prime13 и маркеры (mark)

Расположение камер и экспериментальной площадки показано на рисунке 4.2. Система Optitrack использует оптическую технологию захвата движения для сбора информации о положении и ориентации маркеров (mark), размещённых на разных частях тела человека. В результате трёхмерные координаты преобразуются в необходимые кинематические данные.



Рисунок 4.2. Расположение оборудования системы захвата движения

4.1.2 Экспериментальный процесс

После завершения подготовки экспериментальной площадки следующим шагом является калибровка системы Motive. Камеры настраиваются на фокусировку с последующей фиксацией диафрагмы, а в программном обеспечении Motive регулируется значение экспозиции для повышения видимости камер.

После настройки параметров камеры корректируется угол обзора, чтобы гарантировать, что испытуемый находится в зоне видимости камер [90,91]. Для записи данных о походке, из-за большого диапазона движения, используется Т-образный калибровочный стержень длиной 500 мм, как показано на рисунке 4.3.



Рисунок 4.3. Т-образный калибровочный стержень

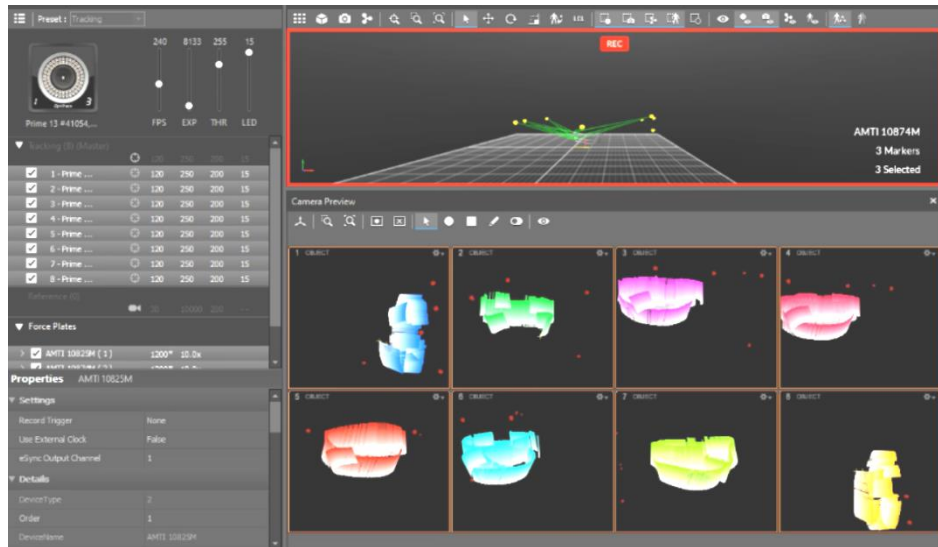


Рисунок 4.4. Интерфейс калибровки системы

После завершения калибровки системы проводится сбор пространственных данных объектов, подвергающихся захвату. На кожу испытуемого приклеиваются отражающие маркеры, или mark-точки, представляющие собой серебристо-серые шарики с покрытием из специального отражающего материала.

Так как во время записи движений одежда испытуемого может смещаться относительно точек измерения, это может привести к неточной фиксации пространственного положения mark-точек или даже к их частичному перекрытию, что вызывает сбой в записи

движений. Поэтому оптимальным решением является приклеивание mark-точек непосредственно на кожу испытуемого.

На рисунке 4.5 представлены эксперименты по захвату движений верхних и нижних конечностей, а также движения таза отдельно.



(a) 21 маркер



(b) 5 маркеров на тазовой области

Рисунок 4.5. Размещение маркеров (mark) на теле испытуемого

Светодиодные модули на фиксированных камерах системы захвата движения излучают инфракрасный свет на mark-точки, размещённые на теле испытуемого. Камеры улавливают отражённый от mark-точек инфракрасный свет с помощью матриц датчиков. После того как информация о mark-точках захвачена несколькими камерами одновременно, вычисляются их трёхмерные координаты в пространстве.

Позиционная информация о отражающих mark-точках, полученная камерами, передаётся через сетевой кабель на коммутатор. Затем коммутатор отправляет данные на подключённый компьютер, где они обрабатываются и анализируются с использованием программного обеспечения Motive.

Расстояния между mark-точками на теле испытуемого остаются неизменными. Для отображения взаимного расположения точек соседние точки соединяются линиями, показывающими их относительное положение. Эта группа точек и связей между ними сохраняется в программном обеспечении как единый набор маркеров.

4.1.3 Обработка экспериментальных данных

После того как программное обеспечение получает трёхмерные пространственные координаты заданных mark-точек, оно завершает идентификацию различных частей тела испытуемого и фиксирует координатные данные для разных движений.

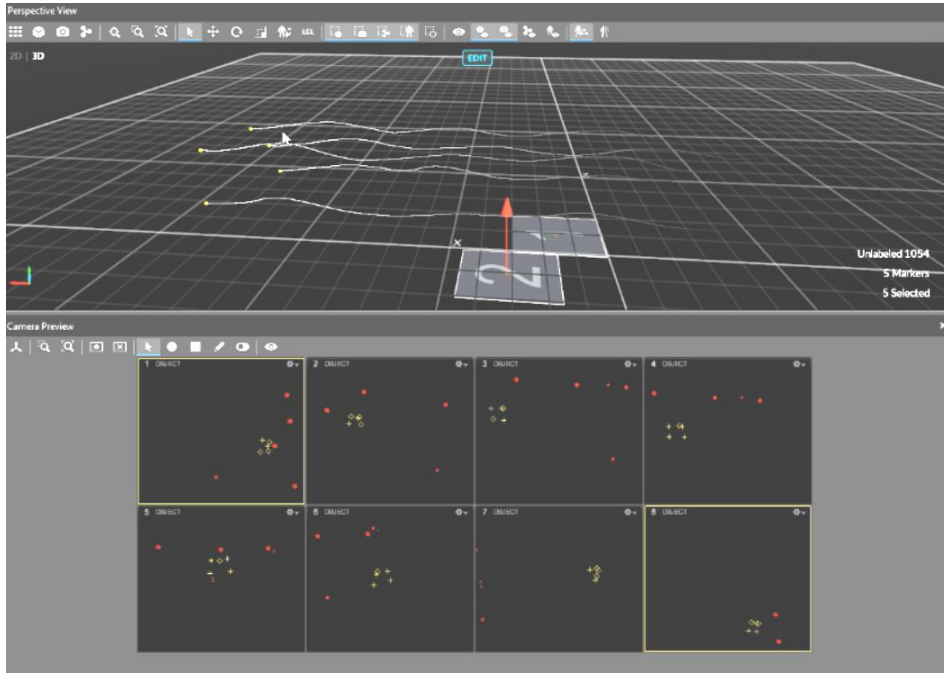


Рисунок 4.6. Траектория движения mark-точек в программном обеспечении

Motive

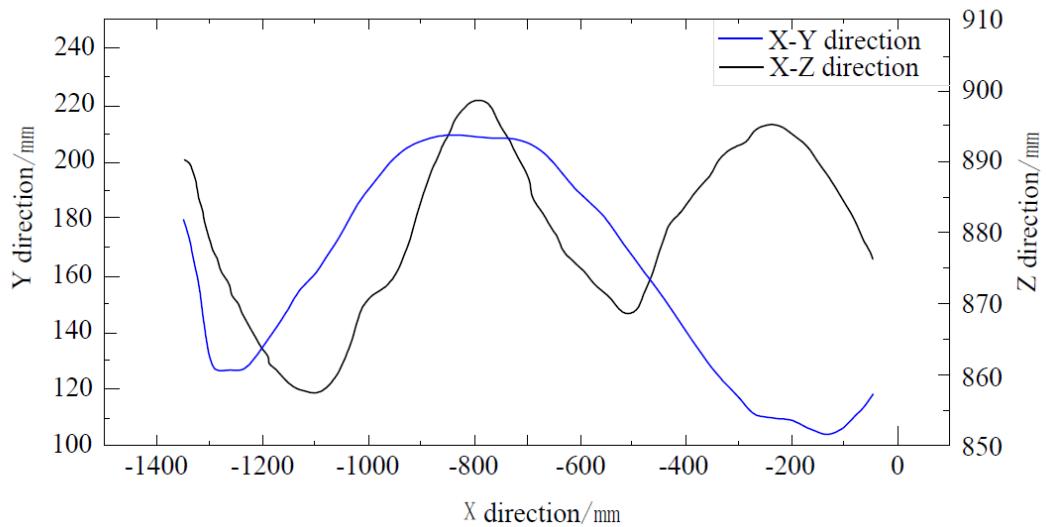


Рисунок 4.7. График данных походки тазобедренного сустава

Из приведённого выше рисунка видно, что движения тазобедренного сустава имеют высокую пространственную сложность, позволяя двигаться одновременно или поочерёдно в направлениях X, Y и Z, демонстрируя сложные трёхмерные

характеристики движения. Поэтому робот для реабилитации нижних конечностей должен обладать многомерной степенью свободы движения.

Конструкция робота, разработанного в данной работе, может адаптироваться к сгибанию и разгибанию, отведению и приведению, а также вращательным движениям тазобедренного сустава. На основе собранных данных была обоснована конструкция: диапазон высоты подъёмного механизма, амплитуда горизонтального перемещения бокового модуля и изменения высоты обратного модуля.

4.2 Обработка сигналов датчиков с использованием фильтрации

В области обработки сигналов фильтрация часто осуществляется с помощью фильтра Калмана. При использовании сигналов датчиков в качестве входных данных для управления выбор подходящего фильтра играет ключевую роль.

4.2.1 Фильтр Калмана

Фильтр Калмана (Kalman Filter, KF) представляет собой линейный гауссовский фильтр, основная идея которого заключается в использовании минимальной средней квадратичной ошибки в качестве критерия оптимальной оценки. Благодаря своей высокой точности и хорошей адаптивности фильтр Калмана широко

применяется для оценки параметров различных линейных процессов [93].

Модель состояния пространства объединяет сигнал и шум, интегрируя оценку предыдущего момента времени с текущим наблюдением. Это позволяет обновить оценку переменных состояния и получить их значения для текущего момента времени.

Основой фильтра Калмана является построение подходящих системных уравнений и уравнений предсказания, что обеспечивает эффективную и точную оценку состояния системы.

Определив переменную состояния системы как $X_K \in R^n$, а управляющий вход системы как U_K , уравнение состояния системы может быть записано следующим образом:

$$X_K = AX_{K-1} + BU_K + W_K \quad (4-1)$$

Определив наблюдаемую переменную как Z_K , уравнение наблюдения записывается следующим образом:

$$Z_K = HX_K + V_K \quad (4-2)$$

Предполагается, что W_K и V_K независимы и подчиняются нормальному распределению белого шума.

Ковариационная матрица шума процесса обозначается как Q ;

Ковариационная матрица шума наблюдения обозначается как R .

Соответственно:

$$W_k \sim N(0, Q) \quad (4-3)$$

$$V_k \sim N(0, R) \quad (4-4)$$

A , B , H называются матрицами перехода состояния, и здесь они предполагаются константами. Алгоритм фильтра Калмана делится на два основных этапа [94,95]:

Этап предсказания: На основе апостериорной оценки состояния на предыдущем шаге производится оценка состояния на текущем шаге.

Этап обновления: Используются текущие измеренные значения для коррекции предсказанной оценки состояния.

Ниже приведены уравнения времени и обновления для дискретного фильтра Калмана.

Уравнение времени:

$$\hat{X}_k = A\hat{X}_{k-1} + B\hat{U}_k \quad (4-5)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (4-6)$$

Уравнение обновления состояния:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (4-7)$$

$$\hat{X}_k = \hat{X}_k^- + K_k (Z_k - H\hat{X}_k^-) \quad (4-8)$$

$$P_k = (I - K_k H)P_k^- \quad (4-9)$$

В моменты времени $k-1$ и k апостериорные оценки состояния равны \hat{X}_{k-1} , \hat{X}_k , соответственно, а априорная оценка состояния в момент времени k равна \hat{X}_k^- .

Матрицы перехода состояния, управления входом и модели наблюдения обозначаются как A , B и H соответственно.

P_k , P_k^- представляют ковариационные матрицы ошибок априорной и апостериорной оценки, Q и R — ковариационные матрицы шума процесса и шума измерений соответственно.

I это единичная матрица. K_k это калмановское усиление.

С использованием приведённого алгоритма реализована фильтрация данных с помощью фильтра Калмана, и полученные данные давления с датчиков были обработаны. Кривая сигнала давления после фильтрации показана на рисунке.

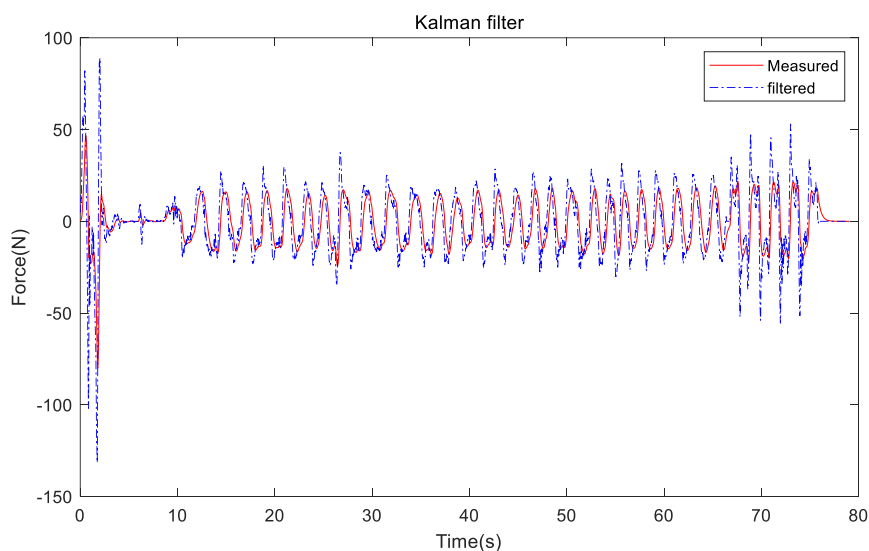


Рисунок 4.8. Фильтрация с использованием фильтра Калмана

4.2.2 FIR-фильтрация

FIR (Finite Impulse Response) фильтр, или фильтр с конечной импульсной характеристикой, является одним из базовых фильтров в цифровых системах обработки сигналов. В отличие от других

типов фильтров, он обладает превосходной линейной характеристикой фазового отклика. Поскольку его импульсная характеристика ограничена, данный фильтр представляет собой устойчивую линейную систему [96,97].

Прямая сетевая структура FIR-системы непосредственно отражает частотную характеристику или разностное уравнение FIR-цифрового фильтра. Разностное уравнение выражается формулой (3-2), где $x[n]$ можно рассматривать как результат прямой свёртки с импульсной характеристикой, заданной формулой (3-3).

Структура прямой реализации показана на рисунке 4.9. Из рисунка видно, что она представляет собой цепную структуру, где каждый сигнал в цепи умножается на соответствующий коэффициент (значение импульсной характеристики). Выход $y[n]$ является суммой произведений сигналов и коэффициентов по всей цепи.

В случае, когда объём данных выборки и коэффициентов фильтра небольшой, но требуется фильтр высокого порядка, рекомендуется использовать систему с прямой сетевой структурой [98-100].

$$y[n] = \sum_{k=0}^M b_k x[n-k] \quad (4-10)$$

$$h[n] = \begin{cases} bn & n = 0, 1, 2, \dots, M \\ 0 & \text{其他} \end{cases} \quad (4-11)$$

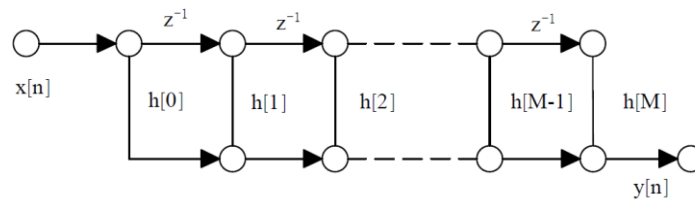


Рисунок 4.9. Прямая реализация FIR-системы

С использованием приведённой программы был реализован FIR-фильтр, который применялся для фильтрации сигналов, полученных от датчиков давления. Кривая сигнала давления после фильтрации показана на соответствующем рисунке.

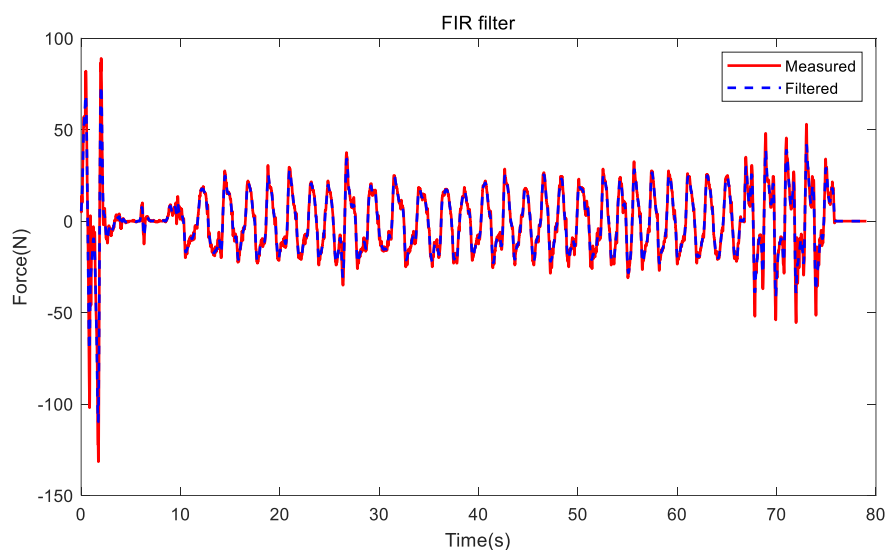


Рисунок 4.0. График1 фильтрации FIR

4.2.3 IIR-фильтрация

IIR-фильтр (Infinite Impulse Response) — это распространённый тип цифрового фильтра, особенностью которого является то, что его выход представляет собой взвешенную сумму текущих и прошлых

значений входного сигнала. Это делает его широко применимым в обработке цифровых сигналов [101].

Основной принцип работы ИИР-фильтра включает понятия разностного уравнения и импульсной характеристики. Типичное разностное уравнение ИИР-фильтра можно записать в следующей форме:

$$y[n] = \sum_{k=0}^M b_k \cdot x[n-k] - \sum_{k=1}^N a_k \cdot y[n-k] \quad (4-12)$$

Где $x[n]$ — входной сигнал, $y[n]$ — выходной сигнал, b_k и a_k — коэффициенты фильтра, M и N — порядки передающего и обратного каналов соответственно.

ИИР-фильтр называется "фильтром с бесконечной импульсной характеристикой", поскольку его импульсная характеристика (или единичный отклик) бесконечно длинна во времени. Это означает, что выходной сигнал зависит не только от текущего значения входного сигнала, но и от предыдущих значений выходного сигнала [102,103].

ИИР-фильтры, как один из самых распространённых типов цифровых фильтров, находят широкое применение в обработке сигналов. Они используются для фильтрации, подавления шума, извлечения признаков и других операций с сигналами [104].

С помощью описанной программы был реализован ИИР-фильтр для обработки сигналов давления, полученных с датчиков. Кривая

фильтрованного сигнала давления показана на соответствующем рисунке.

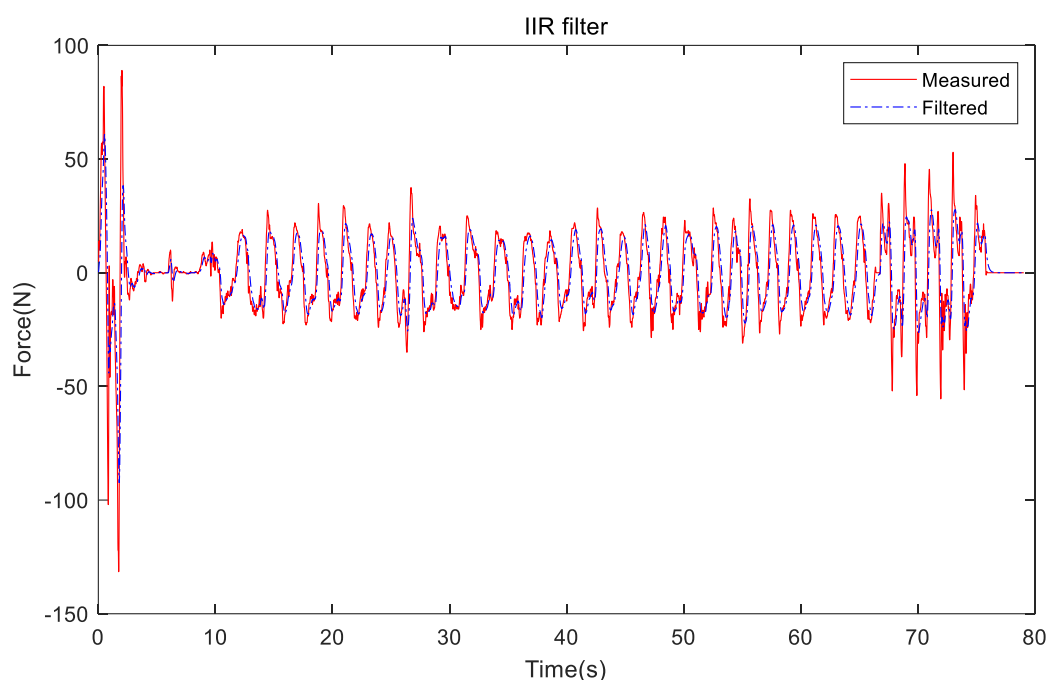


Рисунок 4.11. График фильтрации IIR

Согласно результатам эксперимента, фильтр Калмана показал наилучший эффект среди всех методов фильтрации. Хотя IIR-фильтр эффективно устраняет резкие пики в сигнале, его эффективность ограничивается только высокочастотным шумом. Для более плавных участков сигнала фильтрация оказывается менее заметной, что приводит к сохранению определённых колебаний и нерегулярностей в отфильтрованном сигнале. Кроме того, несмотря на то, что механизм обратной связи IIR-фильтра частично сглаживает сигнал, он не способен полностью справиться с комплексными шумовыми компонентами, что мешает достижению точного сглаживания.

FIR-фильтр, напротив, характеризуется более простой обработкой. Он сглаживает сигнал, используя взвешенное усреднение исторических данных. Хотя это помогает снизить уровень шума, из-за отсутствия механизма динамической настройки к шуму его возможности ограничены. FIR-фильтр менее эффективен при обработке сложного шума или значительных изменений сигнала. Особенно при обработке высокочастотного шума его эффективность значительно уступает фильтру Калмана.

По сравнению с другими методами, фильтр Калмана продемонстрировал выдающиеся преимущества. Он не только эффективно устраняет резкие пики в сигнале, но и обеспечивает тонкое сглаживание стабильных участков сигнала. Комбинируя динамическую модель системы и модель шума, фильтр Калмана способен в реальном времени обновлять оценку состояния и динамически корректировать коэффициент усиления фильтра в зависимости от характеристик шума. Это позволяет достичь более точной и стабильной фильтрации.

Его уникальная способность к оптимальной оценке и моделированию шума позволяет фильтру Калмана максимально сохранять полезную информацию сигнала, устраняя избыточный шум и ненужные колебания. Таким образом, он успешно обеспечивает точное отслеживание сложных сигналов.

В итоге, фильтр Калмана показал непревзойдённые преимущества в обработке сигналов датчиков с шумами. Он обладает не только высокой точностью фильтрации, но и отличной адаптивностью и способностью к динамической настройке. Поэтому в данной работе для обработки сигналов был выбран метод фильтрации на основе фильтра Калмана.

4.3 Экспериментальная проверка движения робота для реабилитации нижних конечностей

4.3.1 Описание условий эксперимента

Для подтверждения работоспособности робота в качестве устройства помощи при ходьбе, а также для проверки эффективности динамической модели и алгоритма управления взаимодействующими силами, в данной главе проведён ряд экспериментальных исследований на основе предыдущих результатов.

Посредством экспериментов по оценке возможности помощи при ходьбе, сравнительных испытаний алгоритмов управления усилием толчка и разгрузочной поддержкой была проверена эффективность модели системы робота для помощи при ходьбе. Также были подтверждены правильность и стабильность предложенных алгоритмов управления. Эти эксперименты заложили

основу для дальнейшего улучшения структуры и алгоритмов управления реабилитационного робота.

Система управления взаимодействует с человеком через датчики крутящего момента и давления, выполняя сбор и фильтрацию сигналов. Затем проводится распознавание намерений движения, и управляющие команды передаются на сервоприводы. Один сервопривод управляет механизмом вертикального перемещения, а два других сервопривода контролируют движение робота вперед, назад или поворот.

Кроме того, система управления отслеживает ускорение механизма вертикального перемещения. Если ускорение превышает 0.5 м/с^2 , это расценивается как падение пациента. В таком случае все сервоприводы блокируются, а промышленный компьютер подаёт сигнал тревоги, чтобы обеспечить безопасность пациента.

В данном исследовании используется прототип реабилитационного робота для помощи при ходьбе. Испытуемый фиксируется к роботу с помощью поясного ремня, а управление движением робота по поверхности и движением таза осуществляется через датчики давления, расположенные по обе стороны таза. Промышленный компьютер взаимодействует с системой управления через протокол связи RS485 для работы с датчиками и через протокол CAN для связи с сервоприводами.

Движение сервоприводов управляется на основе намерений испытуемого.

На рисунке представлен экспериментальный прототип робота для реабилитации нижних конечностей. Полная масса устройства составляет 180 кг, а его габариты — 1150 мм × 892 мм × 1610 мм, что позволяет ему легко помещаться в лифт.

Высота механизма поддержки таза регулируется непрерывно в диапазоне от 550 мм до 1050 мм, что делает робот подходящим для пациентов разного пола и роста.



(a) Вид спереди



(b) Вид сзади

Рисунок 4.12. Экспериментальный прототип робота для реабилитации нижних конечностей

В ходе экспериментов испытуемым требовалось выполнить тесты, включающие подъёмы и спуски таза, а также ходьбу. Эти действия предъявляли высокие требования к реальной временной реакции и стабильности системы.

Так как в экспериментах использовались цифровые преобразователи, подключённые через интерфейс RS485, их встроенная функция цифровой фильтрации частично уменьшала шумы исходного сигнала. Это обеспечивало относительную стабильность сигналов, поступающих с датчиков давления и крутящего момента. Однако при прямом использовании необработанных сигналов от датчиков для управления сервоприводами наблюдались заметные колебания. Это проявлялось в виде резких пиков в сигналах, которые вызывали внезапные изменения момента силы на выходе сервоприводов, что, в свою очередь, приводило к вибрации и нестабильности всей системы.

Для устранения этой проблемы в эксперименте был применён алгоритм фильтрации Калмана для вторичной обработки сигналов датчиков. Фильтр Калмана — это рекурсивный алгоритм, основанный на оценке состояния, который способен сглаживать сигнал даже при наличии шумовых помех и эффективно выделять его истинную составляющую. После обработки с использованием фильтра Калмана колебания сигналов датчиков были значительно

снижены, а момент силы сервоприводов стал более стабильным. Это позволило избежать вибраций, вызванных внезапными изменениями момента силы.

Кроме того, скорость передачи данных цифрового преобразователя была установлена на уровне 115200 бод. Эта высокая скорость передачи данных обеспечивает высокую степень надёжности и своевременность сигнала.

В системе, оптимизированной с использованием фильтрации, испытуемые практически не испытывали задержек или перебоев в ходе эксперимента. Обратная связь на действия была плавной и естественной.

В целом, благодаря совместному применению цифровой фильтрации и фильтра Калмана, удалось не только повысить стабильность и безопасность системы, но и значительно улучшить пользовательский опыт во время экспериментов. Это создало прочную основу для дальнейшей оптимизации и исследований.

4.3.2 Проверка ходьбы

Как показано на рисунке 4.13, сигнал, отражающий изменение высоты таза испытуемого, был получен путём деления значения сигнала с датчика крутящего момента на длину плеча силы. Для повышения стабильности сигнала исходные данные были

обработаны с помощью фильтра Калмана, что позволило получить более сглаженную кривую.

В ходе эксперимента положение, при котором испытуемый стоит, не прикладывая усилий к роботу, было определено как нулевая точка. Эта точка использовалась в качестве базового уровня для обработки и анализа сигналов.



Рисунок 4.13. Испытуемый в процессе тестирования

Как показано на Рисунке 4.14, когда таз испытуемого поднимался или опускался, реабилитационный робот быстро корректировал свою высоту, синхронизируя её с положением таза испытуемого. Эта способность к синхронизации высоты не только обеспечивает координацию между роботом и испытуемым, но и эффективно предотвращает дискомфорт, вызванный отклонениями в

высоте, значительно повышая безопасность и комфорт в процессе реабилитации.

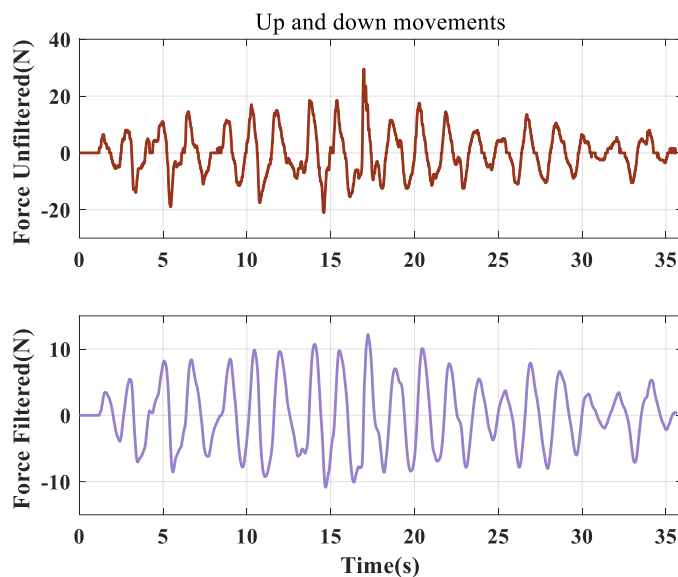


Рисунок 4.14. Сигнал с датчика крутящего момента (после деления на плечо силы) и фильтрация с использованием фильтра Калмана

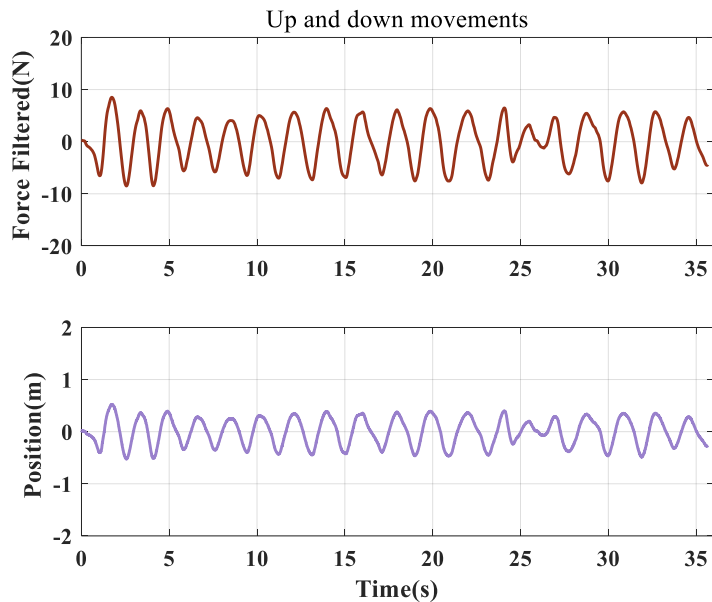


Рисунок 4.15. Сигнал с датчика крутящего момента (после деления на плечо силы) и соответствующие изменения высоты

Как показано на Рисунке 4.16, испытуемый свободно ходил вперёд и назад. Сигнал с датчика давления, фиксирующий

передне-задние движения таза испытуемого, был обработан с помощью фильтра Калмана, что позволило получить сглаженные данные давления.

Отфильтрованный сигнал использовался как вход для обработки с помощью алгоритма нечеткого управления. Нечеткое управление, задавая нечеткие правила для сигнала давления, связывало изменения давления с корректировкой скорости движения робота, обеспечивая плавность и координацию движений.

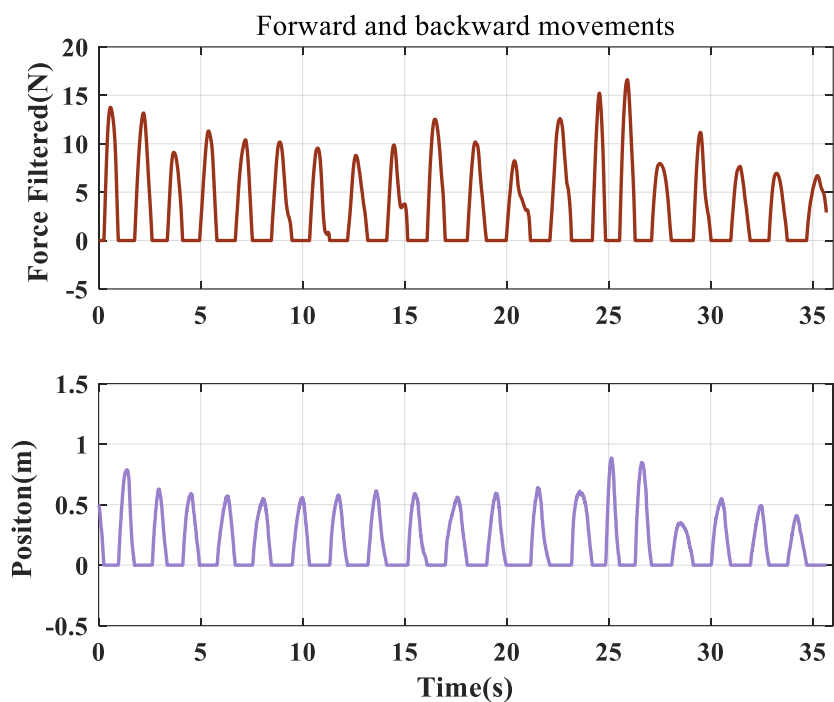


Рисунок 4.16. Сигнал с датчика давления и фильтрация с использованием фильтра Калмана

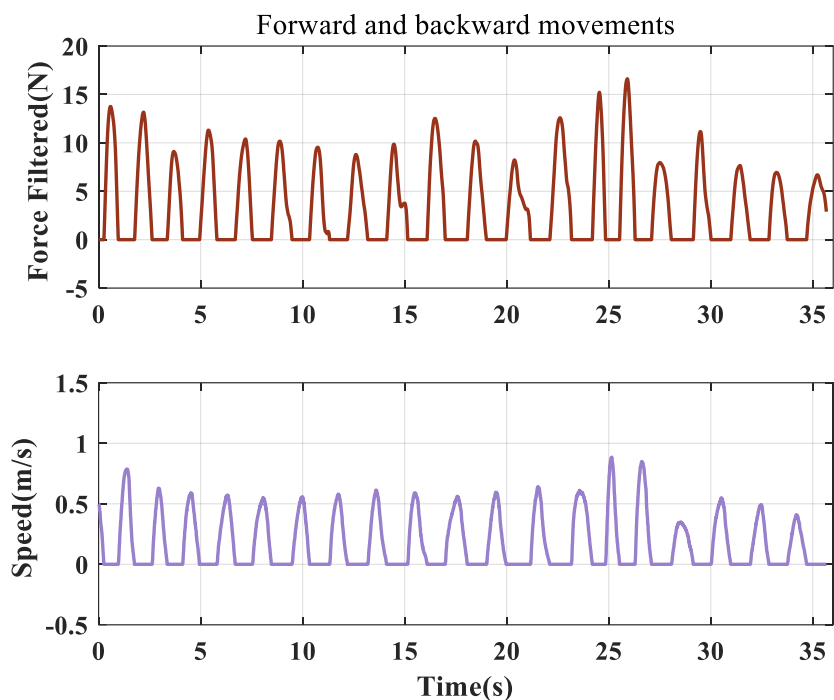


Рисунок 4.17. Сигналы датчиков давления и соответствующая скорость движения

Как показано на рисунке 4.17, обработанные с помощью нечёткого управления сигналы датчиков давления демонстрируют согласованность с тенденцией фактического расстояния, пройденного реабилитационным роботом. В частности, при увеличении сигнала давления расстояние, пройденное роботом, также увеличивается, и наоборот.

Эта согласованность подтверждает способность системы точно распознавать намерения движения испытуемого и выполнять соответствующие действия. Особенно на основе низкошумных и высокоточных сигналов нечёткий алгоритм управления показал отличную способность обработки нелинейных связей, обеспечивая более естественную и плавную реакцию реабилитационного робота.

Результаты эксперимента подтверждают осуществимость и эффективность метода управления движением, основанного на сочетании сигналов датчиков давления и нечёткого управления, в применении к реабилитационным роботам.

Выводы по четвертой главе

1. В данной работе проведена экспериментальная проверка робота для реабилитации нижних конечностей, включая динамическое отслеживание трёхмерного движения таза, обработку сигналов датчиков с использованием фильтрации и тестирование движения робота.

2. Для сбора необходимых кинематических данных была использована система захвата движения OptiTrack, которая фиксировала информацию о положении и ориентации маркеров (mark) на различных частях тела.

3. Были проведены сравнения трёх типов фильтров: фильтра Калмана, FIR-фильтра и IIR-фильтра. Результаты показали, что фильтр Калмана обеспечивает наилучший эффект фильтрации.

4. Проверена применимость и эффективность метода управления движением на основе сигналов от датчиков давления и нечёткого управления в роботе для реабилитации.

ЗАКЛЮЧЕНИЕ

Данный проект направлен на разработку робота для реабилитации нижних конечностей с функцией разгрузки веса и помощи при ходьбе, что позволяет предотвратить случайные падения пациентов во время реабилитационных тренировок. Исследование имеет важное практическое значение для развития медицинского реабилитационного оборудования. В ходе теоретического анализа и экспериментальных проверок были достигнуты следующие промежуточные результаты:

1.С помощью OpenSim была создана модель биомиметического движения суставов, на основе которой была получена траектория движения тазобедренного сустава в трёхмерном пространстве. На основе этой модели разработана структура экзоскелета нижних конечностей, а также реализована система управления, которая объединяет промышленный компьютер, датчики и сервоприводы через шину связи. В ANSYS Workbench проведён статический и динамический анализ конструкции робота, подтвердивший его прочность и безопасность при нагрузке.

2.Методом D-H была построена кинематическая модель робота для реабилитации нижних конечностей. На основе метода Лагранжа были выведены уравнения динамики, которые прояснили взаимосвязь между положением конечного звена и базовой системой

координат, что создало теоретическую основу для дальнейших разработок стратегий управления.

3. Учитывая требования к разгрузке веса пациента, была создана математическая модель и выведены уравнения динамики системы разгрузки. Для управления движением привода предложены методы закрытого регулирования с использованием PID- и скользящего управления. Это позволяет модулю подъёма плавно следовать траектории движения таза пациента. Результаты моделирования в Matlab показали, что PID-регулятор обеспечивает быструю реакцию и высокую точность, удовлетворяя требованиям робота для реабилитации.

4. Проведён кинематический и динамический анализ ходовой платформы. На основе анализа суммы и разницы значений от датчиков давления с обеих сторон была реализована идентификация намерений пациента к движению вперёд или повороту. Применён нечеткий регулятор для настройки входных и выходных доменов, а также подмножества. Контроль ходовой платформы в режиме динамического следования и поворота осуществлялся на основе правил нечёткого управления.

5. С помощью системы захвата движения OptiTrack была зарегистрирована траектория тазобедренного сустава, и её соответствие результатам симуляции в OpenSim было подтверждено.

Проведена экспериментальная оценка трёх методов фильтрации данных: фильтров Калмана, FIR и IIR. Результаты показали, что фильтр Калмана обеспечивает наилучшее качество фильтрации. Дальнейшие эксперименты по ходьбе продемонстрировали хорошую адаптивность прототипа как в конструктивной части, так и в алгоритмах управления.

В заключение, данный проект добился промежуточного прогресса в разработке структуры и стратегии управления роботом для реабилитации нижних конечностей, предоставив важный практический опыт и рекомендации для дальнейших исследований и клинического применения в области реабилитационных технологий.

ЛИТЕРАТУРА

- [1] Zhou J., Yang S., Xue Q. Lower limb rehabilitation exoskeleton robot: A review // *Advances in Mechanical Engineering*. 2021. Т. 13, № 4. С. 16–21.
- [2] Мяо Хайфэн, Сяо Вэйминь, Жуань Шаочуань. Предварительное исследование гомоцистеина у пациентов с инсультом в разных возрастных группах // *Журнал практической неврологии Китая*. 2008. № 9. С. 31–33.
- [3] Ши Наньцян, Лю Ганфэн, Чжэн Тяньцзяо и др. Прогресс исследований и клиническое применение роботов для реабилитации нижних конечностей // *Информация и управление*. 2021. Т. 50, № 1. С. 43–53.
- [4] Лю Цань. Исследование развития системы страхования на долгосрочный уход в Китае: дис. ... магистра наук. Пекин: Пекинский транспортный университет, 2019.
- [5] Ван Фэн, Ли Цзинлун. Прогресс исследований применения экзоскелетных роботов для нижних конечностей при травмах спинного мозга // *Журнал практической медицины*. 2022. Т. 38, № 33. С. 3012–3016.
- [6] HAMIDA I. B., LARIBI M. A., MLIKA A., и др. Многокритериальная оптимизация конструкции параллельного робота с тросовым приводом для реабилитационных задач // *Mechanism and Machine Theory*. 2021. Т. 156.
- [7] Хоу Цзэнгуан, Чжао Синьган, Чэн Лун и др. Прогресс исследований в области реабилитационных роботов и интеллектуальных вспомогательных систем // *Журнал автоматизации*. 2016. Т. 42, № 12. С. 1765–1779.
- [8] Чжоу Бинь, Лю Сяюй, Юй Пэн. Применение робота для реабилитации нарушений равновесия в послеоперационной реабилитации пожилых пациентов с переломами нижних конечностей // *Журнал китайской реабилитационной медицины*. 2021. Т. 36, № 5. С. 533–538.
- [9] Комитет по реабилитационному лечению Китайской ассоциации реабилитационной медицины. Экспертное соглашение по клиническому

применению роботов для реабилитации нижних конечностей // Журнал китайской реабилитационной медицины. 2022. Т. 37, № 5. С. 517–525.

[10] У Сьжюань, Ли Куньбинь, Лоу Шувэй и др. Влияние тренировки с использованием робота для реабилитации нижних конечностей на двигательную активность и функции равновесия у пациентов с инсультом // Журнал реабилитации. 2020. Т. 30, № 2. С. 114–118.

[11] SHENG P., HUA-YUN Z., YONG-LAN X. U. и др. Наблюдение за влиянием тренировки с использованием реабилитационного робота на двигательную функцию нижних конечностей у пациентов с гемиплегией после церебрального ишемического инсульта // China Medical Equipment. 2019. Т. 16, № 6. С. 90–93.

[12] Чжоу Чжиён, Цянь Вэй, Дин Цзятао и др. Онлайн-планирование траектории шага экзоскелетного коленного сустава на основе атомарных движений // Робототехника. 2021. Т. 43, № 5. С. 557–566.

[13] Чжан Цзе, Ли Хуа, Ван Цзяньго. Исследование конструкции и стратегии управления роботом для реабилитации нижних конечностей // Журнал китайской реабилитационной медицины. 2020. Т. 35, № 6. С. 589–595.

[14] Brown J., Smith K., Johnson T. Прогресс в области реабилитационных роботов для нижних конечностей: проектирование и стратегии управления // International Journal of Rehabilitation Robotics. 2021. Т. 12, № 3. С. 203–215.

[15] Molteni F., Gasperini G., Cannaviello G. и др. Экзоскелетные и концевые роботы для реабилитации верхних и нижних конечностей: обзор // PM&R. 2018. Т. 10, № 9. С. S174–S188.

[16] Чэн Хун, Хуан Жуй, Цю Цзин и др. Обзор реабилитационных роботов и их клинического применения // Робототехника. 2021. Т. 43, № 5. С. 606–619.

[17] Чжао Пэн, Ван Цянь, Ли Вэньцзюнь. Состояние исследований и тенденции развития роботов для реабилитации нижних конечностей в Китае и за рубежом // Журнал китайской реабилитационной медицины. 2021. Т. 36, № 3. С. 231–238.

- [18] Лю Цзе, Чэнь Миньхуэй, Ван Юцзе. Анализ прогресса разработки роботов для реабилитации нижних конечностей в сотрудничестве университетов и предприятий // Робототехнические технологии и приложения. 2020. Т. 35, № 4. С. 15–21.
- [19] Manbuzhe Rehabilitation (MBZ). Продуктовый обзор GR-A1: робот для реабилитации нижних конечностей [ЕВ/ОЛ]. 06 марта 2023. Режим доступа: <http://www.manbuzhekangfu.com/pd.jsp?id=34#pp=103488>.
- [20] Fourier Intelligent Technology Co., Ltd. EхоMotus: технические характеристики и применение экзоскелетного робота для реабилитации нижних конечностей [R]. Шанхай, 2019.
- [21] Гуанчжоу Икан Медицинские Оборудования и Промышленность. Технический белый документ по системе тренировки и оценки походки серии А3 [R]. Гуанчжоу, 2023.
- [22] Wang J., Li X., Zhang Y. Прогресс в роботах для реабилитации походки в Китае // Journal of Rehabilitation Engineering. 2023. Т. 15, № 2. С. 102–115. DOI: 10.1016/j.rehabeng.2023.02.003.
- [23] Гуанчжоу Икан Медицинские Оборудования и Промышленность. А3 [ЕВ/ОЛ]. (05 марта 2022). Режим доступа: <http://www.yikangshiye.com/legs/a3.html>.
- [24] Хан Цзюйюй, Ван Яньхон, Ван Дачянь. Прогресс исследований и тенденции развития экзоскелетных роботов для реабилитации нижних конечностей // Журнал Шанхайского транспортного университета. 2022. Т. 42, № 2. С. 241–246.
- [25] Bourj M., Clavel R., Bolliger M. MotionMaker: Реабилитационный робот для терапии нижних конечностей // IEEE Transactions on Robotics. 2012. Т. 28, № 5. С. 1142–1152. DOI: 10.1109/TRO.2012.2193538.

- [26] Popovic M. R., Sinkjaer T. Клиническая оценка робота MotionMaker в терапии реабилитации походки // *Journal of NeuroEngineering and Rehabilitation*. 2013. Т. 10, № 1. С. 45–52.
- [27] Hocoma AG. Lokomat: Роботизированная терапия для нейрореабилитации [R]. Hocoma Technical White Paper, 2019. Доступно на: www.hocoma.com.
- [28] Koller J. R., Dixon L., Friedman N. Виртуальная среда в тренировках Lokomat: улучшение вовлеченности пациентов // *Journal of NeuroEngineering and Rehabilitation*. 2017. Т. 14, № 1. С. 88–99. DOI: 10.1186/s12984-017-0298-2.
- [29] Vallery H., Duschau-Wicke A., Riener R. Адаптивное управление Lokomat для улучшенной взаимодействия с пациентом // *IEEE Transactions on Robotics*. 2009. Т. 25, № 3. С. 635–644.
- [30] Westlake K. P., Patten C. Пилотное исследование Lokomat против тренировки на тренажере с ручной помощью для восстановления движений после инсульта // *Journal of Neuroengineering and Rehabilitation*. 2009. Т. 6. С. 1–11.
- [31] Эргашева Б. И. Экзоскелеты для нижних конечностей: краткий обзор // *Научно-технический вестник информационных технологий, механики и оптики*. 2017. Т. 17, № 6. С. 1153–1158.
- [32] Люненбургер Л., Коломбо Г., Рейнер Р. и др. Биофидбек в тренировках походки с роботизированным ортезом Lokomat // 26-я Ежегодная международная конференция IEEE по инженерии в медицине и биологии. IEEE, 2004. Т. 2. С. 4888–4891.
- [33] Стадлер К. С., Эльспасс В. Дж., ван де Венн Х. В. Robo-Mate: экзоскелет для повышения производительности в промышленности: специальная сессия: экзоскелеты для новых приложений // *Mobile Service Robotics*. 2014. С. 53–60.
- [34] Раб К., Краков К., Трипп Ф. и др. Влияние тренировки с экзоскелетом ReWalk на качество жизни при неполном повреждении спинного мозга: одно исследование // *Spinal Cord Series and Cases*. 2016. Т. 2, № 1. С. 1–3.

- [35] Квон С. Х., Ли Б. С., Ли Х. Дж. и др. Энергоэффективность и удовлетворенность пациентов при походе с ортезом колена-голеностопный-стопа и с роботизированной (ReWalk) помощью при повреждениях спинного мозга // *Annals of Rehabilitation Medicine*. 2020. Т. 44, № 2. С. 131.
- [36] Ёсикава К., Мутсузаки Х., Сано А. и др. Тренировка с гибридным вспомогательным устройством для восстановления функции ходьбы после тотальной артропластики колена // *Journal of Orthopaedic Surgery and Research*. 2018. Т. 13. С. 1–10.
- [37] Талати М., Эсквинази А., Брисено Ж. Э. Дифференциация способностей пользователей экзоскелета ReWalk ТМ: анализ кинематики ходьбы // 13-я международная конференция IEEE по реабилитационной робототехнике. США, 2013. С. 1–5.
- [38] *Journal of Robotics and Mechatronics*. 2005. Т. 17, № 2. С. 241–252.
- [39] Санкай Я. Cyberdyne и HAL: проблемы внедрения носимой робототехники в повседневную жизнь // *Proceedings of the IEEE International Conference on Robotics and Automation*. 2013. С. 167–174.
- [40] Зосс А. Б., Казерони Х., Чу А. Биомеханический дизайн экзоскелета для нижних конечностей Berkeley (BLEEX) // *IEEE/ASME Transactions on Mechatronics*. 2006. Т. 11, № 2. С. 128–138.
- [41] Rosen J., Fuchs M. B., Arcan M. Performances of Hill-type and neural network muscle models—toward a myosignal-based exoskeleton // *Computers and Biomedical Research*. 1999. Т. 32, № 5. С. 415–439.
- [42] Rosen J., Brand M., Fuchs M. B. и др. A myosignal-based powered exoskeleton system // *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*. 2001. Т. 31, № 3. С. 210–222.

- [43] Nef T., Guidali M., Riener R. ARMin III – arm therapy exoskeleton with an ergonomic shoulder actuation // *Applied Bionics and Biomechanics*. 2009. Т. 6, № 2. С. 127–142.
- [44] Carignan C., Tang J., Roderick S. Development of an exoskeleton haptic interface for virtual task training // 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009. С. 3697–3702.
- [45] Hochberg L. R., Bacher D., Jarosiewicz B. и др. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm // *Nature*. 2012. Т. 485, № 7398. С. 372–375.
- [46] Chan A. D. C., Englehart K. B. Continuous myoelectric control for powered prostheses using hidden Markov models // *IEEE Transactions on Biomedical Engineering*. 2004. Т. 52, № 1. С. 121–124.
- [47] Chu J. U., Moon I., Lee Y. J. и др. A supervised feature-projection-based real-time EMG pattern recognition for multifunction myoelectric hand control // *IEEE/ASME Transactions on Mechatronics*. 2007. Т. 12, № 3. С. 282–290.
- [48] Cavallaro E. E., Rosen J., Perry J. C. и др. Real-time myoprocessors for a neural controlled powered exoskeleton arm // *IEEE Transactions on Biomedical Engineering*. 2006. Т. 53, № 11. С. 2387–2396.
- [49] Artemiadis P. K., Kyriakopoulos K. J. EMG-based control of a robot arm using low-dimensional embeddings // *IEEE Transactions on Robotics*. 2010. Т. 26, № 2. С. 393–398.
- [50] Го Шицзе, Сун Юаньхао, Ван Сюшэн, Лю Цзоцзюнь, Ли Ян. Методы обучения и переноса стратегии активной реабилитации робота верхних конечностей // *Робототехника*. 2024. Т. 46, № 5. С. 45–58.
- [51] Е Чанлун, Пэн Цзиньсинь, Юй Суюя, Цзянь Чуньин. Метод адаптивного управления гибкостью для настольного реабилитационного робота верхних конечностей // *Журнал Шэньянского аэрокосмического университета*. 2024. Т. 41, № 5. С. 62–71.

- [52] Xu G., Wu Y., Zhang L. Advanced Materials for Rehabilitation Robots: Current Trends and Future Perspectives // Journal of Biomedical Materials Research Part B: Applied Biomaterials. 2023. Т. 111, № 4. С. 587–601.
- [53] Chen H., Li Z., Gao Y. Personalized Adaptation in Rehabilitation Robotics: Challenges and Opportunities // IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2022. Т. 30, № 6. С. 983–994.
- [54] Wang J., Zhang T. Safety and Comfort Considerations in Exoskeleton Design for Rehabilitation // Journal of Rehabilitation Engineering. 2023. Т. 45, № 1. С. 101–110.
- [55] Liu Q., Zhao H. Latency Challenges in Control Systems of Rehabilitation Robots // IEEE Robotics and Automation Letters. 2023. Т. 8, № 2. С. 1583–1590.
- [56] Чжэн Тяньцзяо. Разработка лёгкого экзоскелета для ходьбы и метод генерации походки: дис. ... магистра наук. Харбинский промышленный университет, 2020.
- [57] Ли Ваньтин. Исследование восприятия информации и координационного управления человеко-машинным взаимодействием в жёстко-мягком сочленённом экзоскелете нижних конечностей: дис. ... магистра наук. Чанчуньский промышленный университет, 2022.
- [58] Ши Сюйян, Ван Циньсянь, Ли Чэнфэй, и др. Прогресс исследований прикладной анатомии и клиники капсулы тазобедренного сустава // Журнал китайской анатомии и клинической медицины. 2023. Т. 28, № 1. С. 45–50.
- [59] Сюй Фэн, Юань Кайсюань, Чэнь Шэнбао, Чжан Чанцин. Прогресс исследований системы оценки функции тазобедренного сустава // Международный журнал ортопедии. 2018. Т. 39, № 3. С. 185–189.
- [60] Delp S. L., Anderson F. C., Arnold A. S. и др. OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement // IEEE Transactions on Biomedical Engineering. 2007. Т. 54, № 11. С. 1940–1950.

- [61] Seth A., Hicks J. L., Uchida T. K. и др. OpenSim: Simulating Musculoskeletal Dynamics and Neuromuscular Control to Study Human and Animal Movement // PLoS Computational Biology. 2018. Т. 14, № 7. С. e1006223.
- [62] Seth A., Hicks J. L., Uchida T. K. и др. OpenSim: Simulating Musculoskeletal Dynamics and Neuromuscular Control to Study Human and Animal Movement // PLoS Computational Biology. 2018. Т. 14, № 7. С. e1006223. DOI: 10.1371/journal.pcbi.1006223.
- [63] Mei Q., Gu Y., Fernandez J. A Review on the Application of Image-Based Subject-Specific OpenSim Lower Extremity Musculoskeletal Models into ...
- [64] Цзи Синьюй. Исследование стратегии интеллектуального управления поясничным экзоскелетом на основе распознавания движений: дис. ... магистра наук. Уханьский технологический университет, 2021. С. 39–45.
- [65] Дин Ивэй, Шуай Мэй, Чжан Цзицун. Прогресс исследований носимого экзоскелета для реабилитации нижних конечностей // Робототехника. 2022. Т. 44, № 5. С. 523–533.
- [66] Ван Вэй. Проектирование и исследование кинематических характеристик носимого экзоскелета для реабилитации нижних конечностей: магистерская диссертация. Пекинский авиационно-космический университет, 2021.
- [67] Ryu J., Lee B. H., Kim D. H. sEMG signal-based lower limb human motion detection using atop and slope feature extraction algorithm // IEEE Signal Processing Letters. 2016. Т. 24, № 7. С. 929–932.
- [68] Инь Циюн. Эффект снижения веса при тренировке походки в сочетании с контролем коленного сустава на реабилитацию гиперэкстензии колена после инсульта: дис. ... магистра наук. Аньхойский медицинский университет, 2024.
- [69] Сунь Юэцюнь. Анализ применения метода конечных элементов в современной машиностроительной инженерии // Китайская научно-техническая информация. 2021. № 17. С. 138–140.

- [70] Юй Юнсюнь. Текущее состояние применения ANSYS Workbench и анализ // Технология мехатроники и инжиниринга. 2014. № 9. С. 138–141.
- [71] Чжэн Фуцзянь, Чан Ин, Чжао Юшань и др. Анализ методом конечных элементов робота для реабилитации нижних конечностей на базе ANSYS Workbench // Сельскохозяйственные машины Хэбэя. 2020. № 5. С. 45–49.
- [72] Чэнь Голяном. Проектирование структуры и моделирование робота для реабилитации нижних конечностей: дис. ... магистра наук. Чанчуньский промышленный университет, 2023. С. 65–78.
- [73] Лю Кунь, Цзи Шуо, Сунь Чжэньюань и др. Бионическое проектирование конструкции и оптимизация реабилитационного робота для двигательной реабилитации // Проектирование и производство машин. 2018. № 7. С. 59–63.
- [74] Li X., Zhong J., An S., Huang Y. A New Variable-Stiffness Body Weight Support System Driven by Two Active Closed-Loop Controlled Drives // Actuators. 2024. Т. 13, № 8. С. 304.
- [75] Zhang J., Zhu A., Ma P. и др. Design and Evaluation of an Integrated Active and Passive Body Weight Support System for Variable Gravity and Inertia Compensation // IEEE/ASME Transactions on Mechatronics. 2024.
- [76] Dong Z., Luces J. V. S., Hirata Y. Control and Evaluation of a Ground Gait Training Walker with Body Weight Support // IEEE Robotics and Automation Letters. 2021. Т. 6, № 2. С. 3617–3624.
- [77] Gholap R., Thorat S., Chavan A. Review of current developments in lower extremity exoskeleton systems // Materials Today: Proceedings. 2023. Т. 72. С. 817–823.
- [78] Чжоу Шухуа, Чжан Вэньхуэй, Вэнь Чжи, Е Сяопин. Кинематическое моделирование и планирование траектории для декартова робота на основе параметров D–H // Электротехнические технологии. 2020. № 24. С. 128–134.

- [79] Е Цюань, Ван Цзяньхуа. Кинематический анализ и моделирование робота ABB IRB2600 на основе параметров D–H // Современная производственная инженерия. 2020. № 5. С. 45–49.
- [80] Чэнь Вэй, Ван Личжу, Чжан Линьян, Ма Ли. Динамический анализ и моделирование экзоскелета для реабилитации нижних конечностей // Машиностроительное проектирование. 2018. Т. 35, № 4. С. 71–77.
- [81] Лю Сяолун, Чжао Яньцзюнь, Гэ Вэньцин, Ван Ин, Чжан Чжудун. Проектирование медицинского вспомогательного экзоскелета нижних конечностей и его динамический анализ // Журнал инженерного проектирования. 2016. Т. 23, № 4. С. 27–32.
- [82] Чжу Юди, Юй Хунлю. Метод управления роботом для реабилитации нижних конечностей на основе биомеханической модели человека // Прогресс в биомедицинской инженерии. 2023. Т. 44, № 3. С. 294–300.
- [83] Цуй Миньци. Лагранжева динамическая модель SCARA-робота // Проектирование и производство машин. 2012. № 6. С. 56–58.
- [84] Ван Ган, Лю Яньцзе, У Миньюэ, Хань Хайцзюнь. Упрощённое твердотельное моделирование динамики Delta-параллельного робота на основе метода умножителей Лагранжа // Механика и электроника. 2015. № 7. С. 68–72, 76.
- [85] Фан Уи, Го Сюнь, Ли Лян, Чжан Динго. Динамическое моделирование, симуляция и управление роботом с гибкими шарнирами и гибкими звеньями // Журнал механики. 2021. Т. 53, № 5. С. 1234–1245.
- [86] Лю Нин, Чай Тянью. Метод оптимальной настройки параметров PID-контроллера // Журнал автоматизации. 2023. Т. 49, № 11. С. 2272–2285.
- [87] Ху Ин, Лю Чан, Юань Цзячэнь. Поиск оптимальных параметров главного и подчинённого PID-контроллера на основе улучшенного алгоритма MOEAD // Исследование в области компьютерных приложений. 2024. Т. 41, № 5. С. 1434–1440.

- [88] Ван Вэй, Ван Цзяньсинь, Ван Цзянь. Исследование оптимизации правил нечеткого PID на основе улучшенного алгоритма летучих мышей // Компьютерная инженерия и наука. 2020. Т. 42, № 8. С. 305–312.
- [89] Ли Чжиминь. Исследование нечеткого контроллера на основе самоадаптирующегося фактора оптимизации // Вестник Университета Датун. Серия: Естественные науки. 2021. Т. 37, № 2. С. 15–16, 35.
- [90] Сяо Чжэ. Исследование и реализация методов калибровки камеры и 3D-реконструкции в системе оптического захвата движения // Технология электронных измерений. 2023. Т. 41, № 3. С. 120–125.
- [91] Zhang Y., Xu L., Sun G. Система захвата движений на основе слияния данных RGB-D-камеры и IMU // Датчики и микросистемы. 2023. Т. 42, № 2. С. 55–60.
- [93] Чжэн Бинь. Исследование алгоритма вычисления положения буксируемого кабеля на основе сверточной нейронной сети и расширенного фильтра Калмана: дис. ... магистра наук. Гиринский университет, 2022.
- [94] Ма Тяньли, Чжан Ян, Гао Сун и др. IMM-адаптивная фильтрация для систем с шумовой информацией и неопределённостью в модели состояния // Контроль и принятие решений. 2024. Т. 39, № 5. С. 1604–1611.
- [95] Цю Жунь, Ли Цзиньтао, Ли Сяоцян и др. Прогнозирование траектории на основе алгоритма фильтра Калмана // Проектирование электронной техники. 2022. Т. 30, № 6. С. 24–28.
- [96] Лю Сяодун, Цай Сичан. Проектирование и анализ производительности цифрового FIR-фильтра реального времени на базе STM32H7 // Закупки компьютерной техники. 2023. № 44. С. 68–70.
- [97] Юй Сюмин, Фэн Цюаньюань. Проектирование линейного FIR-фильтра высокого порядка // Микроэлектроника. 2023. Т. 41, № 2. С. 130–134.
- [98] Чэнь Юаньюань, Лю Юяо. Проектирование и реализация FIR-фильтра на базе FPGA // Проектирование в электронике. 2017. Т. 24. С. 65–69, 73.

- [99] Ян Хунъю, Дин Пэнчэн, Ван Цюаньчжоу. Проектирование высокоскоростного полностью параллельного FIR-фильтра на базе FPGA // Вестник Северо-Западного педагогического университета. Серия: Естественные науки. 2012. Т. 48, № 1. С. 124–126, 225.
- [100] Гу Чжывэй, Ли Ли, Фу Чжуаньчжан, Фу Юсян, Ли Вэй. Параллельная реализация FIR-алгоритма на реконфигурируемом специализированном процессоре // Журнал электроники и информации. 2016. Т. 38, № 9. С. 2205–2211.
- [101] Цзэн Цзюйжун. Реализация высокопорядкового IIR-фильтра на базе FPGA // Проектирование в электронике. 2011. Т. 19, № 10. С. 173–175, 179.
- [102] Ван Ифань, Ван Минь, Ли Хайлун. Исследование проектирования цифрового IIR-фильтра на базе MATLAB // Мир коммуникаций. 2019. Т. 36, № 4. С. 45–49.
- [103] Liu J., Shen L., Qian G. Maximum Complex Entropy Criterion Adaptive IIR Filter Based on Gauss-Newton Method // IEEE Circuits and Systems Letters. 2023. Т. 24, № 2. С. 153–160.
- [104] Лян Янь, Лин Юнцюань. Проектирование оптимального эквалайзера на базе вейвлет-декомпозиции и IIR-фильтра // Телевизионные технологии. 2021. Т. 45, № 3. С. 89–93, 99.

Приложение 1

(1) The Matlab PID control simulation code is as follows:

```
m = 25;
g = 9.81;
F0 = 0.0;
K_p = 20;
K_i = 0.1;
K_d = 50;
Y = 2;
dt = 0.01;
t = 0:dt:10;
x = zeros(size(t));
v = zeros(size(t));
F_e = zeros(size(t));
F = zeros(size(t));
xd = zeros(size(t));
xd_dot = zeros(size(t));
v_dot = zeros(size(t));
e = zeros(size(t));
e_sum = 0;
amplitude = 10;
frequency = 1;
```



```

F_e = amplitude * sin(2 * pi * frequency * t);

for i = 1:length(t)

    xd(i) = Y * F_e(i);

end

for i = 2:length(t)

    xd_dot(i) = (xd(i) - xd(i - 1)) / dt;

v_dot(i) = (v(i) - v(i - 1)) / dt;

end

for i = 2:length(t)

    e(i) = xd(i) - x(i - 1);

    e_sum = e_sum + e(i);

    u = K_p * e(i) + K_i * e_sum * dt + K_d * (e(i) - e(i - 1)) / dt;

    F(i) = m * (g + v_dot(i)) + F0 + u;

    v(i) = v(i - 1) + (F(i) / m - g) * dt;

    x(i) = x(i - 1) + v(i) * dt;

end

figure;

subplot(4,1,1);

plot(t, F_e, 'b', 'DisplayName', 'External Force (F_e)');

xlabel('Time (s)');

ylabel('Force (N)');

legend;

```

```
grid on;

subplot(4,1,2);

plot(t, F, 'r', 'DisplayName', 'Motor Driving Force (F)');

xlabel('Time (s)');

ylabel('Force (N)');

legend;

grid on;

subplot(4,1,3);

hold on;

plot(t, xd, 'b', 'DisplayName', 'Desired Position (x_d)');

plot(t, x, 'r', 'DisplayName', 'Actual Position (x)');

xlabel('Time (s)');

ylabel('Position (m)');

legend;

grid on;

subplot(4,1,4);

plot(t, v, 'b', 'DisplayName', 'Velocity (v)');

xlabel('Time (s)');

ylabel('Velocity (m/s)');

legend;

grid on;
```

(2) Matlab Симуляция управления скользящим режимом в Matlab

представлена следующим кодом:

```
m = 25;
g = 9.81;
F0 = 5;
lambda = 50;
k = 1;
kp = 2;
ke = 1;
tspan = 0:0.01:10;
dt = tspan(2)-tspan(1);
x = zeros(size(tspan));
v = zeros(size(tspan));
F = zeros(size(tspan));
Fe = zeros(size(tspan));
xd = zeros(size(tspan));
for i = 2:length(tspan)
    Fe(i) = 5*sin(tspan(i));
    xd(i) = kp * Fe(i);
    e = xd(i) - x(i-1);
    de = (xd(i) - xd(i-1))/dt - v(i-1);
    Feq = m * (ke * de + lambda * de) + F0 + m * g;
```

```

Fsw = -k * sign(e + lambda * de);

F(i) = Feq + Fsw;

a = (F(i) - F0 - m * g) / m;

v(i) = v(i-1) + a * dt;

x(i) = x(i-1) + v(i) * dt;

end

figure;

subplot(3,1,1);

plot(tspan, Fe);

title('External Force Fe');

xlabel('Time (s)');

ylabel('Force (N)');

subplot(3,1,2);

hold on;

plot(tspan, xd, 'r--');

plot(tspan, x, 'b');

hold off;

legend('Desired Position x_d', 'Actual Position x');

title('Position');

xlabel('Time (s)');

ylabel('Position (m)');

subplot(3,1,3);

```

```
plot(tspan, F);  
title('Control Force F');  
xlabel('Time (s)');  
ylabel('Force (N)');  
ylim([-30, 500]);  
legend;  
grid on;
```

Приложение 2

Код программы управления на C++

```
// zlgcanDlg.h :
#pragma once
#include "afxwin.h"
#include "zlgcan/zlgcan.h"
#include "afxcmn.h"
#pragma comment(lib,"Pressure.lib")
extern "C" _declspec(dllimport) void InitPowerDll();
extern "C" _declspec(dllimport) void GetPowerData(double* a, double*
b, double* c, double* d);

#pragma comment(lib,"RS232.lib")
extern "C" _declspec(dllimport) int OpenRS232(HANDLE * pHandle);
extern "C" _declspec(dllimport) int ReadRS232aFile(HANDLE hSerial);
DWORD WINAPI ThreadProc(LPVOID lpParam);
class CZlgcanDlg : public CDialog
{
public:
    CZlgcanDlg(CWnd* pParent = NULL);
    enum { IDD = IDD_ZLGCANDEMO_DIALOG };
protected:
```

```

    virtual void DoDataExchange(CDataExchange* pDX);    void
OnRecv();

protected:

    HICON m_hIcon;

    virtual BOOL OnInitDialog();

    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);

    afx_msg void OnPaint();

    afx_msg HCURSOR OnQueryDragIcon();

    afx_msg void OnClose();

    afx_msg void OnCbnSelchangeComboDevice();

    afx_msg void OnCbnSelchangeComboNetMode();

    afx_msg void OnBnClickedButtonOpen();

    afx_msg void OnBnClickedButtonInitcan();

    afx_msg void OnBnClickedButtonStartcan();

    afx_msg void OnBnClickedButtonReset();

    afx_msg void OnBnClickedButtonClose();

    afx_msg void OnBnClickedButtonSend();

    afx_msg void OnBnClickedButtonClear();

    afx_msg void OnBnClickedButtonGetTxAvailable();

    afx_msg void OnBnClickedButtonClearTxQueue();

    afx_msg void OnBnClickedCheckDelaySendMode();

    afx_msg void OnBnClickedButtonGetSendMode();

```

```
afx_msg void OnBnClickedButtonAutosendStart();  
afx_msg void OnBnClickedButtonAutosendStop();  
afx_msg void OnBnClickedButtonStopSingleAutosend();  
afx_msg void OnBnClickedButtonAddAutosend();
```

```
DECLARE_MESSAGE_MAP()
```

```
private:
```

```
void InitCombobox(int ctrl_id, int start, int end, int current);  
void EnableCtrl(BOOL opened);  
BOOL SetCustomBaudrate();  
BOOL SetTransmitType();  
    BOOL SetResistanceEnable();  
    BOOL SetBaudrate();  
BOOL SetCanfdBaudrate();  
static UINT WINAPI OnDataRecv(LPVOID data);  
void AddData(const ZCAN_Receive_Data* data, UINT len);  
void AddData(const ZCAN_ReceiveFD_Data* data, UINT len);  
    void AddData(const ZCAN_Transmit_Data* data, UINT len);  
    void AddData(const ZCAN_TransmitFD_Data* data, UINT len);  
void AddData(const CString& data);  
    void    SetCtrlStateDelaySend(BOOL    delay_send,    BOOL  
delay_send_mode, BOOL get_send_mode);
```



```

    BOOL IsNetCAN(UINT type );

    BOOL IsNetCANFD(UINT type );

    BOOL IsNetTCP(UINT type );

    BOOL IsNetUDP(UINT type );

    void      SetAutoSendCtrlState(BOOL      support_can,      BOOL
support_canfd,  BOOL  support_index,  BOOL  support_single_cancel,
BOOL support_get_autosend_list);

    void GetViewCANFrame(ZCAN_Transmit_Data& can_data, BOOL
add_delay_flag);

    void  GetViewCANFrame(ZCAN_TransmitFD_Data& canfd_data,
BOOL add_delay_flag);

    void          GetZCAN_Data(CString          _id,CString
SData,ZCAN_Transmit_Data& can_data, BOOL add_delay_flag);

    void ZCANSendData(CString _id, CString SData);

    void AddAutosendCan(int nEnable);

    void AddAutosendCanfd(int nEnable);

    void DecToHex(long int x, char* p);

    int ConvertH_To_L(char* source, char* dest);

    void FConvertH_To_LNoSpace(char* source, char* dest);

    void ConvertL_To_H(char* source, char* dest);

    long int Daonagongshi(long int v_n_1, double power);

```

```
void StrToHex(char* pbDest, char* pbSrc, int nLen);
```

```
private:
```

```
int device_type_index_;
```

```
int device_index_;
```

```
int channel_index_;
```

```
int work_mode_index_;
```

```
int baud_index_;
```

```
int abit_baud_index_;
```

```
int dbit_baud_index_;
```

```
BOOL custom_baud_enable_;
```

```
    BOOL resistance_enable_;
```

```
int frame_type_index_;
```

```
int protocol_index_;
```

```
int canfd_exp_index_;
```

```
int send_type_index_;
```

```
    int net_mode_index_;
```

```
CListBox data_recv_list_;
```

```
CString acc_code_;
```

```
CString acc_mask_;
```

```
CString id_;
```

```
CString datas_;
```

```
CString custom_baudrate_;
```

```
CComboBox ctrl_device_type_;
CComboBox ctrl_device_index_;
CComboBox ctrl_channel_index_;
DEVICE_HANDLE device_handle_;
CHANNEL_HANDLE channel_handle_;
IProperty* property_;
BOOL start_;
CButton ctrl_open_device_;
CButton ctrl_int_can_;
CButton ctrl_start_can_;
CButton ctrl_close_device_;
BOOL device_opened_;
int filter_mode_;

CComboBox m_comboAbit;
CComboBox m_comboDbit;
int send_count_once_;
int frm_delay_time_;
BOOL frm_delay_flag_;
CButton ctrl_delay_send_mode_;

BOOL support_delay_send_;

BOOL support_delay_send_mode_;

BOOL support_get_send_mode_;
```

```

    int auto_send_index_;

    int auto_send_period_;

    CSpinButtonCtrl m_spinAutSendIndex;

    CEdit m_editAutoSendIndex;

    double m_data[4];

    int m_count;

    //long int v_n_1, v_n;

public:

    /*HANDLE m_Handle;

    int m_iRS232data;

    int m_iRS232Predata;

    */

public:

    afx_msg void OnBnClickedButtonGetDevAutosend();

    afx_msg void OnCbnSelchangeComboDeviceIndex();

    afx_msg void OnTimer(UINT_PTR nIDEvent);

    CListBox m_list2;

protected:

    void StartRS232Thread();

public:

    int InitRS232();

```

```
afx_msg void OnBnClickedBntSet();  
  
double m_B;  
  
double m_M;  
  
};
```

```
// zlgcanDlg.cpp

#include "stdafx.h"

#include <process.h>

#include "zlgcanDemo.h"

#include "zlgcanDlg.h"

#include "zlgcan/zlgcan.h"

#include "utility.h"

#include "zcloudDlg.h"

#include "zlgcan/KalmanFilter.h"

#include <fstream>

#include <iostream>

using namespace std;

#ifdef _DEBUG

#define new DEBUG_NEW

#endif

#define TMP_BUFFER_LEN      1000

HANDLE m_RS232handle;

int m_iRS232data;

int m_iRS232Predata;

char strRecive[100];

char strDestRecive[100];
```

```

ZCAN_Receive_Data g_can_602data[100];

ZCAN_Receive_Data g_can_603data[100];

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

    enum { IDD = IDD_ABOUTBOX };

protected:
    virtual void DoDataExchange(CDataExchange* pDX);

protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
}

```

```

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)

END_MESSAGE_MAP()

// Cusbcanfdx00udemoDlg

CZlgcanDlg::CZlgcanDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CZlgcanDlg::IDD, pParent)
    , device_type_index_(3)
    , device_index_(0)
    , channel_index_(0)
    , work_mode_index_(0)
    , abit_baud_index_(0)
    , dbit_baud_index_(0)
    , custom_baud_enable_(FALSE)
    , resistance_enable_(TRUE)
    , frame_type_index_(0)
    , protocol_index_(0)
    , canfd_exp_index_(1)
    , send_type_index_(0)
    , acc_code_( _T("00000000"))
    , acc_mask_( _T("FFFFFFFF"))
    , id_( _T("602"))
    , datas_( _T(""))
    , custom_baudrate_( _T(""))

```



```

, filter_mode_(1)

, net_mode_index_(0)

, baud_index_(2)

, send_count_once_(1)

, frm_delay_time_(1000)

, frm_delay_flag_(FALSE)

, auto_send_index_(0)

, auto_send_period_(1000)

, m_B(60)

, m_M(10)

{

//00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63

m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);

device_opened_ = FALSE;

start_ = FALSE;

property_ = NULL;

support_delay_send_ = FALSE;

support_delay_send_mode_ = FALSE;

support_get_send_mode_ = FALSE;

m_iRS232data = 0;

```

```

        m_iRS232Predata = 0;
    }

void CZlgcanDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    DDX_CBIndex(pDX,          IDC_COMBO_DEVICE,
device_type_index_);
    DDX_CBIndex(pDX,          IDC_COMBO_DEVICE_INDEX,
device_index_);
    DDX_CBIndex(pDX,          IDC_COMBO_CHANNEL_INDEX,
channel_index_);
    DDX_CBIndex(pDX,          IDC_COMBO_DEVICE_INDEX2,
work_mode_index_);
    DDX_CBIndex(pDX, IDC_COMBO_ABIT, abit_baud_index_);
    DDX_CBIndex(pDX, IDC_COMBO_ABIT2, dbit_baud_index_);
    DDX_CBIndex(pDX, IDC_COMBO_BAUD, baud_index_);
    DDX_Check(pDX,          IDC_CHECK_CUSTOM_BAUDRATE,
custom_baud_enable_);
    DDX_Check(pDX,          IDC_CHECK_RESISTANCE,
resistance_enable_);
    DDX_CBIndex(pDX,          IDC_COMBO_FRAME_TYPE,

```

```

frame_type_index_);

    DDX_CBIndex(pDX,                IDC_COMBO_PROTOCOL,
protocol_index_);

    DDX_CBIndex(pDX,                IDC_COMBO_CANFD_EXP,
canfd_exp_index_);

    DDX_CBIndex(pDX,                IDC_COMBO_SEND_TYPE,
send_type_index_);

    DDX_Control(pDX, IDC_LIST1, data_recv_list_);

    DDX_Text(pDX, IDC_EDIT_ACC_CODE, acc_code_);

    DDV_MaxChars(pDX, acc_code_, 8);

    DDX_Text(pDX, IDC_EDIT_ACC_MASK, acc_mask_);

    DDV_MaxChars(pDX, acc_mask_, 8);

    DDX_Text(pDX, IDC_EDIT_FILTER_START2, id_);

    DDV_MaxChars(pDX, id_, 8);

    DDX_Text(pDX, IDC_EDIT_FILTER_START3, datas_);

    DDX_Control(pDX, IDC_COMBO_DEVICE, ctrl_device_type_);

    DDX_Control(pDX,                IDC_COMBO_DEVICE_INDEX,
ctrl_device_index_);

    DDX_Control(pDX,                IDC_COMBO_CHANNEL_INDEX,
ctrl_channel_index_);

    DDX_Text(pDX,                IDC_EDIT_CUSTOM_BAUDRATE,
custom_baudrate_);

```

```

DDX_Control(pDX, IDC_BUTTON_OPEN, ctrl_open_device_);
DDX_Control(pDX, IDC_BUTTON_INITCAN, ctrl_int_can_);
DDX_Control(pDX, IDC_BUTTON_STARTCAN, ctrl_start_can_);
DDX_Control(pDX, IDC_BUTTON_CLOSE, ctrl_close_device_);
DDX_CBIndex(pDX,          IDC_COMBO_FILTER_MODE,
filter_mode_);
DDX_CBIndex(pDX,          IDC_COMBO_NET_MODE,
net_mode_index_);
DDX_Control(pDX, IDC_COMBO_ABIT, m_comboAbit);
DDX_Control(pDX, IDC_COMBO_ABIT2, m_comboDbit);
DDX_Text(pDX, IDC_EDIT_SEND_ONCE, send_count_once_);
DDX_Text(pDX, IDC_EDIT_TIME, frm_delay_time_);
DDV_MinMaxInt(pDX, frm_delay_time_, 0, 65535);
DDX_Check(pDX,          IDC_CHECK_DELAY_SEND_FLAG,
frm_delay_flag_);
DDV_MinMaxInt(pDX, send_count_once_, 1, 10000);
DDX_Control(pDX,          IDC_CHECK_DELAY_SEND_MODE,
ctrl_delay_send_mode_);
DDX_Text(pDX,          IDC_EDIT_AUTOSEND_INDEX,
auto_send_index_);
DDX_Text(pDX,          IDC_EDIT_AUTOSEND_TIME_PERIOD,
auto_send_period_);

```

```

        DDX_Control(pDX,          IDC_SPIN_AUTOSEND_INDEX,
m_spinAutSendIndex);

        DDX_Control(pDX,          IDC_EDIT_AUTOSEND_INDEX,
m_editAutoSendIndex);

        DDX_Control(pDX, IDC_LIST2, m_list2);

        DDX_Text(pDX, IDC_EDIT_B, m_B);

        DDX_Text(pDX, IDC_EDIT_M, m_M);
    }

BEGIN_MESSAGE_MAP(CZlgcanDlg, CDialog)

    ON_WM_SYSCOMMAND()

    ON_WM_PAINT()

    ON_WM_QUERYDRAGICON()

    //}}AFX_MSG_MAP

    ON_BN_CLICKED(IDC_BUTTON_OPEN,
&CZlgcanDlg::OnBnClickedButtonOpen)

    ON_BN_CLICKED(IDC_BUTTON_INITCAN,
&CZlgcanDlg::OnBnClickedButtonInitcan)

    ON_BN_CLICKED(IDC_BUTTON_STARTCAN,
&CZlgcanDlg::OnBnClickedButtonStartcan)

    ON_BN_CLICKED(IDC_BUTTON_RESET,
&CZlgcanDlg::OnBnClickedButtonReset)

    ON_BN_CLICKED(IDC_BUTTON_CLOSE,

```

```

&CZlgcanDlg::OnBnClickedButtonClose)

    ON_BN_CLICKED(IDC_BUTTON_SEND,
&CZlgcanDlg::OnBnClickedButtonSend)

    ON_CBN_SELCHANGE(IDC_COMBO_DEVICE,
&CZlgcanDlg::OnCbnSelchangeComboDevice)

    ON_CBN_SELCHANGE(IDC_COMBO_NET_MODE,
&CZlgcanDlg::OnCbnSelchangeComboNetMode)

    ON_WM_CLOSE()

    ON_BN_CLICKED(IDC_BUTTON_CLEAR,
&CZlgcanDlg::OnBnClickedButtonClear)

    ON_BN_CLICKED(IDC_BUTTON_GET_TX_AVAILABLE,
&CZlgcanDlg::OnBnClickedButtonGetTxAvailable)

    ON_BN_CLICKED(IDC_BUTTON_CLEAR_TX_QUEUE,
&CZlgcanDlg::OnBnClickedButtonClearTxQueue)

    ON_BN_CLICKED(IDC_CHECK_DELAY_SEND_MODE,
&CZlgcanDlg::OnBnClickedCheckDelaySendMode)

    ON_BN_CLICKED(IDC_BUTTON_GET_SEND_MODE,
&CZlgcanDlg::OnBnClickedButtonGetSendMode)

    ON_BN_CLICKED(IDC_BUTTON_AUTOSEND_START,
&CZlgcanDlg::OnBnClickedButtonAutosendStart)

    ON_BN_CLICKED(IDC_BUTTON_AUTOSEND_STOP,
&CZlgcanDlg::OnBnClickedButtonAutosendStop)

```

```

        ON_BN_CLICKED(IDC_BUTTON_STOP_SINGLE_AUTOSEND,
&CZlgcanDlg::OnBnClickedButtonStopSingleAutosend)

        ON_BN_CLICKED(IDC_BUTTON_ADD_AUTOSEND,
&CZlgcanDlg::OnBnClickedButtonAddAutosend)

        ON_BN_CLICKED(IDC_BUTTON_GET_DEV_AUTOSEND,
&CZlgcanDlg::OnBnClickedButtonGetDevAutosend)

        ON_CBN_SELCHANGE(IDC_COMBO_DEVICE_INDEX,
&CZlgcanDlg::OnCbnSelchangeComboDeviceIndex)

        ON_WM_TIMER()

        ON_BN_CLICKED(IDC_BNT_SET,
&CZlgcanDlg::OnBnClickedBntSet)

END_MESSAGE_MAP()

// Cusbcanfdx00udemoDlg

BOOL CZlgcanDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // IDM_ABOUTBOX
    ASSERT((IDM_ABOUTBOX & 0xFFFF) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {

```

```

    BOOL bNameValid;

    CString strAboutMenu;

    bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);

    ASSERT(bNameValid);

    if (!strAboutMenu.IsEmpty())
    {
        pSysMenu->AppendMenu(MF_SEPARATOR);

        pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX,
strAboutMenu);
    }
}

SetIcon(m_hIcon, TRUE);

SetIcon(m_hIcon, FALSE);

// TODO: InitCombobox(IDC_COMBO_DEVICE_INDEX, 0, 32,
0);

//CComboBox*          pComboDeviceBox          =
(CComboBox*)GetDlgItem(IDC_COMBO_DEVICE);

//pComboDeviceBox->SetCurSel(3);

//CComboBox*          pComboBaudBox            =
(CComboBox*)GetDlgItem(IDC_COMBO_BAUD);

//pComboBaudBox->SetCurSel(2);

OnCbnSelchangeComboDevice();

```



```

SetDlgItemText(IDC_EDIT_LOCAL_PORT, TEXT("1080"));

SetDlgItemText(IDC_EDIT_REMOTE_ADDR,
TEXT("192.168.3.176"));

SetDlgItemText(IDC_EDIT_REMOTE_PORT, TEXT("8000"));

m_spinAutSendIndex.SetBuddy(&m_editAutoSendIndex);

m_spinAutSendIndex.SetRange(0, 32);

m_count = 0;

InitPowerDll();

//int n = OpenRS232(&m_Handle);

//CString str;

//str.Format(_T("%lf"), m_data[0]);

//m_list2.AddString(str);

//int data;

//data = ReadRS232aFile(m_Handle);

//str.Format(_T("%d"), n);

//m_list2.AddString(str);

//CreateThread(NULL, 0, ThreadProc, hTmp, 0, NULL);

int n = InitRS232();

StartRS232Thread();

SetTimer(1, 8, NULL);

return TRUE;

}

```

```

void CZlgcanDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

```

```

void CZlgcanDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this);

        SendMessage(WM_ICONERASEBKGND,
reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        int cxIcon = GetSystemMetrics(SM_CXICON);

```

```

int cyIcon = GetSystemMetrics(SM_CYICON);

CRect rect;

GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;

int y = (rect.Height() - cyIcon + 1) / 2;

dc.DrawIcon(x, y, m_hIcon);
}

else
{
    CDialog::OnPaint();
}
}

HCURSOR CZlgcanDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

typedef struct _DeviceInfo
{
    UINT device_type;

    UINT channel_count;
}

```

```

}DeviceInfo;

static const DeviceInfo kDeviceType[] = {

    {ZCAN_USBCAN1, 1},

    {ZCAN_USBCAN2, 2},

    {ZCAN_PCI9820I,2},

    {ZCAN_USBCAN_E_U, 1},

    {ZCAN_USBCAN_2E_U, 2},

    {ZCAN_USBCAN_4E_U, 4},

    {ZCAN_PCIE_CANFD_100U, 1},

    {ZCAN_PCIE_CANFD_200U, 2},

    {ZCAN_PCIE_CANFD_400U_EX, 4 },

    {ZCAN_USBCANFD_200U, 2},

    {ZCAN_USBCANFD_100U, 1},

    {ZCAN_USBCANFD_MINI, 1},

    {ZCAN_CANETTCP, 1},

    {ZCAN_CANETUDP, 1},

    {ZCAN_WIFICAN_TCP, 1},

    {ZCAN_WIFICAN_UDP, 1},

    {ZCAN_CLOUD, 1},

    {ZCAN_CANFDWIFI_TCP, 1},

    {ZCAN_CANFDWIFI_UDP, 1},

    {ZCAN_CANFDNET_TCP, 2},

```

```

    {ZCAN_CANFDNET_UDP, 2},
    {ZCAN_CANFDNET_400U_TCP, 4},
    {ZCAN_CANFDNET_400U_UDP, 4},
};

void CZlgcanDlg::OnBnClickedButtonOpen()
{
    UpdateData(TRUE);

    if (kDeviceType[device_type_index_].device_type ==
ZCAN_CLOUD)
    {
        zcloudDlg dlg;

        if (IDCANCEL == dlg.DoModal())
        {
            return;
        }

        device_index_ = dlg.GetDeviceIndex();
    }

    device_handle_ =
ZCAN_OpenDevice(kDeviceType[device_type_index_].device_type,
device_index_, 0);

    if (INVALID_DEVICE_HANDLE == device_handle_)

```

```

    {
        AddData(_T("    "));
        return;
    }

    device_opened_ = TRUE;

    EnableCtrl(TRUE);

    AddData(_T("    !"));
}

//USBCANFD

static const UINT kAbitTimingUSB[] = {

    1000000,//1Mbps

    800000,//800kbps

    500000,//500kbps

    250000,//250kbps

    125000,//125kbps

    100000,//100kbps

    50000 //50kbps

};

static const UINT kDbitTimingUSB[] = {

    5000000,//5Mbps

    4000000,//4Mbps

    2000000,//2Mbps

```

```

    1000000 //1Mbps
};

//PCIECANFD brp=1
static const UINT kAbitTimingPCIE[] = {
    1000000, //1M(80%)
    800000, //800K(80%)
    500000, //500K(80%)
    250000, //250K(80%)
    125000 //125K(80%)
};

static const UINT kDbitTimingPCIE[] = {
    8000000, //8Mbps(80%)
    5000000, //5Mbps(75%)
    5000000, //5Mbps(87.5%)
    4000000, //4Mbps(80%)
    2000000, //2Mbps(80%)
    1000000 //1Mbps(80%)
};

static const unsigned kBaudrate[] = {
    1000000,
    800000,

```

```

500000,
250000,
125000,
100000,
50000,
20000,
10000,
5000
};

void CZlgcanDlg::OnBnClickedButtonInitcan()
{
    if (!device_opened_)
    {
        AddData(_T(""));
        return;
    }

    ZCAN_CHANNEL_INIT_CONFIG config;
    memset(&config, 0, sizeof(config));

    UpdateData(TRUE);

    UINT type = kDeviceType[device_type_index_].device_type;
    const BOOL cloudDevice = type==ZCAN_CLOUD;

```



```

const BOOL netcanfd = IsNetCANFD(type);

const BOOL netcan = IsNetCAN(type);

const BOOL netDevice = (netcan || netcanfd);

const BOOL tcpDevice = IsNetTCP(type);

const BOOL server = net_mode_index_ == 0;

const BOOL usbcanfd = type==ZCAN_USBCANFD_100U ||
    type==ZCAN_USBCANFD_200U ||
type==ZCAN_USBCANFD_MINI;

const BOOL pciecanfd = type==ZCAN_PCIE_CANFD_100U ||
    type == ZCAN_PCIE_CANFD_200U || type ==
ZCAN_PCIE_CANFD_400U_EX;

const BOOL canfdDevice = usbcanfd || pciecanfd;

if (cloudDevice)
{
}

else if (netDevice)
{
    char path[50] = {0};

    char value[100] = {0};

    TCHAR buffer[100] = {0};

    if (tcpDevice)
    {

```

```

sprintf_s(path, "%d/work_mode", channel_index_);
sprintf_s(value, "%d", server?1:0);
ZCAN_SetValue(device_handle_, path, value);
if (server)
{
    sprintf_s(path, "%d/local_port", channel_index_);
    GetDlgItemText(IDC_EDIT_LOCAL_PORT, buffer,
100);
    sprintf_s(value, "%s",
Utility::W2AEx(buffer).c_str());
    ZCAN_SetValue(device_handle_, path, value);
} // server
else
{
    sprintf_s(path, "%d/ip", channel_index_);
    GetDlgItemText(IDC_EDIT_REMOTE_ADDR,
buffer, 100);
    sprintf_s(value, "%s",
Utility::W2AEx(buffer).c_str());
    ZCAN_SetValue(device_handle_, path, value);
    sprintf_s(path, "%d/work_port", channel_index_);
    GetDlgItemText(IDC_EDIT_REMOTE_PORT,

```

```

buffer, 100);

        sprintf_s(value,                "%s",
Utility::W2AEx(buffer).c_str());

        ZCAN_SetValue(device_handle_, path, value);
    }
} // tcp
else
{
    sprintf_s(path, "%d/local_port", channel_index_);
    GetDlgItemText(IDC_EDIT_LOCAL_PORT, buffer, 100);
    sprintf_s(value, "%s", Utility::W2AEx(buffer).c_str());
    ZCAN_SetValue(device_handle_, path, value);
    sprintf_s(path, "%d/ip", channel_index_);
    GetDlgItemText(IDC_EDIT_REMOTE_ADDR,    buffer,
100);
    sprintf_s(value, "%s", Utility::W2AEx(buffer).c_str());
    ZCAN_SetValue(device_handle_, path, value);
    sprintf_s(path, "%d/work_port", channel_index_);
    GetDlgItemText(IDC_EDIT_REMOTE_PORT,    buffer,
100);
    sprintf_s(value, "%s", Utility::W2AEx(buffer).c_str());
    ZCAN_SetValue(device_handle_, path, value);

```

```

        }
    }
else
{
    if (custom_baud_enable_)
    {
        if (!SetCustomBaudrate())
        {
            AddData(_T("    "));
            return;
        }
    }
else
{
    if (!canfdDevice && !SetBaudrate())
    {
        AddData(_T("    "));
        return;
    }
}

if (usbcanfd) {

```

```

char path[50] = {0};

char value[100] = {0};

sprintf_s(path, "%d/canfd_standard", channel_index_);

sprintf_s(value, "%d", 0);

ZCAN_SetValue(device_handle_, path, value);

}

if (usbcanfd)

{

if (custom_baud_enable_)

{

if (!SetCustomBaudrate())

{

AddData(_T("      "));

return;

}

}

else

{

if (!SetCanfdBaudrate())

{

AddData(_T("      "));

```

```

        return;
    }
}

config.can_type = TYPE_CANFD;

config.canfd.mode = work_mode_index_;

config.canfd.filter = filter_mode_;

config.canfd.acc_code = _tctoul(acc_code_, 0, 16);

config.canfd.acc_mask = _tctoul(acc_mask_, 0, 16);
}

else if (pciecanfd)
{
    char path[50] = { 0 };

    char value[100] = { 0 };

    if (!SetCanfdBaudrate())
    {
        AddData(_T("      "));

        return;
    }

    if (type == ZCAN_PCIE_CANFD_400U_EX )
    {
        sprintf_s(path, "0/set_device_recv_merge");

        sprintf_s(value, "0");
    }
}

```

```

        ZCAN_SetValue(device_handle_, path, value);
    }

    config.can_type = TYPE_CANFD;

    config.canfd.mode = work_mode_index_;

    config.canfd.filter = filter_mode_;

    config.canfd.acc_code = _tcstoul(acc_code_, 0, 16);

    config.canfd.acc_mask = _tcstoul(acc_mask_, 0, 16);
}

else
{

    config.can_type = TYPE_CAN;

    config.can.mode = work_mode_index_;

    config.can.filter = filter_mode_;

    config.can.acc_code = _tcstoul(acc_code_, 0, 16);

    config.can.acc_mask = _tcstoul(acc_mask_, 0, 16);
}
}

channel_handle_ = ZCAN_InitCAN(device_handle_, channel_index_,
&config);

if (INVALID_CHANNEL_HANDLE == channel_handle_)
{

    AddData(_T("    "));
}

```

```

        return;
    }

    if (usbcanfd)
    {
        if (resistance_enable_ && !SetResistanceEnable())
        {
            AddData(_T(""));
            return;
        }
    }

    ctrl_int_can_.EnableWindow(FALSE);

    AddData(_T(""));
}

```

```

void CZlgcanDlg::OnBnClickedButtonStartcan()
{
    if (ZCAN_StartCAN(channel_handle_) != STATUS_OK)
    {
        AddData(_T(""));
        return;
    }

    ctrl_start_can_.EnableWindow(FALSE);
}

```



```

start_ = TRUE;

_beginthreadex(NULL, 0, OnDataRecv, this, 0, NULL);

AddData(_T("    "));

if(start_)
{
    UpdateData(TRUE);

    ZCANSendData(_T("601"), _T("3F 40 60 00 0F 00 00 00"));

    ZCANSendData(_T("601"), _T("2F 60 60 00 03 00 00 00"));

    ZCANSendData(_T("602"), _T("3F 40 60 00 0F 00 00 00"));

    ZCANSendData(_T("602"), _T("2F 60 60 00 03 00 00 00"));

    // ZCANSendData(_T("603"), _T("3F 40 60 00 0F 00 00 00"));

    //ZCANSendData(_T("603"), _T("2F 60 60 00 03 00 00 00"));

}

}

void CZlgcanDlg::OnBnClickedButtonReset()

{

    if (ZCAN_ResetCAN(channel_handle_) != STATUS_OK)

    {

        AddData(_T("    "));

        return;

    }

}

```

```

    }

    ctrl_start_can_.EnableWindow(TRUE);

    start_ = FALSE;

    AddData(_T("    "));

}

void CZlgcanDlg::OnBnClickedButtonClose()
{
    // TODO: Add your control notification handler code here

    ZCAN_ResetCAN(channel_handle_);

    ZCAN_CloseDevice(device_handle_);

    start_ = FALSE;

    EnableCtrl(FALSE);

    ctrl_start_can_.EnableWindow(TRUE);

    ctrl_int_can_.EnableWindow(TRUE);

    device_opened_ = FALSE;

    AddData(_T("    "));

}

void CZlgcanDlg::ZCANSendData(CString _id, CString SData)
{
    ZCAN_Transmit_Data can_data;

```

```

    GetZCAN_Data(_id, SData, can_data,TRUE);

    int  nSendCount = 1;

    ZCAN_Transmit_Data*      pData      =      new
ZCAN_Transmit_Data[nSendCount];

    for (int i = 0; i < nSendCount; ++i)
    {
        memcpy_s(&pData[i],      sizeof(ZCAN_Transmit_Data),
&can_data, sizeof(can_data));
    }

    AddData(pData, 1);

    UINT result;

    result = ZCAN_Transmit(channel_handle_, pData, nSendCount);

    delete[] pData;
}

void CZlgcanDlg::OnBnClickedButtonSend()
{
    CWaitCursor waitCursor;

    UpdateData(TRUE);

    char s[15] = { '\0' };

    char strdata[15] = { '\0' };

    DecToHex(273000, s);

```

```

ConvertH_To_L(s, strdata);

CString strSenddata(strdata);

// ZCANSendData(_T("201"), strSenddata);

//ZCANSendData(_T("201"), _T("5C 95 FF FF")); //-27300

ZCANSendData(_T("201"), _T("98 D5 FB FF")); //-273000

ZCANSendData(_T("202"), _T("68 2A 04 00"));

//ZCANSendData(_T("202"), _T("00 00 00 00"));

//ZCANSendData(_T("201"), _T("00 00 00 00"));

//ZCANSendData(_T("603"), _T("23 FF 60 00 00 00 00 00"));

return;

if (datas_.IsEmpty())

{

    AddData(_T("    "));

    return;

}

int  nSendCount = send_count_once_;

UINT result;

if (0 == protocol_index_)//can

{

    ZCAN_Transmit_Data can_data;

    GetViewCANFrame(can_data, TRUE);

```

```

        if (nSendCount > 0)
        {
            ZCAN_Transmit_Data* pData = new
ZCAN_Transmit_Data[nSendCount];

            for (int i=0; i<nSendCount; ++i)
            {
                memcpy_s(&pData[i], sizeof(ZCAN_Transmit_Data),
&can_data, sizeof(can_data));
            }

            AddData(pData, 1);

            result = ZCAN_Transmit(channel_handle_, pData,
nSendCount);

            delete [] pData;
        }
    }
else //canfd
{
    ZCAN_TransmitFD_Data canfd_data;

    GetViewCANFrame(canfd_data, TRUE);

    if (nSendCount > 0)
    {
        ZCAN_TransmitFD_Data* pData = new

```

```

ZCAN_TransmitFD_Data[nSendCount];

    for (int i=0; i<nSendCount; ++i)
    {
        memcpy_s(&pData[i],
sizeof(ZCAN_TransmitFD_Data), &canfd_data, sizeof(canfd_data));

        }

        AddData(pData, 1);

        result = ZCAN_TransmitFD(channel_handle_, pData,
nSendCount);

        delete [] pData;

    }
}

CString csText;

csText.Format(_T("          "), nSendCount, result);

if (result != nSendCount)
{
    AddData(_T("          ") + csText);
}

else
{
    AddData(_T("          ") + csText);
}

```

```
}
```

```
void CZlgcanDlg::InitCombobox( int ctrl_id, int start, int end, int  
current )
```

```
{
```

```
    CComboBox* ctrl =
```

```
static_cast<CComboBox*>(GetDlgItem(ctrl_id));
```

```
    ASSERT(ctrl != NULL);
```

```
    ctrl->ResetContent();
```

```
    CString temp;
```

```
    for (int i = start; i < end; ++i)
```

```
    {
```

```
        temp.Format(_T("%d"), i);
```

```
        ctrl->AddString(temp);
```

```
    }
```

```
    ctrl->SetCurSel(current);
```

```
}
```

```
void CZlgcanDlg::OnCbnSelchangeComboDevice()
```

```
{
```

```
    UpdateData(TRUE);
```

```
    InitCombobox(IDC_COMBO_CHANNEL_INDEX, 0,
```

```

kDeviceType[device_type_index_].channel_count, 0);

    UINT type = kDeviceType[device_type_index_].device_type;

    const BOOL cloudDevice = type==ZCAN_CLOUD;

    const BOOL netcanfd = IsNetCANFD(type);

    const BOOL netcan = IsNetCAN(type);

    const BOOL netDevice = (netcan || netcanfd);

    const BOOL tcpDevice = IsNetTCP(type);

    const BOOL usbcanfd = type==ZCAN_USBCANFD_100U ||
        type==ZCAN_USBCANFD_200U ||
type==ZCAN_USBCANFD_MINI;

    const BOOL pciecanfd = type==ZCAN_PCIE_CANFD_100U ||
        type==ZCAN_PCIE_CANFD_200U ||
type==ZCAN_PCIE_CANFD_400U||
type==ZCAN_PCIE_CANFD_400U_EX;

    const BOOL canfdDevice = usbcanfd || pciecanfd || netcanfd;

    const BOOL accFilter = pciecanfd || type==ZCAN_USBCAN1 ||
type==ZCAN_USBCAN2;

    support_delay_send_ = usbcanfd || pciecanfd || netcanfd;

    support_delay_send_mode_ = usbcanfd || pciecanfd;

    support_get_send_mode_ = usbcanfd || pciecanfd;

    SetCtrlStateDelaySend(support_delay_send_,
support_delay_send_mode_, support_get_send_mode_);

```



```

ctrl_delay_send_mode_.SetCheck(BST_UNCHECKED);

const BOOL support_autosend_canfd = canfdDevice;    // CANFD

const BOOL support_autosend_can = canfdDevice ;    // CANFD

const  BOOL  support_autosend_index  = (support_autosend_can
&& !pciecanfd);    // PCIECANFD , PCIECANFD

const BOOL support_stop_single_autosend = usbcanfd;

const BOOL support_get_autosend_list = netcanfd;

SetAutoSendCtrlState(support_autosend_can,
support_autosend_canfd,                support_autosend_index,
support_stop_single_autosend, support_get_autosend_list);

if (usbcanfd)
{
    m_comboAbit.ResetContent();

    m_comboAbit.AddString(TEXT("1Mbps"));
    m_comboAbit.AddString(TEXT("800kbps"));
    m_comboAbit.AddString(TEXT("500kbps"));
    m_comboAbit.AddString(TEXT("250kbps"));
    m_comboAbit.AddString(TEXT("125kbps"));
    m_comboAbit.AddString(TEXT("100kbps"));
    m_comboAbit.AddString(TEXT("50kbps"));

    m_comboAbit.SetCurSel(0);

    m_comboDbit.ResetContent();
}

```

```

        m_comboDbit.AddString(TEXT("5Mbps"));
        m_comboDbit.AddString(TEXT("4Mbps"));
        m_comboDbit.AddString(TEXT("2Mbps"));
        m_comboDbit.AddString(TEXT("1Mbps"));
        m_comboDbit.SetCurSel(0);
    }
else if (pciecanfd)
{
    m_comboAbit.ResetContent();
    m_comboAbit.AddString(TEXT("1Mbps(80%)"));
    m_comboAbit.AddString(TEXT("800kbps(80%)"));
    m_comboAbit.AddString(TEXT("500kbps(80%)"));
    m_comboAbit.AddString(TEXT("250kbps(80%)"));
    m_comboAbit.AddString(TEXT("125kbps(80%)"));
    m_comboAbit.AddString(TEXT("100kbps"));
    m_comboAbit.AddString(TEXT("50kbps"));
    m_comboAbit.SetCurSel(0);
    m_comboDbit.ResetContent();
    m_comboDbit.AddString(TEXT("5Mbps(80%)"));
    m_comboDbit.AddString(TEXT("4Mbps(80%)"));
    m_comboDbit.AddString(TEXT("2Mbps(80%)"));
    m_comboDbit.AddString(TEXT("1Mbps(80%)"));

```

```
        m_comboDbit.SetCurSel(2);  
    }
```

```
GetDlgItem(IDC_COMBO_DEVICE_INDEX2)->EnableWindow(!cloud  
Device && !netDevice);
```

```
GetDlgItem(IDC_CHECK_RESISTANCE)->EnableWindow(usbcanfd);
```

```
    GetDlgItem(IDC_COMBO_BAUD)->EnableWindow(!canfdDevice  
&& !netDevice && !cloudDevice);
```

```
    GetDlgItem(IDC_COMBO_ABIT)->EnableWindow(canfdDevice  
&& !netDevice && !cloudDevice);
```

```
    GetDlgItem(IDC_COMBO_ABIT2)->EnableWindow(canfdDevice  
&& !netDevice && !cloudDevice);
```

```
GetDlgItem(IDC_CHECK_CUSTOM_BAUDRATE)->EnableWindow(!  
cloudDevice && !netDevice);
```

```
GetDlgItem(IDC_EDIT_CUSTOM_BAUDRATE)->EnableWindow(!clo  
udDevice && !netDevice);
```

```
GetDlgItem(IDC_COMBO_FILTER_MODE)->EnableWindow(accFilter  
&& !cloudDevice && !netDevice);
```

```
GetDlgItem(IDC_EDIT_ACC_CODE)->EnableWindow(accFilter
&& !cloudDevice && !netDevice);

GetDlgItem(IDC_EDIT_ACC_MASK)->EnableWindow(accFilter
&& !cloudDevice && !netDevice);

GetDlgItem(IDC_STATIC_WORK_MODE)->EnableWindow(!cloudDev
ice && !netDevice);

GetDlgItem(IDC_STATIC_BAUD)->EnableWindow(!canfdDevice
&& !netDevice && !cloudDevice);

GetDlgItem(IDC_STATIC_A_BAUDRATE)->EnableWindow(canfdDevi
ce && !netDevice && !cloudDevice);

GetDlgItem(IDC_STATIC_D_BAUDRATE)->EnableWindow(canfdDevi
ce && !netDevice && !cloudDevice);

GetDlgItem(IDC_STATIC_CUSTOM_BAUDRATE)->EnableWindow(!c
loudDevice && !netDevice);

GetDlgItem(IDC_STATIC_FILTER_MODE)->EnableWindow(accFilter
&& !cloudDevice && !netDevice);

GetDlgItem(IDC_STATIC_ACC_CODE)->EnableWindow(accFilter
```

```
&& !cloudDevice && !netDevice);
```

```
GetDlgItem(IDC_STATIC_ACC_MASK)->EnableWindow(accFilter  
&& !cloudDevice && !netDevice);
```

```
GetDlgItem(IDC_STATIC_NET_MODE)->EnableWindow(tcpDevice);
```

```
GetDlgItem(IDC_COMBO_NET_MODE)->EnableWindow(tcpDevice);  
    OnCbnSelchangeComboNetMode();  
}
```

```
void CZlgcanDlg::OnCbnSelchangeComboNetMode()
```

```
{  
    UpdateData(TRUE);  
    UINT type = kDeviceType[device_type_index_].device_type;  
    BOOL tcpDevice = IsNetTCP(type);  
    BOOL udpDevice = IsNetUDP(type);  
    if (tcpDevice)  
    {  
        const BOOL server = net_mode_index_ == 0;
```

```
GetDlgItem(IDC_STATIC_LOCAL_PORT)->EnableWindow(server);
```

```
GetDlgItem(IDC_STATIC_REMOTE_PORT)->EnableWindow(!server);
```

```
GetDlgItem(IDC_STATIC_REMOTE_ADDR)->EnableWindow(!server)
```

```
;
```

```
GetDlgItem(IDC_EDIT_LOCAL_PORT)->EnableWindow(server);
```

```
GetDlgItem(IDC_EDIT_REMOTE_PORT)->EnableWindow(!server);
```

```
GetDlgItem(IDC_EDIT_REMOTE_ADDR)->EnableWindow(!server);
```

```
    } else if (udpDevice)
```

```
    {
```

```
GetDlgItem(IDC_STATIC_LOCAL_PORT)->EnableWindow(TRUE);
```

```
GetDlgItem(IDC_STATIC_REMOTE_PORT)->EnableWindow(TRUE);
```

```
GetDlgItem(IDC_STATIC_REMOTE_ADDR)->EnableWindow(TRUE);
```

```
GetDlgItem(IDC_EDIT_LOCAL_PORT)->EnableWindow(TRUE);
```

```

GetDlgItem(IDC_EDIT_REMOTE_PORT)->EnableWindow(TRUE);

GetDlgItem(IDC_EDIT_REMOTE_ADDR)->EnableWindow(TRUE);
    } else
    {

GetDlgItem(IDC_STATIC_LOCAL_PORT)->EnableWindow(FALSE);

GetDlgItem(IDC_STATIC_REMOTE_PORT)->EnableWindow(FALSE);

GetDlgItem(IDC_STATIC_REMOTE_ADDR)->EnableWindow(FALSE)
;

GetDlgItem(IDC_EDIT_LOCAL_PORT)->EnableWindow(FALSE);

GetDlgItem(IDC_EDIT_REMOTE_PORT)->EnableWindow(FALSE);

GetDlgItem(IDC_EDIT_REMOTE_ADDR)->EnableWindow(FALSE);
    }
}

void CZlgcanDlg::EnableCtrl( BOOL opened )
{

```

```

ctrl_device_type_.EnableWindow(!opened);
ctrl_device_index_.EnableWindow(!opened);
ctrl_channel_index_.EnableWindow(!opened);
ctrl_open_device_.EnableWindow(!opened);
}

BOOL CZlgcanDlg::SetCustomBaudrate()
{
    char path[50] = {0};
    char value[100] = {0};
    string baudrate = Utility::W2AEx(custom_baudrate_);
    sprintf_s(path, "%d/aud_rate_custom", channel_index_);
    sprintf_s(value, "%s", baudrate.data());
    return 1 == ZCAN_SetValue(device_handle_, path, value);
}

UINT WINAPI CZlgcanDlg::OnDataRecv( LPVOID data )
{
    CZlgcanDlg* this_ = static_cast<CZlgcanDlg*>(data);
    if (this_)
    {
        this_->OnRecv();
    }
}

```



```

    return 0;
}

void CZlgcanDlg::OnRecv()
{
    ZCAN_Receive_Data can_data[100];
    ZCAN_ReceiveFD_Data canfd_data[100];
    UINT len;
    while(start_)
    {
        if (len == ZCAN_GetReceiveNum(channel_handle_,
TYPE_CAN))
        {
            len = ZCAN_Receive(channel_handle_, can_data, 100, 50);
            AddData(can_data, len);
            //CString strr;
            //strr.Format(_T("%d"), len);
            //m_list2.AddString(strr);
        }
        if (len == ZCAN_GetReceiveNum(channel_handle_,
TYPE_CANFD))
        {

```

```

        len = ZCAN_ReceiveFD(channel_handle_, canfd_data, 100,
50);

        AddData(canfd_data, len);

    }

    Sleep(8);

}
}

```

```

void CZlgcanDlg::AddData( const ZCAN_Receive_Data* data, UINT
len )
{
    char item[TMP_BUFFER_LEN];
    for (UINT i = 0; i < len; ++i)
    {
        const ZCAN_Receive_Data& can = data[i];
        const canid_t& id = can.frame.can_id;
        sprintf_s(item, "CAN ID:%08X %s %s :%d :", GET_ID(id),
IS_EFF(id)?"":""
, IS_RTR(id)?"":"" , can.frame.can_dlc);
        for (UINT i = 0; i < can.frame.can_dlc; ++i)
        {
            size_t item_len = strlen(item);

```

```

        sprintf_s(&item[item_len],    TMP_BUFFER_LEN-item_len,
"%02X ", can.frame.data[i]);

        sprintf_s(strRecive+i*2, 100, "%02X", can.frame.data[i]);
    }

    ConvertL_To_H(strRecive, strDestRecive);

    CString sss;

    sss.Format(_T("%d"), strlen(strDestRecive));

    AddData(CString(item));

    //char revicedata[1000];

    //sprintf_s(revicedata, 1000, "%02X ", can.frame.data[i]);

    //m_list2.AddString(CString(strDestRecive));

    //m_list2.AddString(sss);

    break;

}

}

```

```

void CZlgcanDlg::AddData( const ZCAN_ReceiveFD_Data* data, UINT
len )
{
    char item[TMP_BUFFER_LEN];

    for (UINT i = 0; i < len; ++i)

```

```

{
    const ZCAN_ReceiveFD_Data& canfd = data[i];

    const canid_t& id = canfd.frame.can_id;

    sprintf_s(item, " CANFD ID:%08X %s %s :%d :",
GET_ID(id), IS_EFF(id)?" ":" "
        , IS_RTR(id)?" ":" ", canfd.frame.len);

    for (UINT i = 0; i < canfd.frame.len; ++i)
    {
        size_t item_len = strlen(item);

        sprintf_s(&item[item_len], TMP_BUFFER_LEN-item_len,
"%02X ", canfd.frame.data[i]);

    }

    AddData(CString(item));

}
}

```

```

void CZlgcanDlg::AddData( const CString& data )
{
    SYSTEMTIME stTime;

    GetLocalTime(&stTime);

    CString csData;

    csData.Format(_T("[%02d:%02d:%02d.%03d] "), stTime.wHour,

```

```

stTime.wMinute, stTime.wSecond, stTime.wMilliseconds);

    csData += data;

    const int MAX_LEN_IN_ONELINE = 100;

    int nStrLen = csData.GetLength();

    int nPrinted = 0;

    CString csLine;

    int nOnce = 0;

    while (nPrinted < nStrLen)
    {
        nOnce = min(MAX_LEN_IN_ONELINE, nStrLen - nPrinted);

        csLine = csData.Mid(nPrinted, nOnce);

        data_recv_list_.AddString(csLine);

        nPrinted += nOnce;
    }

    data_recv_list_.SetCurSel(data_recv_list_.GetCount() - 1);
}

void CZlgcanDlg::AddData( const ZCAN_Transmit_Data* data, UINT
len )
{
    char item[TMP_BUFFER_LEN];

    for (UINT i = 0; i < len; ++i)
    {

```

```

const ZCAN_Transmit_Data& can = data[i];

const canid_t& id = can.frame.can_id;

sprintf_s(item, "Tx CAN ID:%08X %s %s :%d :",
GET_ID(id), IS_EFF(id)? " " "
, IS_RTR(id)? " " " ", can.frame.can_dlc);

for (UINT i = 0; i < can.frame.can_dlc; ++i)
{
size_t item_len = strlen(item);

sprintf_s(&item[item_len], TMP_BUFFER_LEN-item_len,
"%02X ", can.frame.data[i]);

}

AddData(CString(item));

}

}

void CZlgcanDlg::AddData( const ZCAN_TransmitFD_Data* data,
UINT len )
{
char item[TMP_BUFFER_LEN];

for (UINT i = 0; i < len; ++i)
{
const ZCAN_TransmitFD_Data& can = data[i];

const canid_t& id = can.frame.can_id;

```

```

        sprintf_s(item, "Tx CANFD ID:%08X %s %s :%d :",
GET_ID(id), IS_EFF(id)? " " "
        , IS_RTR(id)? " " " ", can.frame.len);
    for (UINT i = 0; i < can.frame.len; ++i)
    {
        size_t item_len = strlen(item);
        sprintf_s(&item[item_len], TMP_BUFFER_LEN-item_len,
"%02X ", can.frame.data[i]);
    }
    AddData(CString(item));
}
}

void CZlgcanDlg::OnClose()
{
    // TODO: Add your message handler code here and/or call default

    if (ZCLOUD_IsConnected())
    {
        ZCLOUD_DisconnectServer();
    }

    OnBnClickedButtonClose();

    CDialog::OnClose();
}

```

```

void CZlgcanDlg::OnBnClickedButtonClear()
{
    data_recv_list_.ResetContent();
}

BOOL CZlgcanDlg::SetTransmitType()
{
    char path[50] = {0};
    char value[100] = {0};
    sprintf_s(path, "%d/send_type", channel_index_);
    sprintf_s(value, "%d", send_type_index_);
    return 1 == ZCAN_SetValue(device_handle_, path, value);
}

BOOL CZlgcanDlg::SetResistanceEnable()
{
    char path[50] = {0};
    sprintf_s(path, "%d/initenal_resistance", channel_index_);
    char value[10] = {0};
    sprintf_s(value, "%d", resistance_enable_);
    return 1 == ZCAN_SetValue(device_handle_, path, value);
}

BOOL CZlgcanDlg::SetBaudrate()
{

```



```

char path[50] = {0};

sprintf_s(path, "%d/aud_rate", channel_index_);

char value[10] = {0};

sprintf_s(value, "%d", kBaudrate[aud_index_]);

return 1 == ZCAN_SetValue(device_handle_, path, value);
}

BOOL CZlgcanDlg::SetCanfdBaudrate()

{

char path[50] = { 0 };

sprintf_s(path, "%d/canfd_abit_baud_rate", channel_index_);

char value[10] = { 0 };

sprintf_s(value, "%d", kAbitTimingUSB[abaud_index_]);

int ret_a = ZCAN_SetValue(device_handle_, path, value);

sprintf_s(path, "%d/canfd_dbit_baud_rate", channel_index_);

sprintf_s(value, "%d", kDbitTimingUSB[dbit_baud_index_]);

int ret_d = ZCAN_SetValue(device_handle_, path, value);

return 1 == (ret_a&&ret_d);

}

void CZlgcanDlg::OnBnClickedButtonGetTxAvailable()

{

// TODO: Add your control notification handler code here

```

```

char path[50] = {0};

char value[100] = {0};

CString csText;

sprintf_s(path,          "%d/get_device_available_tx_count/1",
channel_index_);

const      char*      pRet      =(const      char*)
ZCAN_GetValue(device_handle_,path);

if (pRet)
{
    UINT nSpace = *(int*)pRet;
    csText.Format(_T("      :%d"), nSpace);
}
else
{
    csText.Format(_T("      "));
}

AddData(csText);
}

void CZlgcanDlg::OnBnClickedButtonClearTxQueue()
{
    // TODO: Add your control notification handler code here

```

```

char path[50] = {0};

char value[100] = {0};

sprintf_s(path, "%d/clear_delay_send_queue", channel_index_);

int nRet = ZCAN_SetValue(device_handle_, path, value);

CString csText;

csText.Format(_T(" [%s]"), nRet > 0 ? _T(" ") : _T(" "));

AddData(csText);

}

void CZlgcanDlg::OnBnClickedCheckDelaySendMode()

{

// TODO: Add your control notification handler code here

char path[50] = {0};

char value[100] = {0};

int nDelaySendQueueMode = (ctrl_delay_send_mode_.GetCheck()

== BST_CHECKED);

sprintf_s(path, "%d/set_send_mode", channel_index_);

sprintf_s(value, "%d", nDelaySendQueueMode);

int nRet = ZCAN_SetValue(device_handle_, path, value);

CString csText, csRet;

csText = (nDelaySendQueueMode ? _T(" ") : _T(" "));

csRet.Format(_T(" [%s]"), nRet > 0 ? _T(" ") : _T(" "));

AddData(csText + csRet);

```

```

}

void CZlgcanDlg::SetCtrlStateDelaySend( BOOL delay_send, BOOL
delay_send_mode, BOOL get_send_mode )
{
    ctrl_delay_send_mode_.EnableWindow(delay_send      &&
delay_send_mode);

    GetDlgItem(IDC_BUTTON_GET_TX_AVAILABLE)->EnableWindow(
delay_send);

    GetDlgItem(IDC_BUTTON_CLEAR_TX_QUEUE)->EnableWindow(de
lay_send);

    GetDlgItem(IDC_CHECK_DELAY_SEND_FLAG)->EnableWindow(de
lay_send);

    GetDlgItem(IDC_BUTTON_GET_SEND_MODE)->EnableWindow(del
ay_send && get_send_mode);
}

BOOL CZlgcanDlg::IsNetCAN( UINT type )
{
    return (type==ZCAN_CANETUDP || type==ZCAN_CANETTCP ||

```

```

type==ZCAN_WIFICAN_TCP || type==ZCAN_WIFICAN_UDP ||
        type==ZCAN_CANDTU_NET ||
type==ZCAN_CANDTU_NET_400);
}

BOOL CZlgcanDlg::IsNetCANFD( UINT type )
{
    return (type==ZCAN_CANFDNET_TCP ||
type==ZCAN_CANFDNET_UDP ||
        type==ZCAN_CANFDNET_400U_TCP ||
type==ZCAN_CANFDNET_400U_UDP ||
        type==ZCAN_CANFDWIFI_TCP ||
type==ZCAN_CANFDWIFI_UDP);
}

BOOL CZlgcanDlg::IsNetTCP( UINT type )
{
    return (type==ZCAN_CANETTCP ||
type==ZCAN_WIFICAN_TCP || type==ZCAN_CANDTU_NET ||
type==ZCAN_CANDTU_NET_400 ||
        type==ZCAN_CANFDNET_TCP ||
type==ZCAN_CANFDNET_400U_TCP
||type==ZCAN_CANFDWIFI_TCP);
}

```

```

BOOL CZlgcanDlg::IsNetUDP( UINT type )
{
    return          (type==ZCAN_CANETUDP          ||
type==ZCAN_WIFICAN_UDP ||
                    type==ZCAN_CANFDNET_UDP      ||
type==ZCAN_CANFDNET_400U_UDP
||type==ZCAN_CANFDWIFI_UDP);
}

void CZlgcanDlg::OnBnClickedButtonGetSendMode()
{
    // TODO: Add your control notification handler code here

    CString csText;

    char path[50] = {0};

    sprintf_s(path, "%d/get_send_mode/1", channel_index_);

    const char* pRet = (const char*)ZCAN_GetValue(device_handle_,
path);

    if (pRet)
    {
        UINT nMode = *(int*)pRet;

        csText.Format(_T("      :[%s]"), nMode ? _T("  ") : _T("  "));
    }

    else

```

```

    {
        csText.Format(_T("  "));
    }
    AddData(csText);
}

void CZlgcanDlg::SetAutoSendCtrlState( BOOL support_can, BOOL
support_canfd, BOOL support_index, BOOL support_single_cancel,
BOOL support_get_autosend_list)
{
    BOOL support_autosend = support_can || support_canfd;

    GetDlgItem(IDC_EDIT_AUTOSEND_INDEX)->EnableWindow(support_index);

    GetDlgItem(IDC_EDIT_AUTOSEND_TIME_PERIOD)->EnableWindow(support_autosend);

    GetDlgItem(IDC_BUTTON_ADD_AUTOSEND)->EnableWindow(support_autosend);

    GetDlgItem(IDC_BUTTON_AUTOSEND_START)->EnableWindow(su

```

```

ppport_autosend);

GetDlgItem(IDC_BUTTON_AUTOSEND_STOP)->EnableWindow(sup
port_autosend);

GetDlgItem(IDC_BUTTON_STOP_SINGLE_AUTOSEND)->EnableWi
ndow(support_single_cancel);

GetDlgItem(IDC_BUTTON_GET_DEV_AUTOSEND)->EnableWindow
(support_get_autosend_list);
}

void CZlgcanDlg::AddAutosendCan(int nEnable)
{
    // TODO: Add your control notification handler code here

    UpdateData(TRUE);

    ZCAN_AUTO_TRANSMIT_OBJ autoObj;

    memset(&autoObj, 0, sizeof(autoObj));

    autoObj.enable = nEnable;

    autoObj.interval = auto_send_period_;

    autoObj.index = auto_send_index_;

    GetViewCANFrame(autoObj.obj, FALSE);

    char path[50] = {0};

    sprintf_s(path, "%d/auto_send", channel_index_);

    int nRet = ZCAN_SetValue(device_handle_, path, (const

```



```

char*)&autoObj);

    CString csText;

    csText.Format(_T("   CAN   :%d   :%d   :%d ms ID:0x%X
[%s]"),    autoObj.index,    autoObj.enable,    autoObj.interval,
autoObj.obj.frame.can_id, (nRet ? _T("   ") : _T("   ")));

    AddData(csText);
}

void CZlgcanDlg::AddAutosendCanfd(int nEnable)
{
    // TODO: Add your control notification handler code here

    UpdateData(TRUE);

    ZCANFD_AUTO_TRANSMIT_OBJ autoObj;

    memset(&autoObj, 0, sizeof(autoObj));

    autoObj.enable = nEnable;

    autoObj.interval = auto_send_period_;

    autoObj.index = auto_send_index_;

    GetViewCANFrame(autoObj.obj, FALSE);

    char path[50] = {0};

    sprintf_s(path, "%d/auto_send_canfd", channel_index_);

    int  nRet  =    ZCAN_SetValue(device_handle_,  path,  (const
char*)&autoObj);

    CString csText;

```

```

        csText.Format(_T("  CANFD   :%d   :%d   :%d ms ID:0x%X
[%s]"),      autoObj.index,      autoObj.enable,      autoObj.interval,
autoObj.obj.frame.can_id, (nRet ? _T("  ") : _T(" ")));

        AddData(csText);
}

```

```

void CZlgcanDlg::OnBnClickedButtonAutosendStart()
{
    // TODO: Add your control notification handler code here

    char path[50] = {0};

    char value[100] = {0};

    sprintf_s(path, "%d/apply_auto_send", channel_index_);

    int nRet = ZCAN_SetValue(device_handle_, path, "0");

    CString csText;

    csText.Format(_T("  [%s]"), (nRet ? _T(" ") : _T(" ")));

    AddData(csText);
}

```

```

void CZlgcanDlg::OnBnClickedButtonAutosendStop()
{
    // TODO: Add your control notification handler code here

    char path[50] = {0};

    char value[100] = {0};

```

```

    sprintf_s(path, "%d/clear_auto_send", channel_index_);

int nRet = ZCAN_SetValue(device_handle_, path, "0");

    CString csText;

    csText.Format(_T(" [%s]"), (nRet ? _T(" ") : _T(" ")));

    AddData(csText);

}

void CZlgcanDlg::OnBnClickedButtonStopSingleAutosend()

{

    // TODO: Add your control notification handler code here

    UpdateData(TRUE);

    if (0 == protocol_index_)

    {

        //CAN

        AddAutosendCan(0);

    }

    else

    {

        // CANFD

        AddAutosendCanfd(0);

    }

}

void CZlgcanDlg::GetViewCANFrame( ZCAN_Transmit_Data&

```

```

can_data, BOOL add_delay_flag )
{
    UpdateData(TRUE);

    UINT id = _tcstoul(id_, 0, 16);

    string data = Utility::W2AEx(datas_);

    BOOL bDelay = frm_delay_flag_;

    UINT nDelayTime = frm_delay_time_;

    memset(&can_data, 0, sizeof(can_data));

    can_data.frame.can_id = MAKE_CAN_ID(id, frame_type_index_, 0,
0);

    can_data.frame.can_dlc    =    Utility::split(can_data.frame.data,
CAN_MAX_DLEN, data, ' ', 16);

    can_data.transmit_type = send_type_index_;

    if (add_delay_flag && bDelay)
    {
        can_data.frame.__pad |= TX_DELAY_SEND_FLAG;

        can_data.frame.__res0 = LOBYTE(nDelayTime);

        can_data.frame.__res1 = HIBYTE(nDelayTime);

    }
}

void CZlgcanDlg::GetViewCANFrame( ZCAN_TransmitFD_Data&
canfd_data, BOOL add_delay_flag )

```

```

{
    UpdateData(TRUE);

    UINT id = _tcstoul(id_, 0, 16);

    string data = Utility::W2AEx(datas_);

    BOOL bDelay = frm_delay_flag_;

    UINT nDelayTime = frm_delay_time_;

    memset(&canfd_data, 0, sizeof(canfd_data));

    canfd_data.frame.can_id = MAKE_CAN_ID(id, frame_type_index_,
0, 0);

    canfd_data.frame.len = Utility::split(canfd_data.frame.data,
CANFD_MAX_DLEN, data, '\', 16);

    canfd_data.transmit_type = send_type_index_;

    canfd_data.frame.flags |= canfd_exp_index_ ? CANFD_BRS : 0;

    if (add_delay_flag && bDelay)
    {
        canfd_data.frame.flags |= TX_DELAY_SEND_FLAG;

        canfd_data.frame.__res0 = LOBYTE(nDelayTime);

        canfd_data.frame.__res1 = HIBYTE(nDelayTime);

    }

}

void CZlgcanDlg::GetZCAN_Data(CString _id, CString SData,

```

```

ZCAN_Transmit_Data& can_data, BOOL add_delay_flag)
{
    UINT id = _tcstoul(_id, 0, 16);

    string data = Utility::W2AEx(SData);

    BOOL bDelay = frm_delay_flag_;

    UINT nDelayTime = frm_delay_time_;

    memset(&can_data, 0, sizeof(can_data));

    can_data.frame.can_id = MAKE_CAN_ID(id, frame_type_index_, 0,
0);

    can_data.frame.can_dlc    =    Utility::split(can_data.frame.data,
CAN_MAX_DLEN, data, ', ', 16);

    can_data.transmit_type = send_type_index_;

    if (add_delay_flag && bDelay)
    {
        can_data.frame.__pad |= TX_DELAY_SEND_FLAG;

        can_data.frame.__res0 = LOBYTE(nDelayTime);

        can_data.frame.__res1 = HIBYTE(nDelayTime);

    }
}

void CZlgcanDlg::OnBnClickedButtonAddAutosend()
{
    // TODO: Add your control notification handler code here

```

```

UpdateData(TRUE);

if (0 == protocol_index_)
{
    //CAN

    AddAutosendCan(1);
}
else
{
    // CANFD

    AddAutosendCanfd(1);
}
}

void CZlgcanDlg::OnBnClickedButtonGetDevAutosend()
{
    // TODO: Add your control notification handler code here

    CString csText;

    UINT nCount = 0;

    char path[50] = {0};

    //CAN

    sprintf_s(path, "%d/get_auto_send_can_count/1", channel_index_);

    const char* pRet = (const
char*)ZCAN_GetValue(device_handle_,path);

```

```

if (pRet)
{
    nCount = *(int*)pRet;

    csText.Format(_T("CAN   :%d"), nCount);

    AddData(csText);

    if (nCount > 0)
    {
        sprintf_s(path,          "%d/get_auto_send_can_data/1",
channel_index_);

        pRet          =          (const          char*)
ZCAN_GetValue(device_handle_,path);

        if (pRet)
        {
            const ZCAN_AUTO_TRANSMIT_OBJ* pData =
(ZCAN_AUTO_TRANSMIT_OBJ*)pRet;

            for (UINT i = 0; i < nCount; i++)
            {
                csText.Format(_T("CAN   :%d   :%d ms
ID:0x%08x"),          pData[i].index,          pData[i].interval,
pData[i].obj.frame.can_id);

                AddData(csText);
            }
        }
    }
}

```



```

        }
    else
    {
        csText.Format(_T("    "));
        AddData(csText);
        return;
    }
}
}
else
{
    csText.Format(_T("    "));
    AddData(csText);
    return;
}

//CANFD
sprintf_s(path,                "%d/get_auto_send_canfd_count/1",
channel_index_);

pRet = (const char*)ZCAN_GetValue(device_handle_,path);
if (pRet)
{
    nCount = *(int*)pRet;

```

```

csText.Format(_T("CANFD :%d"), nCount);

AddData(csText);

if (nCount > 0)
{
    sprintf_s(path,          "%d/get_auto_send_canfd_data/1",
channel_index_);

    pRet = (const char*)ZCAN_GetValue(device_handle_, path);

    if (pRet)
    {
        const ZCANFD_AUTO_TRANSMIT_OBJ* pData =
(ZCANFD_AUTO_TRANSMIT_OBJ*)pRet;

        for (UINT i = 0; i < nCount; i++)
        {
            csText.Format(_T("CANFD      :%d      :%d ms
ID:0x%08x"),          pData[i].index,          pData[i].interval,
pData[i].obj.frame.can_id);

            AddData(csText);

        }
    }
else
{
    csText.Format(_T("      "));

```

```

        AddData(csText);

        return;
    }
}
else
{
    csText.Format(_T("  "));
    AddData(csText);
    return;
}
}

void CZlgcanDlg::OnCbnSelchangeComboDeviceIndex()
{
    // TODO:  }

```

```

long int CZlgcanDlg::Daonagongshi(long int v_n_1, double power)
{
    //UpdateData(TRUE);

    double k0 = 1.0f / m_B;

    double T0 = (m_M*1.0f) / (m_B*1.0f);

    double Ts = 0.008;

```

```

        int v_n = k0 * (Ts / (T0 + Ts)) * power*3184*2730*80 + (T0 / (T0 +
Ts)) * v_n_1;

        return v_n;
}

```

```

int num = 0;

int mm = 0;

double dmax = 1.0, dmin = 2.0;

void CZlgcanDlg::OnTimer(UINT_PTR nIDEvent)
{
// TODO:

    if (nIDEvent==1)
    {

GetPowerData(&m_data[0],&m_data[1],&m_data[2],&m_data[3]);

        CString str;

        str.Format(_T("%lf"), m_data[2]);

        //m_list2.AddString(str);

        /*if (m_iRS232data != m_iRS232Predata)
        {

```

```

        str.Format(_T("n=%d,%d"), num, m_iRS232data);

        m_list2.AddString(str);

        m_iRS232Predata = m_iRS232data;

    }*/

    if (m_count < 300)

    {

        m_count++;

        return;

    }

    //if (m_data[2] < 1.2)

    //    return;

    //mm++;

    double spower = m_data[0] + m_data[2];

    double power = KalmanFilter1(&kfpVar, spower);

    if (power > dmax)

        dmax = power;

    if (power < dmin)

        dmin = power;

    str.Format(_T("%lf"), power);

    m_list2.AddString(str);

    power = power - 1.12;

    if (fabs(power) < 0.065)

```

```

        power = 0;
if (power < 0)
    power = 0;
//power = -power;
char s1[15] = { '\0' }, s2[15] = { '\0' };;
char strdata1[15] = { '\0' }, strdata2[15] = { '\0' },fstrdata[15] =
{ '\0' };
long int v_n_1;
char bDest[4];
str.Format(_T("%s"), strDestRecive);
//m_list2.AddString(CString(strDestRecive));
if(strDestRecive[0] == 'F')
{
    FConvertH_To_LNoSpace(strDestRecive, fstrdata);
    StrToHex(bDest, fstrdata, 8);
    memcpy(&v_n_1, bDest, 4);
    v_n_1 = fabs(v_n_1);
    // str.Format(_T("%ld"), v_n_1);
    //m_list2.AddString(str);
}
else
{

```

```

        v_n_1 = strtol(strDestRecive, NULL, 16);
    }

    long int v_n = Daonagongshi(v_n_1, power);
    //long int v_n = Daonagongshi(v_n_1, 0.2*80);

    DecToHex(-v_n, s1);

    ConvertH_To_L(s1, strdata1);

    DecToHex(v_n, s2);

    ConvertH_To_L(s2, strdata2);

    str.Format(_T("%ld  %ld"), v_n_1,v_n);

    std::ofstream file("example.txt", std::ios::app);

    if (!file.is_open()) {

        std::cerr << "      " << std::endl;

    }

    file << power << '\t' << v_n_1 << '\t' << v_n << std::endl;

    file.close();

    //m_list2.AddString(CString(strdata1));

    CString strSenddata1(strdata1), strSenddata2(strdata2);

    ZCANSendData(_T("201"), strSenddata1);

    ZCANSendData(_T("202"), strSenddata2);

}

CDialog::OnTimer(nIDEvent);

}

```

```

char buffer[100];

DWORD bytesRead, bytesWritten;

unsigned __stdcall ReadRS232File(LPVOID pObject)
{
    while (TRUE)
    {
        if (ReadFile(m_RS232handle, buffer, sizeof(buffer) - 1,
&bytesRead, NULL))
        {
            if (bytesRead > 0)
            {
                buffer[bytesRead] = '\0';

                //int data = buffer[0];

                m_iRS232data = buffer[0];

                num++;

                //printf("Received: %d\n", data);

                //return data;
            }
        }
    }

    free(pObject);

    _endthreadex(0);
}

```



```

return 0;

}

void CZlgcanDlg::StartRS232Thread()

{

    // TODO:

    HANDLE          hThread;

unsigned          threadID;

    HANDLE          hMutex;

hMutex = CreateMutex(NULL, FALSE, _T("RS"));

if (ERROR_ALREADY_EXISTS == GetLastError()) {

    return ;

}

CloseHandle(hMutex);

hThread = (HANDLE)_beginthreadex(NULL, 0, &ReadRS232File,

NULL, 0, &threadID);

    CloseHandle(hThread);

}

int CZlgcanDlg::InitRS232()

{

    // TODO: .

    //

    m_RS232handle = CreateFile(_T("COM4"), GENERIC_READ |

```

```

GENERIC_WRITE, 0, NULL,

        OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);

if (m_RS232handle == INVALID_HANDLE_VALUE)

{

    printf("Failed to open serial port\n");

    return 1;

}

DCB dcbSerialParams = { 0 };

dcbSerialParams.DCBlength = sizeof(dcbSerialParams);

if (!GetCommState(m_RS232handle, &dcbSerialParams))

{

    printf("Failed to get serial port parameters\n");

    CloseHandle(m_RS232handle);

    return 1;

}

dcbSerialParams.BaudRate = CBR_9600;

dcbSerialParams.ByteSize = 8;

dcbSerialParams.StopBits = ONESTOPBIT;

dcbSerialParams.Parity = NOPARITY;

if (!SetCommState(m_RS232handle, &dcbSerialParams))

{

    printf("Failed to set serial port parameters\n");

```

```

        CloseHandle(m_RS232handle);

        return 1;
    }

    //printf("%d\n", m_Handle);

    return 0;
}

void CZlgcanDlg::StrToHex(char* pbDest, char* pbSrc, int nLen)
{
    char h1, h2;

    char s1, s2;

    int i;

    for (i = 0; i < nLen / 2; i++)
    {
        h1 = pbSrc[2 * i];

        h2 = pbSrc[2 * i + 1];

        s1 = toupper(h1) - 0x30; //toupper
        if (s1 > 9)
            s1 -= 7;

        s2 = toupper(h2) - 0x30;

        if (s2 > 9)
            s2 -= 7;

        pbDest[i] = s1 * 16 + s2;
    }
}

```

```

    }
}

void CZlgcanDlg::DecToHex(long int x, char* p)
{
    memset(p, 0, sizeof(p));

    int i = 0;

    if (x >= 41900000)
        x = 41900000;

    //return;

    /* while (x)
    { //一、
        if (x < 0)
        {
            p[i++] = '-';
            x = -x;
        }

        p[i] = x % 16;

        if (p[i] > 9)
        {
            p[i] += 55;
            i++;

```

```

    }
    else
    {
        p[i] += 48;
        i++;
    }
    x /= 16;
}
p[i] = '\0';
int left = (p[0] == '-') ? 1 : 0, right = i - 1;
while (left < right)
{
    char temp = p[left];
    p[left] = p[right];
    p[right] = temp;
    left++;
    right--;
}
sprintf(p, "%X", x);
}
void CZlgcanDlg::FConvertH_To_LNoSpace(char* source, char* dest)
{

```

```

memset(dest, 0, sizeof(dest));

int nn = 0, kk = 0;

if (source[0] == 'F')
{
    nn = 8;

    kk = 0;

    while (nn >= 2 && kk < 10)
    {
        dest[kk] = source[nn - 2];

        dest[kk + 1] = source[nn - 1];

        kk = kk + 2;

        nn = nn - 2;

    }

    dest[kk] = '\0';

    return;
}

int CZlgcanDlg::ConvertH_To_L(char* source, char* dest)
{

    memset(dest, 0, sizeof(dest));

    int n = 0, k = 0;

```

```

if (source[0]=='F')
{
    n = 8;
    k = 0;
    while (n >= 2 && k < 10)
    {
        dest[k] = source[n - 2];
        dest[k + 1] = source[n - 1];
        dest[k + 2] = ' ';
        k = k + 3;
        n = n - 2;
    }
    dest[k] = '\0';
    return 0;
}

n = 0;
k = 0;
while (source[n] != '\0')
{
    n++;
}
if (n == 0)

```

```

{
    while (k < 10)
    {
        dest[k] = '0';
        dest[k + 1] = '0';
        dest[k + 2] = ' ';
        k = k + 3;
    }
    dest[k] = '\0';
}
else if (n % 2 == 0)//n
{
    while (n >= 2 && k < 10)
    {
        dest[k] = source[n - 2];
        dest[k + 1] = source[n - 1];
        dest[k + 2] = ' ';
        k = k + 3;
        n = n - 2;
    }
    while (k < 10)
    {

```



```

        dest[k] = '0';
        dest[k + 1] = '0';
        dest[k + 2] = ' ';
        k = k + 3;
    }
    dest[k] = '\0';
}
else//
{
    while (n >= 3 && k < 10)
    {
        dest[k] = source[n - 2];
        dest[k + 1] = source[n - 1];
        dest[k + 2] = ' ';
        k = k + 3;
        n = n - 2;
    }
    if (k < 10)
    {
        dest[k] = '0';
        dest[k + 1] = source[n - 1];
        dest[k + 2] = ' ';
    }
}

```

```

        k = k + 3;
    }
    while (k < 10)
    {
        dest[k] = '0';

        dest[k + 1] = '0';

        dest[k + 2] = ' ';

        k = k + 3;
    }

    dest[k] = '\0';
}

return 0;
}

```

```

void CZlgcanDlg::ConvertL_To_H(char* source, char* dest)
{
    memset(dest, 0, sizeof(dest));

    int n = 0, k = 0;

    n = strlen(source);

    if (n != 8)

        return;

    while (n >= 0)

```

```
{  
    dest[k] = source[n - 2];  
    dest[k + 1] = source[n - 1];  
    n = n - 2;  
    k = k + 2;  
}  
dest[8] = '\0';  
}  
void CZlgcanDlg::OnBnClickedBntSet()  
{  
    UpdateData(TRUE);  
}
```