

РАЗРАБОТКА ПРОГРАММЫ ПРЕОБРАЗОВАНИЯ ФОРМАТА ФАЙЛА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Соломыкин А.А.
E-mail: anokrid@mail.ru

Научный руководитель: Павлов В.М. доцент, к.т.н. Кафедра электроники и автоматики физических установок.

При снятии экспериментальных данных абсолютно различной формы зачастую встаёт вопрос об их хранении, структурировании, а также передачи и дальнейшем использовании. Для передачи данных с подсистем сбора данных (ПСД) используется X-файл. Данный файл является структурированным контейнером для хранения и перемещения данных с ПСД на серверы хранения и обработки данных. В контейнере, можно хранить следующие структуры данных:

- Одномерный массив данных зависимый от времени;
- Одномерный массив данных зависимый от переменной;
- Многомерный массив данных;
- Многомерный массив данных зависимых от времени;
- Мультимедиа данные.

Так как файл может хранить различные структуры данных, то доступ к ним становится затруднителен для программиста. Чтобы упростить обращение к данным, находящимся в файле, была реализована библиотека WorkXF. При использовании данной библиотеки упрощается не только доступ к данным, но и ускоряется процесс создания приложений обрабатывающих данные [1]. Сама библиотека WorkXF представляет из себя набор файлов, реализованных на языке C++. Ниже представлена структура X-файла, который можно обрабатывать и создавать с помощью вышеупомянутой библиотеки:

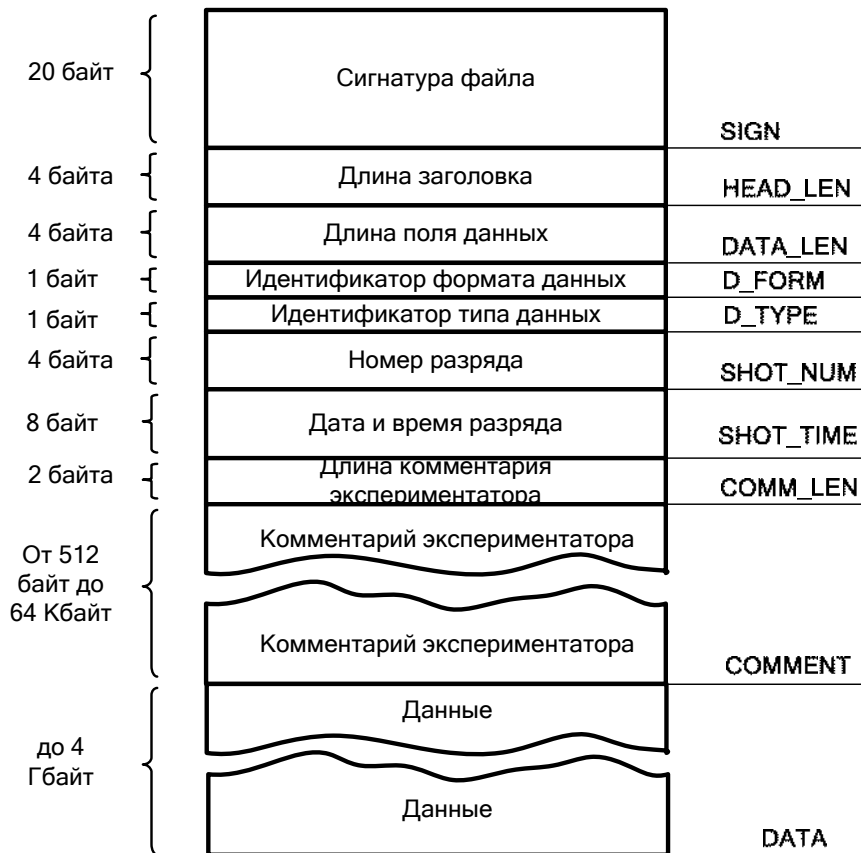


Рисунок 1. Структура файла, формируемого подсистемой сбора данных.

Получив и структурировав экспериментальные данные в специальный X-файл, встаёт вопрос о передаче и хранении непосредственно самих X-файлов, а также об удобном в дальнейшем их использовании. Решить данную задачу позволяет использование системы управления базы данных (СУБД). Из их достаточного количества была выбрана СУБД PostgreSQL.

PostgreSQL – это объектно-реляционная система управления базами данных, работающая как клиент-серверная система. Может возникнуть вопрос – почему именно PostgreSQL? Лучшее всего ответить на него, сравнив этот продукт с не менее популярным и, пожалуй, более известным – MySQL. Прежде всего следует отметить различия в организации данных. Если в MySQL каждая таблица заносится в собственный файл (для большинства типов БД), то PostgreSQL организует единую файловую структуру, в которой отдельные файлы не соответствуют непосредственно таблицам или другим объектам базы данных. То есть в MySQL вы можете создать резервную копию лишь части БД, сохранив соответствующие файлы. Для PostgreSQL такой номер пройдет только для всей структуры, включающей в себя все обслуживаемые сервером БД базы. Как следствие – MySQL полагается при организации БД на файловую систему ОС, в то время как PostgreSQL зависит от нее меньше, но требует дополнительного обслуживания – периодической дефрагментации базы данных [2].

PostgreSQL имеет более широкие возможности, как для обработки данных, так и для администрирования, но это, в свою очередь, несколько повышает сложность работы с этой СУБД. Таким образом, можно сказать, что место PostgreSQL – базы данных, требующие высокой степени надежности хранения информации, предъявляющие повышенные требования к проверке всех изменений, имеющие необходимость в автоматической корректировке большого числа данных при изменении информации в одной из таблиц, а также задачи, где требуется возможность разработки нетривиальных решений, использование нестандартных операторов и т. д [2].

Выбрав необходимую СУБД, можно приступить к главной задаче моей работы: разработка структуры базы данных для хранения экспериментальной информации и ее реализация в PostgreSQL, а также разработка и тестирование программы преобразования формата. В конечном счёте написанная мною программа должна будет позволять отправлять на хранение X-файлы на удаленную базу данных, реализованную на СУБД PostgreSQL, их структурированное хранение в зависимости от структуры данных, хранящихся в файле, а также должна быть обеспечена возможность удобного обращения к хранящейся информации по желаемым критериям (дата создания файла, структура файла, размер файла и т.п.).

Для структурирования данных принято использовать индексы. Индексом называется объект базы данных, позволяющий значительно повысить скорость обращения к базе за счет ускоренной обработки команд, содержащих сравнительные критерии. Хранимая в индексах информация о размещении данных по одному или нескольким полям таблицы повышает эффективность поиска записей при условной выборке [3].

Индексы ускоряют доступ к базе, но они также сопряжены с дополнительными затратами ресурсов. При изменении данных в поле индекс приходится обновлять, поэтому поддержание редко используемых индексов отрицательно сказывается на быстродействии системы (затраты времени на поддержание индексов превышают экономию от их использования). Как правило, индексы определяются только для полей, часто указываемых в условиях поиска [3].

Однако написание требуемой программы не является концом работы. Помимо этого требуется разработка руководства оператора и руководства программиста.

Руководство оператора необходимо для того, чтобы оператор (человек, который будет в дальнейшем работать с программой) мог спокойно пользоваться готовой программой, а при возникновении вопросов касательно её работы мог найти ответы в этом руководстве. Кроме того, руководство оператора должно позволить научиться работать с данной программой любому желающему, не имеющему начальных знаний по работе с ней. Также в руководстве оператора указаны возможные ошибки, причины их возникновения и способы устранения (само собой, что всё возникновение большинства ошибок должно быть предусмотрено заранее и должны быть приняты все меры, для недопущения их возникновения, но вероятность возникновения есть всегда, поэтому оператор должен быть способен самостоятельно устранить неполадки в работе программы).

Руководство программиста имеет несколько другие цели. Программист, желающий модифицировать программу, или же разобратся в принципе её работы, с помощью руководства программиста имеет возможность достичь своей цели, т.к. в руководстве программиста детально пояснены и расписаны основные моменты работы программы, принцип работы используемых функций, а также приведены сведения об используемых X-файлах и PostgreSQL.

Литература.

1. WorkXF руководство программиста, 2008г – 3-4с.
2. Супрунов С. PostgreSQL: первые шаги. – Журнал «Системный администратор», №7, 2004 г. – 26-27 с .
3. Уорсли Дж., Дрейк Дж. PostgreSQL. Для профессионалов. //СПб.: Питер, 2003.– 496 с:ил. – 210 с.

ФИЛЬТРАЦИЯ УСТОЙЧИВЫХ ПОЛОЖЕНИЙ ЧЕЛОВЕКОПОДОБНОГО РОБОТА

Шеломенцев Е.Е.

E-mail: see4me@mail.ru

Научный руководитель: Александрова Т.В., ассистент кафедры ИКСУ

Введение

Одним из недостатков метода управления антропоморфным роботом с помощью сенсора Kinect [1,2,3], является необходимость обеспечения стабильного вертикального положения в ходе работы алгоритма управления. Действительно, некоторые действия оператора, будь то резкое движение рукой или наклон, могут привести к дестабилизации робота и его повреждению. Немаловажным является устойчивость статических положений робота. В данной работе будет рассмотрен один из способов обеспечения стабильного вертикального положения робота в случае применения метода Kinect-управления.

Оценка стабильности положения

Существует несколько различных способов обеспечения стабильного вертикального положения робота, к которым относятся статическая стабилизация как с учетом внешних воздействий, так и без него. Наиболее распространенным методом решения задачи статической стабилизации является метод точки нулевого момента (ZMP – Zero Moment Point), позволяющий определить устойчиво ли положение робота или нет [4,5].

Суть метода заключается в вычислении точки ZMP, в которой суммы моментов по осям X и Y равны нулю (рис.1). Для оценки устойчивости положения робота необходимо определить принадлежит ли найденная точка области стопы, в случае нахождения робота в фазе с одиночной поддержкой, или на отрезке, соединяющего центры стоп, в случае двухопорной фазы. Ключевое значение имеет момент, создаваемый массой самого робота. Для того, чтобы вычислить этот момент необходимо определить точку приложения силы тяжести – координаты центра масс. Воспользуемся формулой:

$$\bar{r}_c = \frac{\sum_i m_i \bar{r}_i}{\sum_i m_i}, \quad (1)$$

где m_i – масса i -ой точки, \bar{r}_i – радиус-вектор i -ой точки, \bar{r}_c – радиус-вектор центра масс.

Без учета влияния внешних сил и сил инерции, точка нулевого момента является проекцией точки центра масс на опорную плоскость (рис.1). Действительно, в этом случае вектор силы тяжести пересекает оси X и Y, т.е. момент, создаваемый этой силой равен нулю. Таким образом, для оценки стабильности положения робота необходимо лишь определить лежит ли проекция точки центра масс в окрестности точки опоры.