Минобрнауки России

Федеральное государственное автономное образовательное учреждение высшего образования   
«Национальный исследовательский университет   
«Московский институт электронной техники»

Институт cистемной и программной инженерии и информационных технологий

(СПИНТех)

Артамонова Анастасия Юрьевна

Магистерская диссертация   
по направлению 09.04.04 «Программная инженерия»

Исследование и разработка методики и алгоритма составления тренировок для реабилитации после травм на основе математической модели движения человека

Студент Артамонова А.Ю.

Руководитель, Доцент, к.т.н. Федоров А.Р.

Москва 2024

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc184924873)

[ГЛАВА 1. АНАЛИТИЧЕСКИЙ ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ И СРЕДСТВ МОДЕЛИРОВАНИЯ ДВИЖЕНИЙ ЧЕЛОВЕКА 5](#_Toc184924874)

[1.1. Анализ методов моделирования движений человека 5](#_Toc184924875)

[1.2. Анализ средств моделирования движений человека 19](#_Toc184924876)

[1.3. Обоснование метода решения 23](#_Toc184924877)

[Выводы по главе 1 25](#_Toc184924878)

[ГЛАВА 2. ФОРМАЛИЗОВАННОЕ ПРЕДСТАВЛЕНИЕ ПРОЦЕССА МОДЕЛИРОВАНИЯ ДВИЖЕНИЙ ЧЕЛОВЕКА 26](#_Toc184924879)

[2.1. Формализованное представление задачи математического моделирования движений 26](#_Toc184924880)

[2.2. Разработка методики моделирования движений человека 29](#_Toc184924881)

[2.3. Разработка алгоритма моделирования движений человека 29](#_Toc184924882)

[Выводы по главе 2 31](#_Toc184924883)

[ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТАЛЬНОЕ ПОДТВЕРЖДЕНИЕ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ 32](#_Toc184924884)

[3.1 Разработка алгоритма моделирования движений человека 32](#_Toc184924885)

[3.2 Верификация и сравнение модели с помощью метрики качества (выбор метрики качества) 32](#_Toc184924886)

[3.3 Обоснование достоверности полученных результатов 32](#_Toc184924887)

ВВЕДЕНИЕ

В современном обществе вопросы реабилитации после травм и восстановления двигательной активности становятся все более актуальными и значимыми. Увеличение числа случаев травм, особенно в области спорта и повседневной жизни, подчеркивает неотложность не только качественного и эффективного лечения, но и индивидуализированных программ реабилитации.

**Актуальность исследования.** Одним из ключевых моментов в данном контексте является создание методики и алгоритма, основанных на математической модели движения человека, для составления персонализированных тренировочных программ. В настоящее время существует явная потребность в разработке систем, способных учесть индивидуальные особенности пациентов и обеспечить оптимальный путь восстановления после полученных травм.

Научное исследование, направленное на создание математической модели движения человека и разработку алгоритма для составления тренировок, обосновывается стремлением к повышению эффективности реабилитации и улучшению результатов лечения. Предполагаемый результат – инновационный подход к реабилитации, учитывающий не только физиологические аспекты, но и индивидуальные особенности пациента.

Новизна данного исследования заключается в том, что оно стремится заполнить существующий пробел в знаниях, предлагая интегрированный и математически обоснованный метод составления тренировок для реабилитации. Проработанность направления подтверждается предшествующими исследованиями, которые рассматривают биомеханику, кинематику и динамику движений человека, но до сих пор остаются недостаточно ориентированными на создание персонализированных тренировок.

Результаты данного исследования обладают высокой значимостью для области медицины и спорта. Разработка инновационного подхода к реабилитации после травм не только улучшит качество жизни пациентов, но также окажет влияние на развитие методологии тренировок в области физической подготовки и спортивной медицины.

**Проблемная ситуация.** Существующие методики реабилитации часто не учитывают индивидуальные физиологические особенности пациентов, оставляя важные нюансы восстановления без должного внимания. Математически обоснованные методики могут предоставить инструменты для создания персонализированных программ, адаптированных к конкретным потребностям пациентов.

**Объектом исследования** является процесс реабилитации после травм с использованием математических уравнений для моделирования движений человека и разработке алгоритма для составления индивидуализированных тренировок.

**Предмет исследования** является математическое моделирование движений человека в контексте реабилитации после травм, с особым вниманием к разработке алгоритма для индивидуализированных тренировок. В рамках этого исследования рассматриваются методы оптимизации процесса восстановления, учет индивидуальных особенностей пациентов, а также разработка критериев эффективности программ реабилитации.

**Цель и задачи исследования.** Разработка и реализация математической модели движений человека в контексте реабилитации после травм для создания эффективного алгоритма индивидуализированных тренировок.В соответствии с целью и предметом исследования в диссертации необходиморешить следующие задачи:

* аналитический обзор существующих средств и методов моделирования движений человека;
* формализация задачи моделирования движений человека;
* разработка методики моделирования движений человека;
* программная реализация разработанной методики;
* оценка достоверности полученных результатов.

**Научная новизна исследования.** Успешно решена новая задача разработки математической модели движений человека с учетом специфики реабилитации, представляя важный вклад в область медицинского моделирования.

**Обоснованность и достоверность результатов.** Реальность использованных исходных данных при расчетах подтверждает их соответствие физиологическим и биомеханическим характеристикам пациентов, обеспечивая тем самым доверенность результатов.

**Практическая значимость результатов,** полученных в диссертационной работе, заключается в их применимости в медицинской сфере деятельности для оптимизации программ восстановления после травм.

ГЛАВА 1. АНАЛИТИЧЕСКИЙ ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ И СРЕДСТВ МОДЕЛИРОВАНИЯ ДВИЖЕНИЙ ЧЕЛОВЕКА

* 1. Анализ методов моделирования движений человека

В современном обществе вопросы реабилитации после травм и восстановления двигательной активности становятся все более актуальными и значимыми. Увеличение числа случаев травм, особенно в области спорта и повседневной жизни, подчеркивает неотложность не только качественного и эффективного лечения, но и индивидуализированных программ реабилитации. Эффективность восстановления во многом зависит от того, насколько точно разработанная программа учитывает биомеханические особенности движения конкретного человека, а также характер и степень повреждений.

Одним из ключевых моментов в данном контексте является создание методики и алгоритма, основанных на математической модели движения человека, для составления персонализированных тренировочных программ. Точные математические модели позволяют не только прогнозировать возможные сценарии восстановления, но и оптимизировать двигательные нагрузки, снижая риск повторных травм. Разработка таких моделей требует учета множества факторов: анатомических особенностей, кинематических параметров, биомеханики суставов и работы мышц, а также характеристик внешней среды.

В настоящее время существует явная потребность в разработке систем, способных учесть индивидуальные особенности пациентов и обеспечить оптимальный путь восстановления после полученных травм. В этом контексте задача инверсной кинематики становится одной из ключевых. Инверсная кинематика, как раздел механики, занимается определением угловых положений суставов и сегментов тела для достижения заданного положения или траектории конечностей. Это особенно важно в задачах реабилитации, где необходимо учитывать сложные многосуставные движения человека, такие как ходьба, бег или манипуляции руками.

Существующие методы решения задач инверсной кинематики предлагают разнообразные подходы, начиная от аналитических методов, основанных на строгих математических расчетах, и заканчивая численными методами и алгоритмами машинного обучения. Каждый из этих подходов имеет свои преимущества и ограничения, которые зависят от специфики применяемой модели и поставленных целей. Рассмотрение существующих методов позволяет выявить наиболее эффективные инструменты и подходы, способствующие разработке персонализированных систем реабилитации.

Существуют несколько численных методов, которые можно использовать для решения задачи инверсной кинематики.

Метод Ньютона-Рафсона – это итерационный численный метод для приближенного нахождения корней уравнения f(x) = 0. Он основан на последовательных линейных приближениях функции и использовании касательных для нахождения лучшего приближения к корню.

Алгоритм метода Ньютона-Рафсона:

1. Начальная аппроксимация x0 выбирается достаточно близко к истинному корню.
2. Следующее приближение xn+1 вычисляется по формуле:

где – значение функции в точке ,– значение производной в точке .

1. Процесс повторяется, пока не достигнется заданная точность или пока изменения между последовательными приближениями не станут достаточно малы.

Для решения задачи инверсной кинематики методом Ньютона-Рафсона необходимо выполнить следующие шаги:

1. Определить начальные значения для θ1 и θ2.

2. Определить функции ошибки:

3. Составить матрицу Якоби, которая представляет собой частные производные функций f1 и f2 по переменным и :

Вычисление производных:

4. Итеративно обновить значения и , используя формулу Ньютона-Рафсона:

5. Продолжить процесс до тех пор, пока ошибка между текущей и целевой позицией не станет меньше заданного значения.

Основное преимущество метода Ньютона-Рафсона — это квадратичная сходимость при достаточно хорошем начальном приближении. Это означает, что при каждом шаге ошибка уменьшается пропорционально квадрату предыдущей ошибки, что делает метод очень быстрым по сравнению с другими численными методами, такими как метод бисекции или секущих.

Метод легко применим к уравнениям, где производную функции можно вычислить аналитически или численно, что позволяет быстро приближаться к решению.

Метод Ньютона-Рафсона используется не только для нахождения корней уравнений, но и для задач оптимизации, где необходимо находить экстремумы функций, решать системы нелинейных уравнений и выполнять другие сложные вычислительные задачи.

Метод можно расширить для решения систем нелинейных уравнений с помощью матрицы Якоби, что делает его мощным инструментом для многомерных задач.

К недостаткам метода Ньютона-Рафсона относится качество и скорость сходимости сильно зависят от выбора начальной точки 𝑥0. Если начальное приближение выбрано неудачно, метод может сходиться медленно, застрять в локальном минимуме или не сойтись вовсе.

Метод не гарантирует сходимость для всех функций. Вблизи точек перегиба, разрывов производной или для функций с несколькими корнями метод может давать ошибочные или расходящиеся решения. Например, если производная 𝑓′(𝑥) = 0, то следующая итерация метода не определена.

Метод требует, чтобы функция была дифференцируемой, и её производная была вычислима во всех точках. Если производная не существует или трудно вычисляется, метод может оказаться неприменимым.

Для каждой итерации необходимо вычислять производную функции, что может быть вычислительно дорого или затруднительно, особенно для сложных функций или систем уравнений.

Метод может сходиться к разным корням в зависимости от начального приближения. Это создает трудности при решении уравнений с несколькими корнями, так как нельзя точно знать, к какому корню метод будет стремиться.

Если производная функции в точке приближения очень мала, это может привести к большим шагам в направлении поиска, что может нарушить сходимость или сделать метод нестабильным.

Метод градиентного спуска – это один из основных численных методов оптимизации, используемый для минимизации (или максимизации) функции, обычно называемой функцией потерь. Он основывается на идее, что значение функции уменьшается быстрее всего, если двигаться в направлении, противоположном градиенту функции в данной точке.

Алгоритм метода градиентного спуска:

1. Выбирается начальная точка x0 (начальная аппроксимация).
2. На каждом шаге вычисляется градиент функции ∇f(xn) в текущей точке xn, который указывает направление наибольшего роста функции. Градиент – это вектор, который указывает направление наибольшего увеличения функции. Для многомерной функции f(x1, x2, …, xn) градиентом будет вектор частных производных:
3. Следующее приближение вычисляется по формуле:

где α – шаг обучения, который определяет, насколько сильно изменится точка на каждом шаге, – градиент функции в точке xn, который показывает направление наибольшего роста функции.

1. Процесс повторяется, пока не достигнется минимальное значение функции или изменение функции станет достаточно малым.

Для решения задачи инверсной кинематики методом градиентного спуска необходимо выполнить следующие шаги:

1. Определить начальные значения для θ1 и θ2.

2. Определить функцию ошибки:

3. Вычислить частные производные (градиенты) функции ошибки по углам и .

4. Итеративно обновить значения и в направлении, противоположном градиенту ошибки с использованием шага обучения :

5. Продолжить процесс до тех пор, пока ошибка не станет меньше заданного значения или не будет превышено максимальное количество итераций.

Главное достоинство метода градиентного спуска состоит в том, что он обеспечивает эффективный поиск минимумов функции, особенно в тех случаях, когда другие подходы требуют значительных вычислительных мощностей. Метод градиентного спуска отлично справляется с большими объемами данных и многомерными функциями, где аналитическое решение задачи может быть либо слишком сложным, либо недоступным.

Тем не менее, у данного метода имеются свои недостатки. Во-первых, он зависит от выбора величины шага обучения. Если шаг слишком мал, оптимизация будет происходить очень медленно, а если он слишком велик, метод может не достичь оптимального решения и даже привести к дивергенции. Кроме того, градиентный спуск может "застревать" в локальных минимумах, особенно в задачах с сильно нелинейными функциями. В таких ситуациях метод может не обнаружить глобальный минимум и остановиться на одном из локальных решений.

Еще одной сложностью является необходимость вычисления градиента, что может быть затруднительно для сложных моделей или больших объемов данных. В заключение, градиентный спуск может демонстрировать медленное сходимость на плоских участках функции потерь, где градиенты практически равны нулю. Это может потребовать применения дополнительных методов, таких как адаптивные алгоритмы изменения шага.

Матрица Якоби — это матрица частных производных векторной функции f(x), которая описывает локальные линейные зависимости между функцией и её переменными. Для функции , векторная функция имеет матрицу Якоби Jf(x) следующего вида:

Задачи, связанные с матрицей Якоби, включают решение систем нелинейных уравнений вида f(x) = 0, где f(x) – векторная функция.

Матрица Якоби описывает локальное поведение нелинейной функции, что позволяет использовать её для приближенного решения систем уравнений. Матрица Якоби является основным инструментом в методах оптимизации, таких как метод Ньютона и другие градиентные методы. Она позволяет проводить анализ устойчивости динамических систем. Матрица Якоби легко обобщается на многомерные функции, что делает её полезной для работы с многими переменными.

Вычисление матрицы Якоби для сложных функций может быть трудоемким и требовать значительных вычислительных ресурсов. Для сильно нелинейных функций матрица может изменяться значительно между итерациями, что усложняет процесс оптимизации. В задачах с шумными данными матрица Якоби может быть нестабильной и не давать адекватных результатов. Для корректного вычисления требуется, чтобы функция была непрерывной и дифференцируемой.

Когда матрица Якоби не является квадратной или не обратима, вместо обычной обратной матрицы используется **псевдообратная матрица Мура-Пенроуза**. Псевдообратная матрица позволяет решить систему уравнений в смысле минимизации ошибок или нахождения наилучшего приближения решения.

Для вычисления псевдообратной матрицы Якоби используют различные методы, но один из самых распространенных – метод сингулярного разложения матрицы Jf(x).

Если матрица Jf(x) представлена в виде Jf(x) = U∑VT, где U и V – ортогональные матрицы, ∑ - диагональная матрица сингулярных чисел, то псевдообратная матрица определяется как:

где - псевдообратная матрица диагольной матрицы ∑, которая вычисляется путем обращения всех ненулевых элементов на диагонали, а все нулевые элементы остаются на месте.

Одним из ключевых достоинств псевдообратной матрицы Якоби является её способность справляться с переопределёнными системами, в которых количество уравнений превышает количество неизвестных, а также с недоопределёнными системами. В таких случаях традиционная обратная матрица не может быть использована, и псевдообратная матрица позволяет находить оптимальные решения в контексте метода наименьших квадратов. Кроме того, ещё одним её преимуществом является возможность стабилизации численных решений, особенно в ситуациях, когда матрица Якоби имеет плохую обусловленность (то есть её определитель близок к нулю).

Тем не менее, у данного метода имеются и свои недостатки. Прежде всего, вычисление псевдообратной матрицы может потребовать значительных ресурсов при работе с большими матрицами, так как включает в себя процесс сингулярного разложения (SVD). Этот этап довольно медленный по сравнению с обычным обращением матриц, особенно в случае крупных систем. Кроме того, псевдообратная матрица может оказаться чувствительной к шумам в данных, что может привести к нестабильным результатам. Это связано с тем, что небольшие сингулярные значения в процессе SVD могут значительно увеличиваться при обращении, что, в свою очередь, приводит к усилению ошибок.

Метод экспоненциального сглаживания – это метод анализа временных рядов, используемая для уменьшения шума и колебаний в данных, что позволяет получить более плавные и предсказуемые результаты. Этот методосновывается на применении экспоненциальных весов к наблюдениям временного ряда. Более свежие данные получают больший вес, а старые данные — меньший. Таким образом, сглаженные значения реагируют на изменения в данных, но менее подвержены случайным колебаниям и шуму.

Сглаженное значение St для момента времени t вычисляется как:

где Yt – наблюдаемое значение временного ряда в момент времени t, St-1 – сглаженное значение в предыдущий момент времени, α – коэффициент сглаживания (0 < α < 1), который определяет, насколько сильно новое значение влияет на сглаженное значение.

Для решения задачи инверсной кинематики методом сглаживания необходимо выполнить следующие шаги:

1. Определить текущее положение конечного сустава (xcurr, ycurr) и позицию, к которой стремится конечный сустав – (xtarget, ytarget).
2. Составить матрицу Якоби:
3. Определить ошибку между текущим положением конца манипулятора и целевой точкой:
4. Воспользоваться псевдообратной матрицей Якоби для улучшения устойчивости вблизи сингулярных конфигураций:

где – транспонированная матрица Якоби, I – единичная матрица, λ – коэффициент сглаживания, обычно выбирается малым (0,01 или 0,001).

1. Изменение в углах сутавов:

где – изменения углов суставов.

Углы суставов обновляются с учетом найденных именений:

1. Процесс повторяется до тех пор, пока величина ошибки не станет достаточно малой, или пока не будет достигнуто максимальное количество итераций.

Одним из ключевых преимуществ является его простота и легкость в применении. Он не требует сложных расчетов или больших объемов информации, что делает его идеальным для быстрого использования в условиях ограниченных ресурсов. Кроме того, данный метод акцентирует внимание на более свежих данных, что особенно полезно для прогнозирования, так как актуальная информация чаще всего лучше отражает текущие тенденции. Еще одним плюсом является гибкость метода: различные варианты экспоненциального сглаживания, такие как метод Хольта или Хольта-Винтерса, позволяют учитывать как тренды, так и сезонные изменения.

Тем не менее, у данного метода имеются и свои недостатки. Один из них заключается в его ограниченной способности обрабатывать временные ряды со сложными, нерегулярными паттернами или резкими изменениями. Экспоненциальное сглаживание эффективно работает с плавными трендами, но может оказаться неэффективным в случаях, когда данные содержат значительный шум или внезапные скачки. Кроме того, существует проблема правильного выбора параметра сглаживания (альфа). Неправильный выбор этого параметра может привести как к избыточному сглаживанию и утрате важных деталей, так и к высокой чувствительности прогноза к случайным колебаниям в данных.

Генетические алгоритмы — это класс эвристических методов оптимизации, которые имитируют процесс естественного отбора, свойственный биологической эволюции. Генетические алгоритмы основаны на таких концепциях, как наследственность, мутация, селекция и скрещивание (кроссовер), и применяются для поиска глобальных оптимумов в сложных многомерных задачах.

Генетический алгоритм представляет возможные решения задачи как "особи" или "хромосомы" и эволюционирует их популяцию через несколько поколений, чтобы найти наилучшее решение. Алгоритм использует принципы естественного отбора для сохранения лучших решений и модифицирует их для получения еще лучших результатов в следующих поколениях.

Этапы работы генетического алгоритма:

1. Генерация начальной популяции возможных решений. Каждое решение кодируется в виде хромосомы — последовательности чисел или битов, представляющих переменные задачи.
2. Для каждой особи (решения) вычисляется значение функции приспособленности (fitness function), которая оценивает, насколько это решение приближает к оптимуму задачи. Цель состоит в том, чтобы максимизировать или минимизировать значение этой функции.
3. Выбор особей, которые будут участвовать в создании следующего поколения. Наиболее приспособленные особи имеют больше шансов быть выбраны для размножения. Один из популярных методов селекции — турнирный отбор, при котором несколько случайных особей сравниваются, и лучшая из них выбирается для скрещивания.
4. Скрещивание, процесс, в ходе которого две родительские хромосомы комбинируются для создания новых потомков. Кроссовер может быть одноточечным (разделение родительских хромосом в одной точке) или многоточечным (разделение в нескольких точках), что позволяет передавать часть информации от каждого родителя потомку.
5. Для поддержания разнообразия в популяции некоторые потомки подвергаются случайным изменениям. Мутации помогают избежать преждевременного застревания алгоритма в локальных оптимумах.
6. После скрещивания и мутаций потомки добавляются в популяцию, и процесс продолжается. На каждом этапе слабые особи могут быть исключены из популяции.
7. Алгоритм продолжает эволюцию, пока не достигнет критерия остановки, например: максимальное количество поколений, достижение заданной точности решения, стабильность популяции (отсутствие значительного улучшения в течение нескольких поколений).

Для решения задачи инверсной кинематики методом сглаживания необходимо выполнить следующие шаги:

1. У каждой особи генотип (хромосома), состоит из двух параметров – углов .
2. Оценка приспособленности:
3. На этапе отбора выбираются наиболее приспособленные особи для размножения.
4. Создаются новые особи(потомки) путем комбинирования генов (углов суставов) двух родителей.
5. Мутация позволяет вносить случайные изменения в особи для поддержания разнообразия популяции.
6. После отбора, скрещивания и мутации создается новая популяция, которая заменяет предыдущую.
7. Процесс повторяется, пока не найдена особь с минимальной ошибкой или не достигнуто максимальное количество итераций.

Одним из ключевых достоинств данного подхода является его умение находить качественные приближённые решения для задач с множеством локальных минимумов, где традиционные методы, такие как градиентный спуск, могут "застревать". Генетический алгоритм не требует, чтобы оптимизируемая функция была непрерывной или дифференцируемой, что делает его универсальным инструментом для различных задач. Используя операторы мутации, кроссинговера и селекции, этот метод охватывает обширные области пространства решений и эффективно их исследует, избегая преждевременной сходимости.

Еще одним достоинством является параллельный характер генетического метода: он одновременно анализирует множество вариантов (популяцию), что повышает шансы на нахождение глобального оптимума. Это качество особенно актуально при работе с обширными параметрическими пространствами или сложными нелинейными функциями. Кроме того, метод легко настраивается для решения различных типов задач, как дискретных, так и непрерывных, и может быть объединен с другими алгоритмами для повышения своей эффективности.

Тем не менее, у генетического метода имеются и серьезные недостатки. Во-первых, он может быть достаточно медленным, поскольку требует множество итераций с различными возможными решениями. Это приводит к значительным затратам вычислительных ресурсов, особенно при решении сложных задач с большим количеством параметров. Во-вторых, генетический метод не всегда обеспечивает нахождение точного глобального оптимума, особенно если параметры алгоритма (размер популяции, вероятность мутаций и кроссинговера) настроены неправильно. В таких ситуациях метод может привести к субоптимальным решениям.

Еще одним минусом является высокая чувствительность к настройке параметров. К примеру, слишком высокая вероятность мутации может привести к беспорядочному поиску без какой-либо структуры, в то время как слишком низкая вероятность может вызвать недостаток разнообразия в решениях. Кроме того, данный метод может оказаться неэффективным для задач, где требуется точное решение, так как он предоставляет скорее приближенные результаты.

**Метод роя частиц** (PSO, Particle Swarm Optimization) — это алгоритм оптимизации, основанный на имитации коллективного поведения (роевого интеллекта), наблюдаемого в природе, например, у стай птиц или косяков рыб. Алгоритм был предложен в 1995 году Джеймсом Кеннеди и Расселом Эберхартом и используется для решения задач глобальной оптимизации в многомерных пространствах.

Метод FABRIK (Forward And Backward Reaching Inverse Kinematics) — это один из популярных методов решения задачи инверсной кинематики, который основан на пошаговом приближении к целевой позиции путём прямого и обратного перемещения суставов. В отличие от других методов, FABRIK не использует матрицы Якоби или сложные вычисления, а использует чисто геометрические соображения.

Метод начинается с инициализации начальной позиции конечности (например, конечности руки) и определяет желаемую целевую позицию. Каждая часть (звено) конечности имеет определённую длину и начальное положение.

Метод FABRIK сначала перемещает исследуемые суставы (от конечной точки к более близким суставам) с целью добиться того, чтобы конечная точка переместилась как можно ближе к заданной цели. Это достигается путем выравнивания суставов на основе их расстояний.

где — позиция i-го звена, — длина сегмента.

После обратной работы происходит прямая работа, где суставы корректируются в противоположном направлении, чтобы сохранить длину звеньев и гарантировать, что конечность остается ограниченной в своих движениях, соблюдая заданные ограничения.

Процесс повторяется для достижения заданной целевой позиции с достаточно высокой точностью. Итерации продолжаются до тех пор, пока конечная точка не будет достаточно близка к цели или не будет достигнуто максимальное количество итераций.

Одним из основных преимуществ метода FABRIK является его высокая эффективность. Алгоритм, благодаря своей итеративной структуре, может быстро достигать нужной конфигурации. Кроме того, он обладает простой реализацией, что делает его доступным для внедрения в различные проекты. FABRIK также обеспечивает качественные результаты, создавая плавные и естественные движения, что является критически важным для анимации персонажей и симуляций. Гибкость алгоритма позволяет адаптировать его под различные задачи и условия, что делает его универсальным инструментом для разработчиков в области анимации и кинематики.

FABRIK обладает рядом недостатков, которые могут повлиять на его эффективность и применимость. Во-первых, его производительность сильно зависит от начальной конфигурации суставов. Если начальная позиция конечной точки далеко от желаемого положения, это может привести к большому числу итераций, необходимых для достижения цели, что негативно сказывается на скорости работы алгоритма.

Кроме того, FABRIK может сталкиваться с проблемой локальных минимумов. Это значит, что алгоритм может застрять в конфигурации, где дальнейшие изменения суставов не приводят к улучшению, что требует дополнительных корректировок и усложняет выполнение задачи.

Другим важным аспектом является ограничение движений суставов. Алгоритм может не находить решения, если суставы имеют жесткие ограничения, что может потребовать ручных изменений или альтернативных методов для достижения желаемого результата.

Наконец, в сложных системах, например, при моделировании тела с высокой детализацией, метод может давать нестабильные результаты из-за коллизий между звеньями. Эта несоответствие часто требует дополнительных оптимизаций, чтобы обеспечить естественное и правильное движение.

Кватернионы – это математическая структура, которая представляет собой расширение комплексных чисел в четырёхмерное пространство. Они используются для описания вращений в трёхмерном пространстве, обеспечивая компактное и эффективное представление ориентации без некоторых проблем, связанных с другими методами, такими как углы Эйлера. Кватернионы имеют вид , где w, x, y, z – вещественные числа, а i, j, k – мнимые единицы, подчиняющиеся специфическим правилам умножения. Чаще всего кватернион записывают в виде , где w – скалярная часть, а v⃗ – векторная часть (x, y, z).

В задачах инверсной кинематики кватернионы применяются для представления и вычисления вращений суставов, а также для определения целевых ориентаций конечности. Кватернионы обеспечивают точное и стабильное управление ориентацией конечностей без артефактов, что делает их предпочтительным выбором для работы с роботизированными манипуляторами, реабилитационными устройствами и анимацией в компьютерной графике. Их применение упрощает вычисления и позволяет создавать плавные движения, что особенно важно для моделирования движений человека.

Кватернионы обладают рядом значительных преимуществ, которые делают их идеальным инструментом для работы с вращениями в трёхмерном пространстве. Одним из главных достоинств является их способность избегать проблемы гимбальной блокировки, с которой сталкиваются углы Эйлера. Это означает, что кватернионы могут корректно описывать вращения даже в сложных конфигурациях, не теряя степени свободы. Кроме того, они представляют вращения в компактной форме, используя всего четыре параметра, что делает их более вычислительно эффективными по сравнению с матрицами вращения, которые требуют девяти параметров. Кватернионы также обеспечивают высокую стабильность при множественных последовательных вращениях, поскольку они менее подвержены накоплению ошибок. Ещё одним важным преимуществом является лёгкость интерполяции. Сферическая линейная интерполяция (SLERP) позволяет плавно изменять ориентацию без артефактов, что особенно важно для анимации и управления движениями.

Однако у кватернионов есть и недостатки. Одной из главных трудностей является их не интуитивность. Поскольку они работают в четырёхмерном пространстве, их сложнее интерпретировать и визуализировать по сравнению с углами Эйлера. Кватернионы требуют постоянной нормализации, так как только единичные кватернионы корректно описывают вращения, что накладывает дополнительные вычислительные затраты. Кроме того, существует сложность в преобразовании между кватернионами и более привычными углами, такими как углы Эйлера, что может вызвать дополнительные трудности в определённых задачах. Эти ограничения, хотя и менее значительны, требуют учитывать особенности применения кватернионов в зависимости от конкретных потребностей и условий задачи.

Метод наименьших квадратов — это статистический метод, применяемый для нахождения функции (обычно линейной или нелинейной), которая лучше всего приближает набор данных. Его цель — минимизировать сумму квадратов отклонений между наблюдаемыми значениями yi и значениями, предсказанными моделью .

Функция, минимизируемая методом наименьших квадратов:

Для решения задачи инверсной кинематики методом наименьших квадратов необходимо выполнить следующие шаги:

1. Определить целевую точку в пространстве и вычисляемую точку , определяемую через функции кинематики:

где - вектор параметров суставов (например, углы поворота).

1. Определить функцию ошибки:
2. Линейное приближение через Якобиан:

Для итерационного решения используется разложение в ряд Тейлора:

где - якобиан (матрица частных производных).  
Вектор изменения параметров:

где , а - псевдообратная матрица (рассчитывается с использованием метода наименьших квадратов).

1. Продолжить процесс до тех пор, пока ошибка между текущей и целевой позицией не станет меньше заданного значения.

Метод наименьших квадратов имеет множество преимуществ, которые делают его одним из наиболее распространенных и часто применяемых способов в статистике, анализе данных и оптимизации. Его главное достоинство заключается в том, что он предлагает легкое и интуитивно понятное решение для минимизации расхождений между фактическими и предсказанными значениями, что делает его идеальным для задач линейной регрессии. Кроме того, данный метод отличается математической строгостью и позволяет находить аналитическое решение для линейных моделей, что упрощает его использование и обеспечивает высокую эффективность вычислений. Метод устойчива к случайным шумам в данных, при условии, что ошибки распределены нормально, из-за чего он часто принят в качестве стандартного метода для анализа данных.

Тем не менее, у метода имеются свои недостатки. Одним из основных является чувствительность к выбросам, так как сумма квадратов ошибок присваивает непропорционально высокий вес значительным отклонениям. Это может вызывать серьёзные искажения в результатах, если в данных присутствуют выбросы. При использовании нелинейных моделей сложность метода возрастает: аналитическое решение невозможно, и требуется прибегать к численным методам, которые могут быть менее точными и более ресурсозатратными. Кроме того, метод основывается на предположении о нормальном распределении ошибок и одинаковой дисперсии, что не всегда соответствует реальным данным. При нарушении этих предположений точность и надёжность получаемых результатов могут существенно ухудшиться.

* 1. Анализ средств моделирования движений человека

AnyBody Modeling System (ABMS) – это программное обеспечение для биомеханического моделирования, которое позволяет создавать подробные виртуальные модели человеческого тела и анализировать их движения. Его используют в различных областях, в том числе в медицине, спорте, эргономике и других.

Основная идея ABMS – предоставить инструмент для создания анатомически точных моделей скелета, мышц и других тканей человеческого тела. Пользователи могут варьировать параметры модели, такие как длина и мышечная масса, и анализировать, как эти изменения влияют на движения тела.

Система работает на основе принципов обратной динамики, что означает, что она может рассчитывать силы, вызывающие движения, и то, как эти силы влияют на структуры тела. Это особенно полезно при моделировании движений при различных физических нагрузках или при решении реабилитационных задач.

ABMS также интегрируется с данными измерения трафика из систем проверки трафика (таких как Vicon или Motion Analysis), чтобы повысить точность моделей и результатов анализа.

Это программное обеспечение широко используется в исследованиях и разработках, связанных с биомеханикой человеческого тела, и дает возможность более глубокого понимания движения и его влияния на физиологию.

ABMS применяет оптимизационные методы для решения задач инверсной кинематики. Главный принцип заключается в определении угловых перемещений суставов, которые минимизируют расхождение между заданной конечной позицией и реальным положением конечностей модели. Для достижения этой цели система использует метод наименьших квадратов, что позволяет учитывать различные ограничения и уменьшать разницу между желаемыми и фактическими положениями.

ABMS создает модели, основываясь на уравнениях равновесия сил и моментов, а также на кинематических зависимостях между частями тела.

Software for Interactive Musculoskeletal Modeling (SIMM) – программное обеспечение, предназначенное для интерактивного биомеханического моделирования мышц и скелета. Разработан в Институте нейронаук в Сан-Диего. Программа позволяет исследователям создавать подробные компьютерные модели человеческого тела с целью анализа и моделирования движений.

SIMM основан на концепции создания виртуальных 3D-моделей анатомии, включая кости, суставы и мышцы. Пользователи могут взаимодействовать с моделями, чтобы изменять параметры мышц, исследуя, как эти изменения влияют на движения и динамику тела.

Одним из ключевых элементов SIMM является его способность предоставлять подробные данные о моментах силы и требованиях к силе в различных суставах. Это полезно для анализа биомеханики движения и понимания того, как различные факторы влияют на нагрузку на мышцы и суставы.

SIMM также предоставляет инструменты интеграции с данными о движении, позволяя пользователям комбинировать биомеханические модели с экспериментальными данными для более точного анализа. Программа может быть использована в медицинских исследованиях, реабилитационной медицине и создании протезно-ортопедических изделий.

В целом SIMM предоставляет ученым и инженерам мощный инструмент для изучения биомеханики человеческого тела и его движений, а также для разработки улучшенных методов лечения и реабилитации.

SIMM применяет метод псевдообратных матриц для решения задач инверсной кинематики. В данном методе вычисляется матрица Якоби, которая устанавливает связь между изменениями углов суставов и перемещениями конечностей. После этого используется псевдообратная матрица Якоби для определения приближенных значений углов суставов, которые минимизируют расхождение с заданной конечной позицией.

Open Simulation for Musculoskeletal Systems (OpenSim) – открытая система моделирования опорно-двигательного аппарата, предоставляющая инструменты для создания детальных вычислительных моделей человеческого тела. Разработан Национальными институтами здравоохранения (NIH) и Стэнфордским университетом.

Основная цель OpenSim – предоставить исследователям и инженерам инструменты для анализа биомеханики движений человека и понимания взаимодействия мышц, костей и суставов. Система позволяет создавать трехмерные модели анатомии, включая скелет, мышцы и другие ткани.

Одним из главных преимуществ OpenSim является его открытость и доступность для научного сообщества. Это позволяет исследователям вносить свой вклад, улучшать систему и обмениваться моделями и данными. OpenSim активно используется в медицинских исследованиях, биомеханике и разработке протезно-ортопедических изделий.

Система позволяет проводить виртуальные эксперименты с моделями, изменять параметры мышц и суставов, чтобы анализировать, как эти изменения влияют на движение и нагрузку на организм. OpenSim также интегрируется с данными движения, позволяя пользователям сравнивать моделирование с реальными данными.

Открытость и гибкость OpenSim делают его важным инструментом для тех, кто занимается биомеханическими и реабилитационными исследованиями. Система поощряет коллективное участие и обмен знаниями, способствуя развитию этой области науки и применению полученных знаний в практике здравоохранения.

OpenSim применяет метод наименьших квадратов для решения проблемы инверсной кинематики. Этот подход направлен на минимизацию расхождений между желаемыми и реальными положениями или ориентациями контрольных точек (например, маркеров) как в модели, так и в экспериментальных данных.

В OpenSim задача представляется в виде оптимизационной: система находит такие угловые значения суставов, которые уменьшают общую квадратичную ошибку между экспериментальными и моделируемыми позициями маркеров, принимая во внимание установленные ограничения на движения суставов.

Blender — мощное бесплатное программное обеспечение для 3D-моделирования, анимации, рендеринга, композитинга, создания игр и многого другого. Среди его многочисленных возможностей важное место занимает возможность имитировать движения человека.

Одна из лучших особенностей Blender — это то, что он бесплатный и имеет открытый исходный код, что делает его доступным для широкого круга пользователей. Blender предоставляет инструменты для создания персонажей, анимации и 3D-сцен, что делает его полезным инструментом для моделирования движений человека в контексте реабилитации.

В Blender вы можете создавать гуманоидных персонажей, размещать их в разных позах и анимировать их движения. Поддерживая различные форматы данных, Blender может взаимодействовать с данными о движении, полученными из различных источников, таких как системы захвата движения.

Для анализа движений в Blender можно использовать инструменты, позволяющие изменять параметры скелета персонажа, настраивать анимацию и даже проводить виртуальные эксперименты по анализу влияния различных факторов на движения.

Важным аспектом Blender является сообщество пользователей и большое количество обучающих программ. Это облегчает изучение программы и обмен опытом с другими специалистами в области 3D-моделирования и анимации. Таким образом, Blender предоставляет среду, в которой исследователи и практики могут создавать и анализировать движения человека, применяя их в контексте реабилитации.

Blender применяет метод, основанный на вычислении псевдообратных матриц, в сочетании с итеративными алгоритмами, такими как метод сглаженной градиентной оптимизации, для решения задач инверсной кинематики.

В частности, в Blender используется метод Дампеда псевдообратной матрицы Якоби (Damped Least Squares), который обеспечивает стабильность решения и позволяет справляться с ситуациями, когда стандартная псевдообратная матрица может приводить к нестабильным или неверным результатам (например, вблизи сингулярностей). Этот подход помогает Blender точно управлять позами сложных систем костей, обеспечивая реалистичное движение и минимизируя расхождения между желаемыми и фактическими позициями конечностей модели.

Unity представляет собой мощный инструмент для разработки, который позволяет создавать и моделировать движения человека с помощью различных инструментов и пакетов. Для этих задач в основном применяются такие расширения, как Unity Animation Rigging, Unity Mecanim и Unity Humanoid Animation System.

Unity Animation Rigging — это инструмент для анимации в Unity, который предоставляет разработчикам возможность в реальном времени создавать и изменять движения персонажей. Он позволяет аниматорам задавать ограничения для костей, включая IK (обратная кинематика) и FK (прямая кинематика), а также управлять физическими взаимодействиями между конечностями и окружающей средой. Это способствует созданию правдоподобных движений персонажей, принимая во внимание их взаимодействие с объектами на сцене и с землей.

Unity Mecanim — это система, предназначенная для анимации персонажей, особенно эффективная в проектах, где необходимо воспроизводить человеческие движения. Mecanim применяет сложные слои и параметры, что позволяет создавать адаптивные и настраиваемые анимации. Разработчики имеют возможность настраивать переходы между анимациями, связывая их с событиями и переменными для более детального управления поведением персонажа.

Система анимации Humanoid от Unity обеспечивает анатомически точное моделирование движений. Она дает возможность загружать и применять готовые скелетные модели или данные о движениях, полученные с помощью систем, таких как Vicon или OptiTrack. Применяя данные захвата движений, можно достичь высокой степени реалистичности, особенно в сложных ситуациях, таких как спортивные тренировки или процессы реабилитации.

В Unity для решения задач инверсной кинематики (IK) применяется несколько подходов, среди которых наиболее распространённым является итеративный метод FABRIK (Forward and Backward Reaching Inverse Kinematics). Этот метод эффективно справляется с задачами IK, принимая во внимание ограничения по длине сегментов и обеспечивая плавное и естественное движение.

FABRIK функционирует итеративно, последовательно перемещаясь вперед и назад по цепочке костей, чтобы достичь необходимой позиции конечности, минимизируя разницу между заданным и текущим положением. Данный метод часто используется для управления конечностями, такими как руки и ноги, благодаря своей стабильности и простоте реализации.

* 1. Обоснование метода решения

Для решения задач инверсной кинематики в системах с высоким числом степеней свободы, таких как моделирование движений человека, необходимо использовать эффективные методы, которые могут адекватно моделировать сложные движения. Одним из таких методов является применение псевдообратной матрицы Якоби, обладающей рядом важных преимуществ для подобных систем. Главным достоинством этой матрицы является то, что она позволяет решать задачи инверсной кинематики даже в случаях, когда система имеет избыточные или недоопределённые параметры. Это означает, что количество степеней свободы может быть как больше, так и меньше, чем требуется для достижения заданного положения. Данный метод обеспечивает оптимальное приближение в рамках метода наименьших квадратов и может быть использован, когда стандартная матрица Якоби не имеет обратной. Таким образом, псевдообратная матрица Якоби позволяет эффективно распределять движения между степенями свободы, что способствует достижению целевого положения с минимальными отклонениями.

Тем не менее, применение псевдообратной матрицы Якоби имеет свои недостатки. Одним из них является шум, который может возникать во время итерационных процессов. Этот шум возникает из-за того, что небольшие изменения в исходных данных могут вызывать значительные колебания в движениях, особенно если система подвержена числовым ошибкам или если сингулярные значения в разложении матрицы Якоби находятся близко к нулю. Это приводит к тому, что движения становятся резкими и нестабильными, что критично для задач, где важны плавность анимации и точность управления.

Для решения этой задачи часто применяется метод сглаживания, который помогает снизить влияние шумов и стабилизировать движения. Экспоненциальное сглаживание считается одним из самых эффективных методов, так как оно фокусируется на последних изменениях и уменьшает значимость более старых данных. Это позволяет системе оставаться чувствительной к изменениям в исходных параметрах, одновременно смягчая резкие колебания, вызванные ошибками в расчетах. В результате метод сглаживания помогает уменьшить резкие скачки в управлении, делая движения более предсказуемыми и плавными.

Кроме того, для повышения эффективности решения задач инверсной кинематики может использоваться представление вращений в виде кватернионов. Кватернионы являются удобным инструментом для описания трёхмерных вращений, поскольку они избегают проблем, связанных с особенностями других форм представления, таких как матрицы поворотов или углы Эйлера. Одним из ключевых преимуществ кватернионов является их способность избегать явления “захвата” (“gimbal lock”), которое часто возникает при использовании углов Эйлера. Кроме того, кватернионы обеспечивают более компактное и вычислительно эффективное представление вращений, что особенно важно для систем с высоким числом степеней свободы.

В контексте инверсной кинематики кватернионы могут быть использованы для точного управления ориентацией конечных звеньев системы. Это достигается путём оптимизации целевой функции, учитывающей как положение, так и ориентацию объектов. Применение кватернионов также упрощает интерполяцию вращений, что делает их незаменимым инструментом в анимации и робототехнике, где требуется плавное и реалистичное движение. В сочетании с псевдообратной матрицей Якоби кватернионы позволяют значительно улучшить точность и стабильность управления движением, минимизируя ошибки и обеспечивая плавность траекторий.

В конечном итоге, использование псевдообратной матрицы Якоби, методов сглаживания и представления вращений с помощью кватернионов представляет собой мощный набор инструментов для решения задач инверсной кинематики. Такой подход обеспечивает точность, адаптивность и стабильность в сложных системах с избыточными или недоопределёнными степенями свободы, делая движения более плавными, предсказуемыми и реалистичными.

Выводы по главе 1

В ходе исследования были произведены следующие работы:

* поставлена цель и задачи исследования;
* определен объект исследования;
* произведен аналитический обзор существующих решений;
* проанализированы основные направления разработки.

ГЛАВА 2. ФОРМАЛИЗОВАННОЕ ПРЕДСТАВЛЕНИЕ ПРОЦЕССА МОДЕЛИРОВАНИЯ ДВИЖЕНИЙ ЧЕЛОВЕКА

2.1. Формализованное представление задачи математического моделирования движений

Для задачи математического моделирования движений объектом движения является человеческое тело. Анатомически оно состоит из различных частей, включая конечности (руки и ноги), туловище, голову и шею, каждая из которых имеет свои особенности и суставы. Суставы, которые учитываются в моделировании представлены на рис. 2.1.

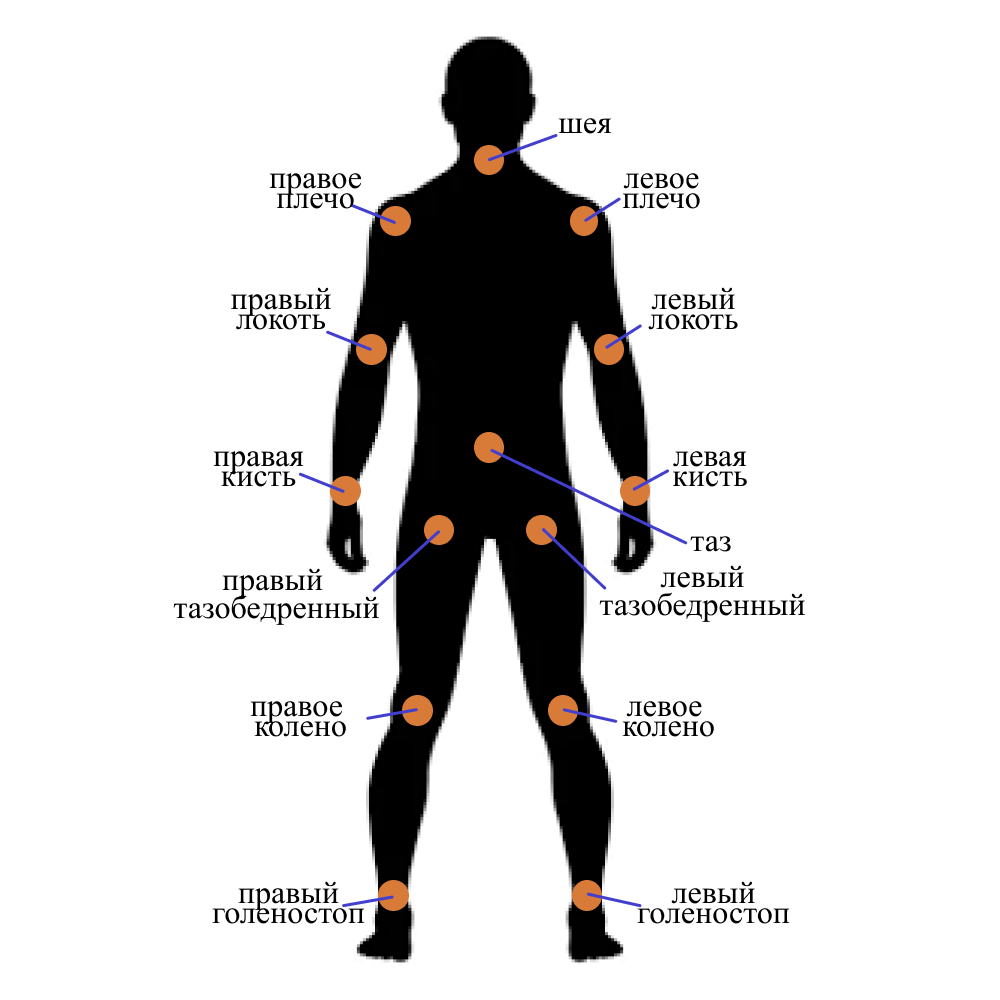


Рис 2.1. – Суставы для моделирования

Также для моделирования необходимы расстояния между суставами (рисунок 2.2).

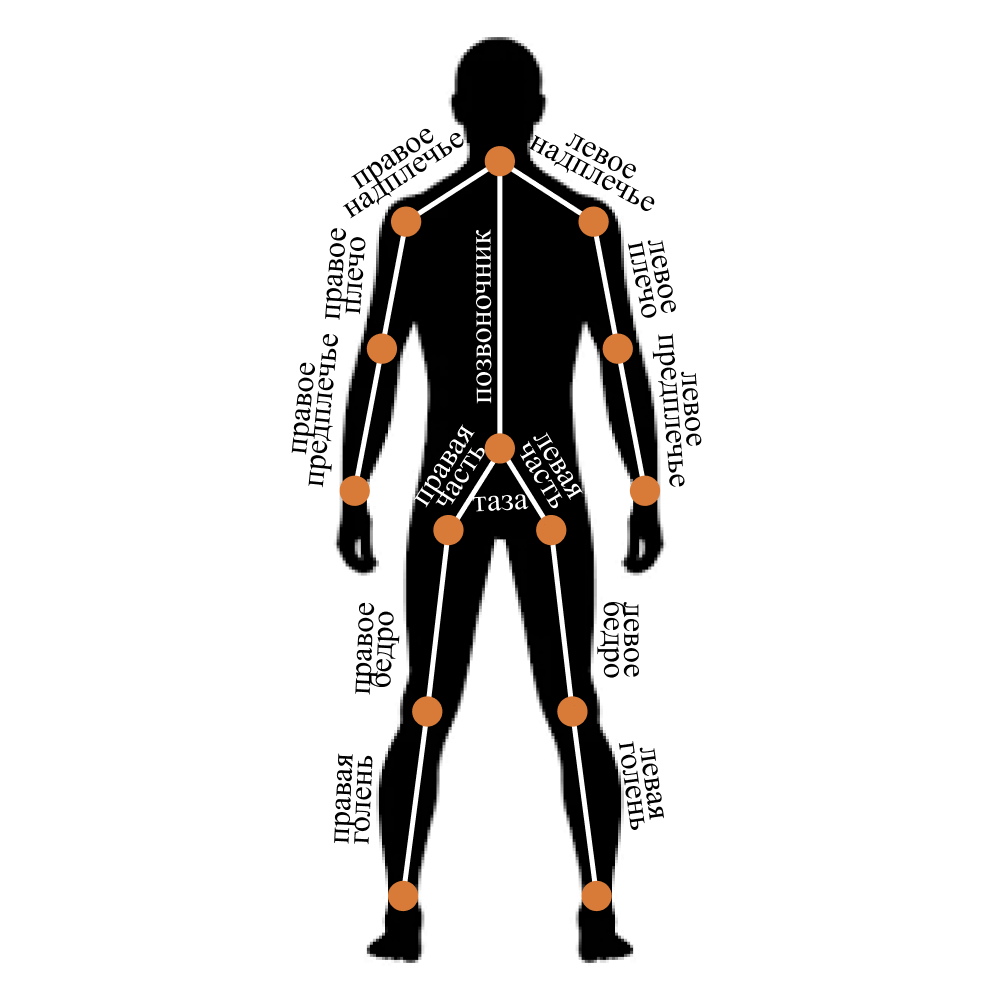


Рис 2.2. – Расстояния от одного сустава до другого

Задача инверсной кинематики заключается в том, чтобы по заданным координатам и ориентации конечного сустава, необходимо найти углы поворота всех остальных суставов.

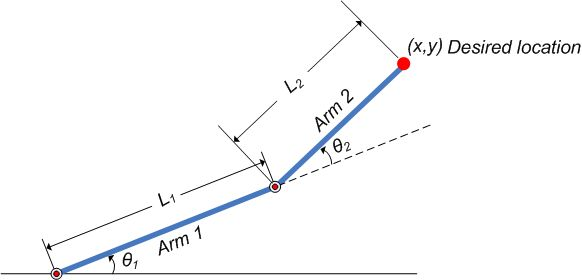


Рис 2.3. – Задача инверсной кинематики

На примере трех суставов рассмотрим алгоритм моделирования плавных движений (рисунок 2.3).

Уравнения прямой кинематики в трехмерном пространстве имеют вид:

, (1)

где L1 и L2 – длины сегментов руки (Arm 1, Arm 2), θ1 и θ2 – углы поворота суставов, θ3 – угол вращения вокруг оси Z.

Для начала определим текущее положение конечного сустава (xcurr, ycurr,zcurr). Позиция, к которой стремится конечный сустав – (xtarget, ytarget, ztarget).

Затем составим матрицу Якоби, которая описывает, как изменение углов θ1, θ2 и θ3 влияет на изменение положения конца манипулятора:

Определяем ошибку между текущим положением конца манипулятора и целевой точкой:

Воспользуемся псевдообратной матрицей Якоби для улучшение устойчивости вблизи сингулярных конфигураций. Для этого введем параметр сглаживания λ. Это изменяет стандартную форму псевдообратной матрицы:

где – транспонированная матрица Якоби, I – единичная матрица, λ – коэффициент сглаживания, обычно выбирается малым (0,01 или 0,001).

После нахождения сглаженной псевдообртной матрицы , находим изменения в углах сутавов.

где – изменения углов суставов.

Углы суставов обновляются с учетом найденных именений:

Процесс повторяется до тех пор, пока величина ошибки не станет достаточно малой, или пока не будет достигнуто максимальное количество итераций.

2.2. Разработка методики моделирования движений человека

2.3. Разработка алгоритма моделирования движений человека

Алгоритмы, использующие инверсную кинематику, находят широкое применение в симуляции движений человека, особенно в областях реабилитации, робототехники и анимации. Одним из важных процессов при решении таких задач является применение псевдообратной матрицы Якоби. Этот метод позволяет вычислять изменения углов суставов, необходимые для того, чтобы конечность достигла заданного положения. Для повышения стабильности и точности алгоритма используется метод сглаживания, который вводит регуляризацию, помогающую избежать неопределенности решений, возникающей при вырожденности матрицы Якоби. Данный подход особенно эффективен при работе с многосуставными моделями, где конфигурация системы может быть неопределенной.

На рисунке 2.4 представлена блок-схема работы программы моделирования движений человека.

Работа программы начинается с инициализации, которая включает подключение к базе данных для загрузки параметров модели, таких как длины сегментов, начальные углы суставов и целевые координаты. На этом этапе также подключаются необходимые библиотеки, включая те, которые предназначены для выполнения численных расчетов и работы с базой данных.

На следующем этапе система определяет текущее местоположение манипулятора, применяя уравнения прямой кинематики. Координаты x, 𝑦 и z вычисляются исходя из актуальных значений углов суставов и геометрических характеристик конструкции. После определения текущей позиции, происходит ввод пользователя, который отображает конечную точку манипулятора.

Далее проводится расчет ошибки, которая отражает разницу между фактическим положением и заданной целью. Этот процесс важен для выявления необходимых корректировок углов суставов для достижения поставленной цели.

Затем строится матрица Якоби, которая описывает, как изменения углов суставов влияют на позицию конечности в пространстве. Для этого производятся вычисления частных производных координат x, y и z относительно каждого угла сустава. Полученная матрица впоследствии используется для обратного преобразования. На данном этапе применяется псевдообратная матрица Якоби с учетом сглаживания, что позволяет определить изменения углов суставов, минимизируя ошибку и обеспечивая стабильность решения.

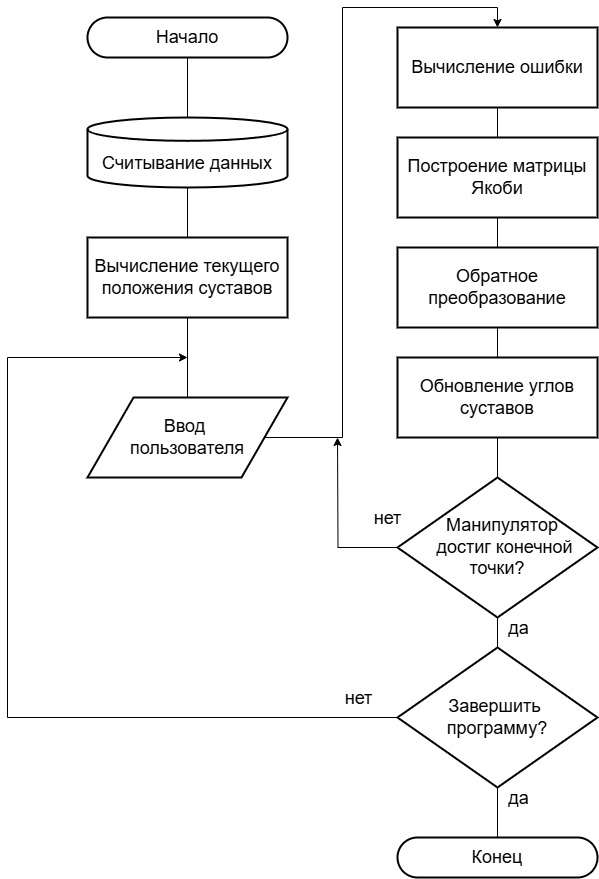


Рис 2.4. – Блок-схема

После того как корректировки рассчитаны, программа обновляет показатели углов суставов. Эти обновленные значения могут быть сохранены в базе данных для дальнейшего анализа или визуализации. Если целевая точка не достигнута, алгоритм вновь выполняет указанные шаги, начиная с расчета текущей позиции и ошибки. Этот процесс продолжается до достижения необходимой точности.

Программа завершает свою работу, когда конечность достигает заданной цели. Этот метод дает возможность эффективно воспроизводить движения человека, принимая во внимание индивидуальные характеристики кинематики и обеспечивая стабильное функционирование системы, даже в сложных конфигурациях.

Выводы по главе 2

В ходе исследования произведены следующие работы:

* выполнена формализация задачи математического моделирования движений;
* описана разработка методики моделирования движений человека;
* описана разработка алгоритма моделирования движений человека.

ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТАЛЬНОЕ ПОДТВЕРЖДЕНИЕ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ

* 1. Разработка алгоритма моделирования движений человека

Для взаимодействия с моделью человека на манипуляторы добавляется коллайдер. Этот коллайдер, помеченный тегом "Target", позволяет системе отслеживать, когда пользователь хватает его и перетаскивает при нажатии на левую клавишу мыши. Когда пользователь нажимает на манипулятор скрипт реагирует на это событие и запускает расчеты углов и расположения связанных суставов.

Листинг 3.1. Взаимодействие с моделью

if (Input.GetMouseButtonDown(0)) // Нажата левая кнопка мыши

{

Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);

if (Physics.Raycast(ray, out RaycastHit hit) && hit.collider.CompareTag("Target"))

{

target = hit.transform; // Сохраняем ссылку на объект

isDragging = true;

offset = target.position - GetMouseWorldPosition();

}

}

if (Input.GetMouseButtonUp(0)) // Отпущена левая кнопка мыши

{

isDragging = false;

target = null; // Сбрасываем ссылку

}

if (isDragging && target != null) // Проверка на null

{

Vector3 newPosition = GetMouseWorldPosition() + offset;

target.position = ApplyAxisConstraint(newPosition, target.position);

}

Для более удобного перемещения манипуляторов добавляется вращение камеры вокруг модели человека при нажатии на правую клавишу мыши.

Листинг 3.2. Вращение камеры вокруг модели

private void HandleCameraRotation()

{

if (Input.GetMouseButton(1)) // Вращение камеры при зажатой правой кнопке мыши

{

float horizontal = Input.GetAxis("Mouse X") \* rotationSpeed \* Time.deltaTime;

float vertical = Input.GetAxis("Mouse Y") \* rotationSpeed \* Time.deltaTime;

// Вращаем камеру вокруг фиксированного центра (pivotPoint)

mainCamera.transform.RotateAround(

pivotPoint,

Vector3.up,

horizontal

);

mainCamera.transform.RotateAround(

pivotPoint,

mainCamera.transform.right,

-vertical

);

// Обновляем направление камеры на центр

mainCamera.transform.LookAt(pivotPoint);

}

}

Также для более удобного перемещения добавлены кнопки для ограничения движений манипуляторов по двум плоскостям (рисунок 3.1).

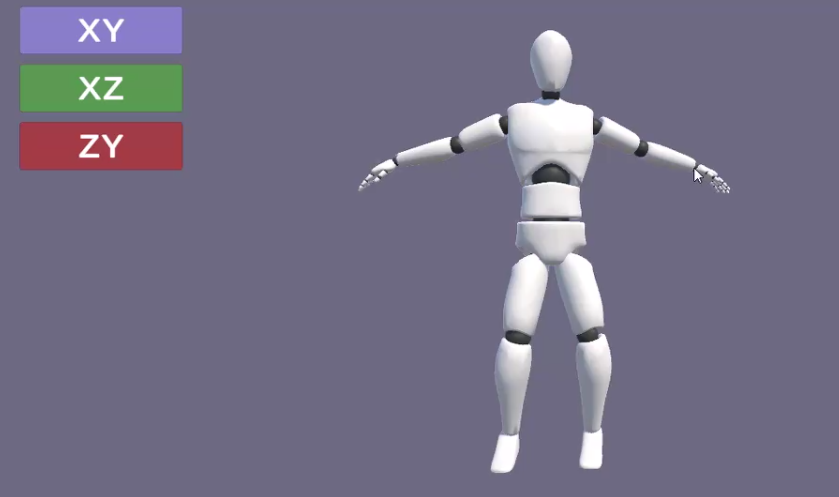


Рис 3.1. – Интерфейс программы

Для выполнения алгоритма добавлено вычисление матрицы Якоби и псевдообратной матрицы Якоби.

Листинг 3.2. Вычисление матрицы Якоби

Matrix4x4 CalculateJacobian()

{

// Матрица Якоби для 3 суставов

Matrix4x4 jacobian = new Matrix4x4();

Vector3 joint1Pos = Root.transform.position;

Vector3 joint2Pos = Mid.transform.position;

Vector3 joint3Pos = Tip.transform.position;

Vector3 arm1 = joint2Pos - joint1Pos;

Vector3 arm2 = joint3Pos - joint2Pos;

jacobian.SetRow(0, new Vector4(-arm1.y, -arm2.y, 0, 0));

jacobian.SetRow(1, new Vector4(arm1.x, arm2.x, 0, 0));

jacobian.SetRow(2, new Vector4(0, 0, 1, 0));

return jacobian;

}

Matrix4x4 PseudoInverse(Matrix4x4 jacobian, Matrix4x4 jacobianT, float lambda)

{

Matrix4x4 identity = Matrix4x4.identity;

Matrix4x4 jacobianProduct = jacobianT \* jacobian;

// Добавляем сглаживание

for (int i = 0; i < 4; i++)

{

jacobianProduct[i, i] += lambda;

}

// Инвертируем матрицу

Matrix4x4 inverse = jacobianProduct.inverse;

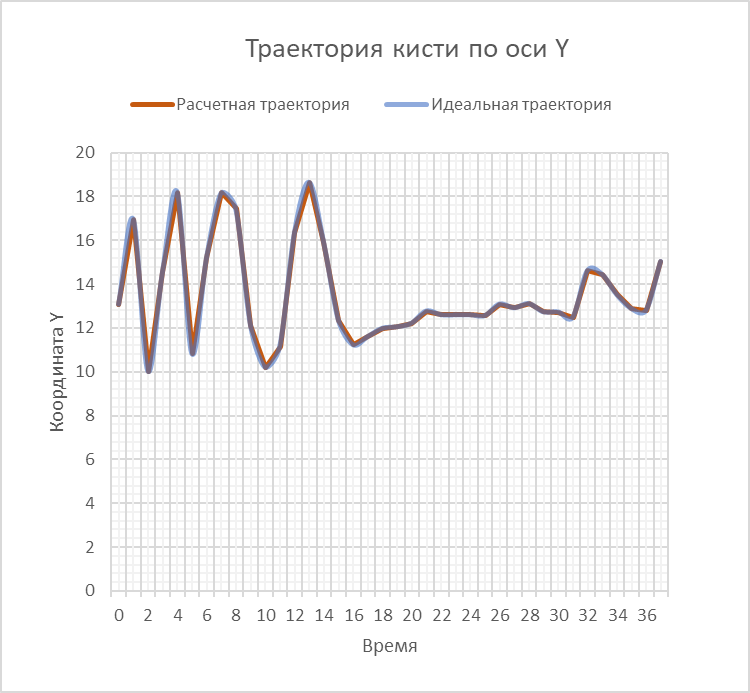
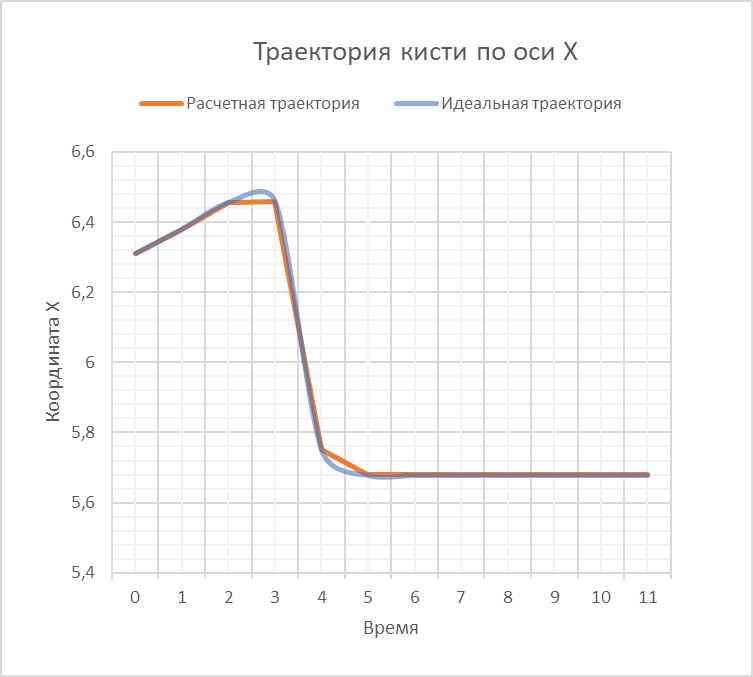
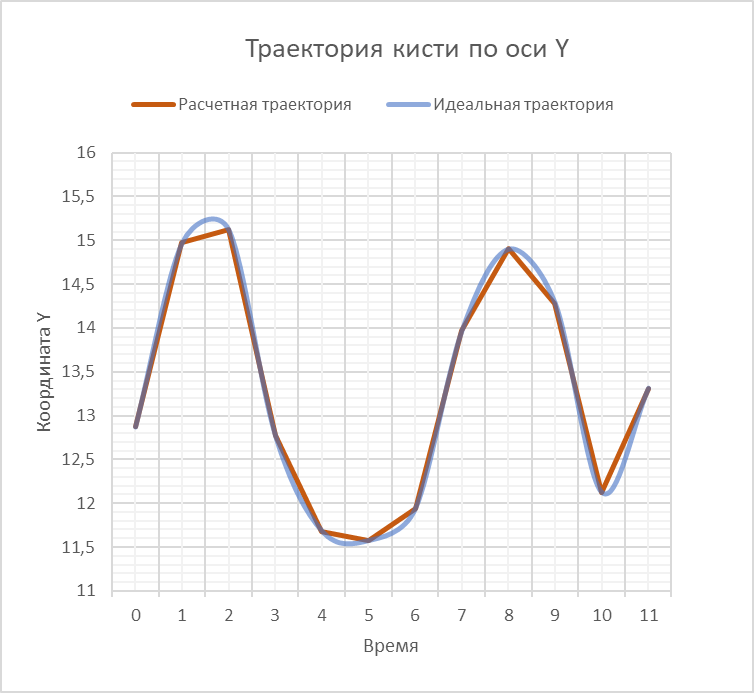
return inverse \* jacobianT;

}

Продолжение

* 1. Верификация и сравнение модели с помощью метрики качества (выбор метрики качества)

Результаты

* 1. Обоснование достоверности полученных результатов