

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ МОНИТОРИНГА СОСТОЯНИЯ ДОРОЖНОГО ПОКРЫТИЯ

© 2017 А.А. Якименко^{1,2}, А.А. Малявко¹, Д.А. Богомолов¹, А.Е. Морозов¹

¹ Новосибирский государственный технический университет

² Институт вычислительной математики и математической геофизики СО РАН, Новосибирск

Статья поступила в редакцию 11.12.2017

В статье процесс разработки инструмента мониторинга состояния дорожного покрытия, в частности решение проблем хранения и обработки данных. В статье утверждается актуальность разработки. Разработан прототип экспериментальной системы, работающей на основе данной платформы. Работа выполнена при поддержке гранта РФФИ 16-37-00240 мол. а.

Ключевые слова: мониторинг, нагрузка, PostgreSQL, PostGIS, платформа, веб-сервис, REST, API.

1. ВВЕДЕНИЕ

Согласно статистике, наибольшее количество дорожно-транспортных происшествий случается по причине неудовлетворительного состояния дорожного покрытия. В связи с этим всё большую значимость приобретает контроль состояния дорожного покрытия. Актуальность исследования определяется также потребностью города Новосибирска в инструменте стратегического планирования дорожно-ремонтных работ, в связи с участием региона в пилотном проекте комплексного развития транспортной инфраструктуры «Безопасные и качественные дороги» [4]. В качестве основных целей данной программы ставится задача по приведению к 2018 году в нормативное состояние не менее половины дорог агломерации (к 2025 году таких дорог должно быть не менее 85%), снижение к 2018 году число мест концентрации ДТП на дорогах агломерации до половины (к 2025 году количество аварийно-опасных участков должно сократиться до 10% от уровня 2016 года). Разработанная система позволяет в режиме реального времени отслеживать текущее качество дорожного покрытия и динамику образования неровностей на отдельных участках.

Целью данной работы является создание информационной системы, позволяющей отслеживать динамику образования неровностей в разное время года на различных участках дорог.

Данная система позволяет прогнозировать образование новых неровностей с помощью сбора статистических данных о характере неровностей (0-3, 0 – нет неровностей, 3 – глубокая яма). Она помогает более корректно получать данные о пробках на дорогах, оперативно выполнять ремонт дорожного покрытия, а также существенно сократить трату средств на ремонт автомобиля, так как его техническое состояние зависит от состояния пройденного дорожного покрытия. Для достижения цели поставлены следующие задачи: выбор СУБД для хранения данных, которая должна отвечать высоким требованиям к скорости работы, выбор и установка дополнительных библиотек для СУБД, разработка SQL-запросов на получение данных, тестирование разработанных запросов и сравнение времени потраченных на их выполнение у различных СУБД.

Общий алгоритм работы системы:

1. Анализ в реальном времени данных получаемых аппаратно-программным комплексом «PitBox», далее АПК «PitBox»
2. При нахождении АПК «PitBox» ощутимых неровностей, отправка данных с характеристической неровности на сервер
3. Отображение состояния дорожного покрытия, посредством выборки данных из СУБД и отображением в web-приложении

2. ПРИМЕНЯЕМЫЕ ТЕХНОЛОГИИ И УСТРОЙСТВА

Для реализации информационной части системы была выбрана технология построения веб-сервисов как RESTful API (REST, REpresentational State Transfer – архитектурный стиль взаимодействия между компонентами распределенного приложения в сети) [6]. Использование методологии разработки веб-сервисов позволяет разрабатывать различные инструменты, доступные через Интернет. Данная технология ис-

Якименко Александр Александрович, кандидат технических наук, доцент кафедры вычислительной техники.
E-mail: yakimenko@corp.nstu.ru

Малявко Александр Антонович, кандидат технических наук, доцент кафедры вычислительной техники.

E-mail: a.malyavko@corp.nstu.ru

Богомолов Дмитрий Александрович, магистрант.

E-mail: Dimones951@yandex.ru

Морозов Антон Евгеньевич, магистрант.

E-mail: tony-morozov@mail.ru

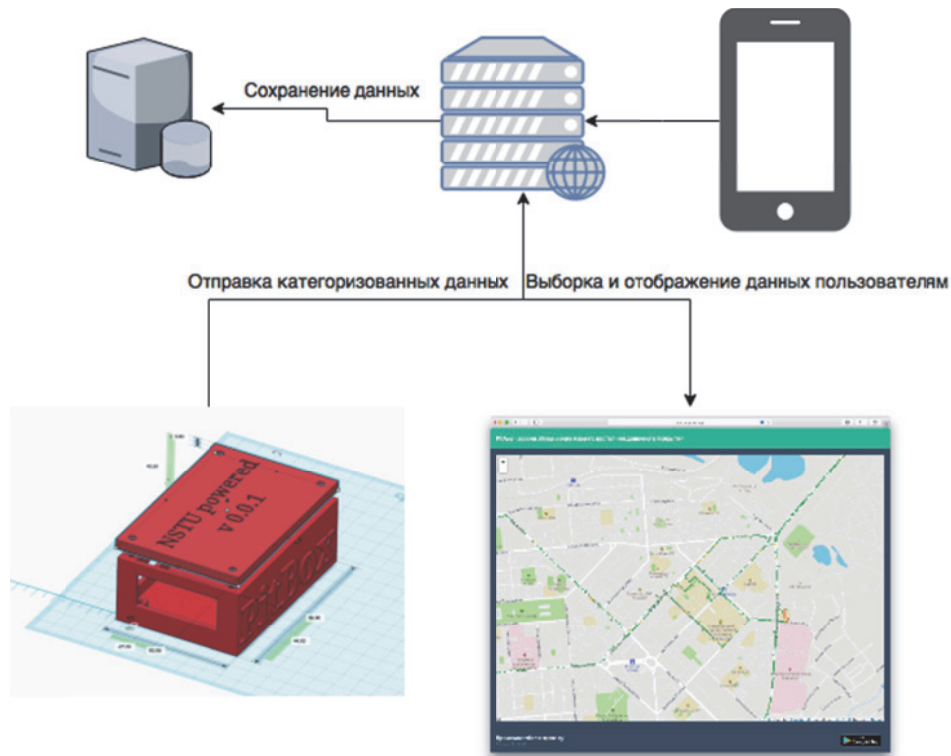


Рис. 1. Общая схема работы системы

пользуется в обмене информацией между АПК «PitBox», она является кроссплатформенной, что делает ее использование возможной на любых языках программирования.

Информационная система базируется на связке Apache 2.4+mod_wsgi+Python, где Apache – свободный кроссплатформенный веб-сервер, позволяющий подключать внешние модули, Python 3 – высокоуровневый язык программирования, ориентированный на производительность разработчика и читаемость кода, а также связка этих элементов, mod_wsgi – модуль для Apache позволяющий запускать Python приложения под Apache, осуществляя это безопасно и быстро.

В качестве аппаратной платформы используется микрокомпьютер Raspberry Pi и их близкие аналоги – Orange Pi. Данные микрокомпьютеры базируются на процессорах ARM (от англ.

Advanced RISC Machine – усовершенствованная RISC-машина; иногда – Acorn RISC Machine) и количеством оперативной памяти более 512 Мб, также для нужд пользователя доступно большое количество выводов GPIO (интерфейс ввода/вывода общего значения). В качестве операционной системы используется система, основанная на ядре Linux – Raspbian,

Использование портов GPIO дает возможность использования внешних модулей таких как акселерометр/гироскоп и GPS-модуль, для осуществления поставленных целей. Для работы с сетью передачи данных может использоваться любой USB-модем. Данное решение позволяет сделать АПК «PitBox». Для реализации приложения сбора и категорирования данных используется высокоуровневый язык программирования Python версии 3.

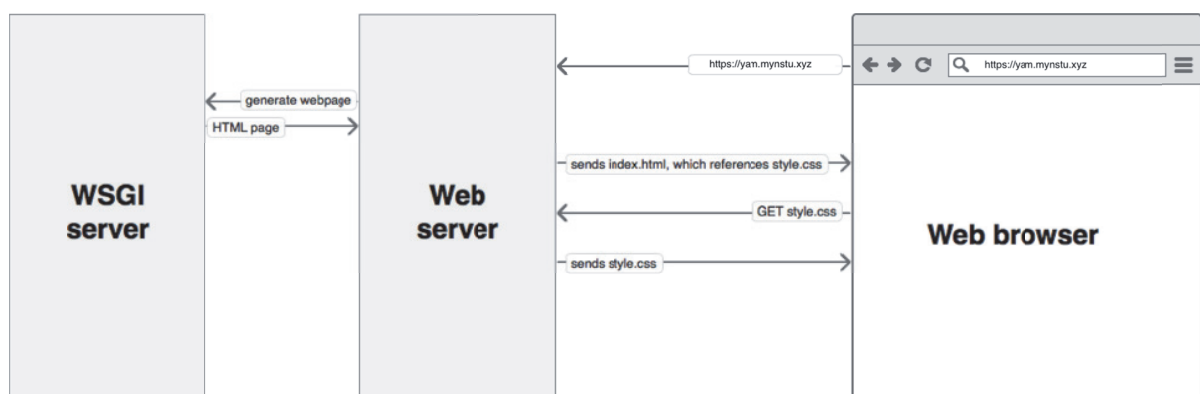


Рис. 2. Архитектура приложений с использованием mod_wsgi

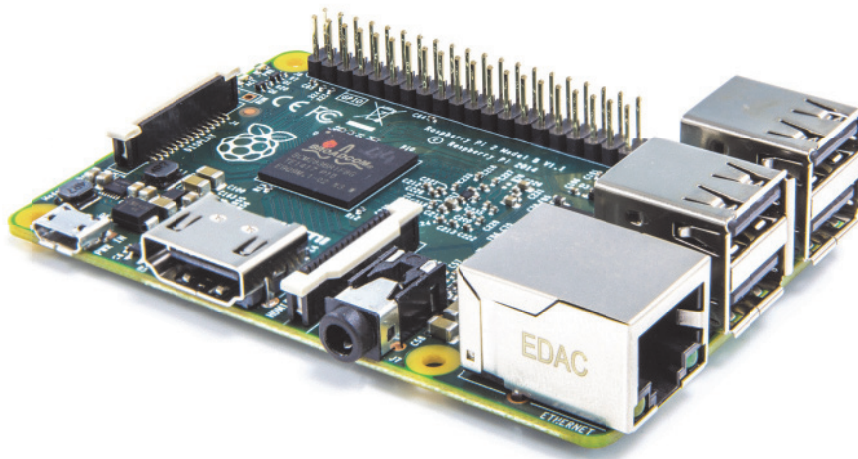


Рис. 3. Микрокомпьютер Raspberry Pi 2 поколения

3. ДАННЫЕ. ХРАНЕНИЕ, ОБМЕН И ОБРАБОТКА

Обмен данными осуществляется в современном формате JSON-структур (JavaScript Objective Notation), который произошел от JavaScript. Выгодно отличается от XML [1], отсутствием избыточности данных, а также поддержкой во всех современных браузерах. Этот формат данных является достаточно популярным во многих Web API, представляет в своей основе одну из двух структур: словарь ключ-значение или упорядоченный массив значений. Рисунок 4 демонстрирует пример JSON-структуры используемой в системе.

Для хранения данных были выбраны две СУБД – MySQL и PostgreSQL, среди которых для определения наиболее производительной СУБД, был проведен тест на реальных данных. В информационной системе мониторинга используются данные дорог в формате GeoJSON (открытый формат, предназначенный для хранения географических структур данных, основан на JSON) [2] полученные из системы OpenStreetMap (сокращённо OSM – некоммерческий веб-картографический проект по созданию силами сообщества участников – пользователей Интернета подробной свободной и бесплатной географической карты мира) [3], в формате разбитых на сегменты по 20 или менее метров. Каждый из сегментов дороги связан логически с объектом дороги, который имеет принадлежность улице,

принадлежащая району города, данная структура представлена на рис. 5.

В таблице, хранящей данные о сегментах дорог, хранится 3612588 записей с геометрическими типами данных Новосибирской области, начала/середины/конца сегмента и относящейся к нему идентификатор дороги, ее содержимое для первых записей отображено на рис. 6. Как можно видеть из рис. 6, данные начала/середины/конца сегмента являются бинарными геометрическими типами данных, а точнее типом *POINT*. Для тестирования производительности двух СУБД был выбран сценарий вставки большого количества данных и проход в дальнейшем по этим данным. Для вставки потребуется создать копию таблиц без индексов, а также предусмотреть копию таблицы с использованием примитивных данных типа *int/double/float*, для сравнения производительности вставки.

После подготовленного массива данных в формате JSON занимающий 740 Мб в файловой системе APFS (файловая система, разработанная Apple Inc. для замены, ранее использовавшейся HFS+), необходимо вставить их СУБД. В табл. 1 приведено время, понадобившееся на вставку 3612588 записей в СУБД и конечный объем этих таблиц в байтах в различных СУБД. Все измерения проводились с одного компьютера до одного сервера с установленными обеими СУБД, поочередно, скорость сетевого соединения составляет на момент тестирования 100 Мбит/с.

```
[{"id":1,"name":"Кировский район"}, {"id":2,"name":
:"Первомайский район"}, {"id":3,"name":"Ленинский район"}
,{"id":4,"name":"Советский район"}, {"id":5,"name"
:"Заельцовский район"}, {"id":6,"name":"Центральный район"}
,{"id":7,"name":"Октябрьский район"}, {"id":8,"name"
:"Дзержинский район"}, {"id":9,"name":"Железнодорожный
район"}]
```

Рис. 4. JSON-структура, хранящая районы города

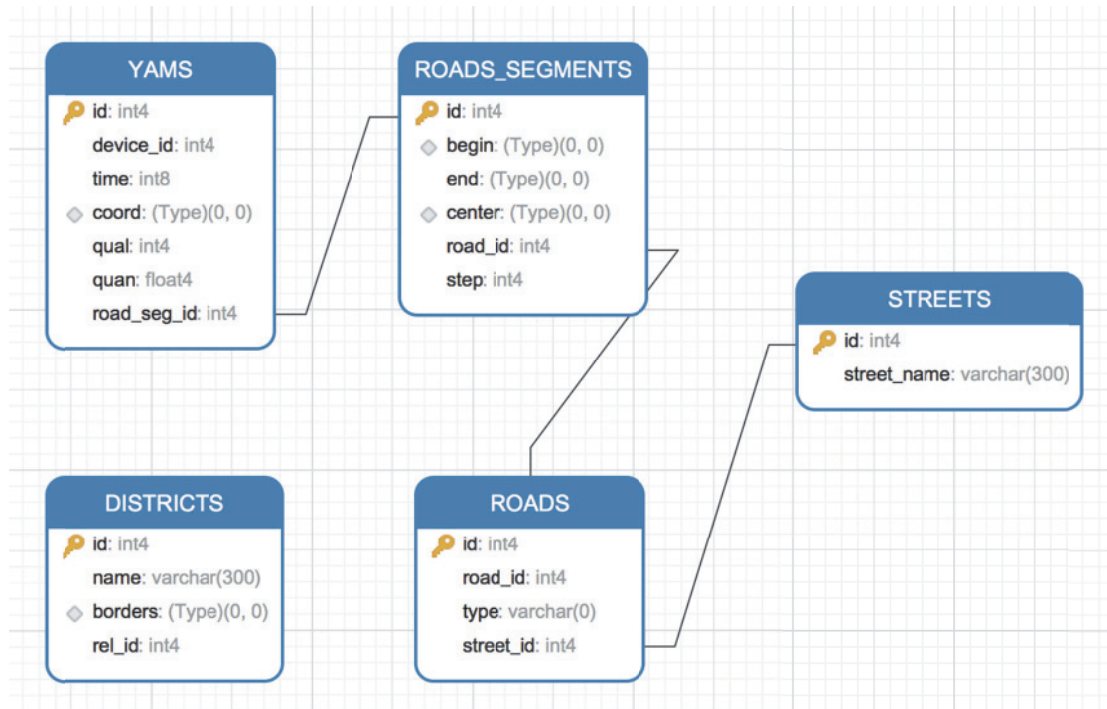


Рис. 5. Структура данных, хранимых в системе

id	begin	end	center	road_id	step
1004256	010100000062D519038B5B53405580441BE5A34B40	0101000000AA1663CDB85B5340106A6F82E4A34B40	010100000006763EE8395B534032F5D9E6E4A34B40	37636	1328
1004257	0101000000AA1663CDB85B5340106A6F82E4A34B40	0101000000F257AC97B65B5340CA539A49E4A34B40	01010000004EB787B2375B5340EDDE047EE4A34B40	37636	1329
1004258	0101000000F257AC97B65B5340CA539A49E4A34B40	01010000003A99F561B45B5340843DC5E0E3A34B40	0101000000096F8D07CB55B5340A7C82F15E4A34B40	37636	1330
1004259	01010000003A99F561B45B5340843DC5E0E3A34B40	010100000082DA3E2C825B53403E27F077E3A34B40	0101000000DE391A47335B534081B25ACE3A34B40	37636	1331
1004260	010100000082DA3E2C825B53403E27F077E3A34B40	0101000000C81B88F6AF5B5340F8101B0FE3A34B40	0101000000267B6311815B53401B9C8543E3A34B40	37636	1332
1004261	0101000000C81B88F6AF5B5340F8101B0FE3A34B40	0101000000135DD1C0AD5B5340B3FA45A6E2A34B40	01010000006FBCACDBAE5B5340D685B0DAE2A34B40	37636	1333
1004262	0101000000135DD1C0AD5B5340B3FA45A6E2A34B40	01010000005B9E1A88AB5B53406DE4703DE2A34B40	010100000087FDF5A5AC5B5340906FDB71E2A34B40	37636	1334
1004263	01010000005B9E1A88AB5B53406DE4703DE2A34B40	0101000000A3DF6355A95B534027CE9BD4E1A34B40	0101000000FF3E3F70AA5B5340A4590609E2A34B40	37636	1335
1004264	0101000000A3DF6355A95B534027CE9BD4E1A34B40	01010000006F960E20A75B5340044E546DE1A34B40	0101000000093BB93AA85B5340160E0F8A0E1A34B40	37636	1337
1004265	01010000006F960E20A75B5340044E546DE1A34B40	01010000003B4DB9EAA45B5340E2CDDC06E1A34B40	0101000000D5F16305A65B5340F38DB039E1A34B40	37636	1338
1004266	01010000003B4DB9EAA45B5340E2CDDC06E1A34B40	010100000006046B5A25B5340B4DC59EE0A34B40	0101000000A0A80EDCA35B5340D00D69D2E0A34B40	37636	1339
1004267	010100000006046B5A25B5340B4DC59EE0A34B40	0101000000D2BA0E80A05B53409CCD7D37E0A34B40	01010000006C5FB99AA15B5340AE8D216BE0A34B40	37636	1340

Рис. 6. Данные таблицы сегментов дорог

Таблица 1. Эксперимент со вставкой большого количества данных

СУБД	Тип данных	Время исполнения, сек	Объем таблицы, Мб	Разница по времени, %	Разница в размере, %
MySQL	Обычный	3.58	294.84	0%	0%
MySQL	Геометрический	5.11	396.94	0%	0%
PostgreSQL	Обычный	6.42	578	79.4%	96%
PostgreSQL	Геометрический	7.24	463	41.58%	16.64%

Данные вставляются множественной командой INSERT по 5000 строк в каждой. Соотношение разниц по времени в таблице производится соответственно типам данных над которыми проведен эксперимент.

Исходя из полученных экспериментальным путем данных, следует отметить что занимаемый размер геометрических типов данных,

больше, что дает основание полагать о преимуществе в данном виде операции у СУБД MySQL, но использование геометрических типов данных дает возможность использовать различные встроенные функции, которые облегчают разработку запросов. Также нужно проверить операцию выборки данных, из сформированных таблиц. На рис. 7 представлен SQL запрос,

```

SELECT
  res.road_seg_id,
  res.qual,
  res.measure_count,
  ST_X(rs.BEGIN) begin_lng,
  ST_Y(rs.BEGIN) begin_lat,
  ST_X(rs.END) end_lng,
  ST_Y(rs.END) end_lat,
  (SELECT d."id" FROM "DISTRICTS" d WHERE st_contains(d.borders, rs.BEGIN) LIMIT 1) district_id
FROM
  (
    SELECT
      y.road_seg_id,
      SUM(y.qual) / COUNT(y.qual) AS qual,
      COUNT(y.qual) as measure_count
    FROM
      "YAMS" y
    WHERE
      y.road_seg_id > 0
    GROUP BY
      y.road_seg_id
  ) res,
  "ROADS_SEGMENTS" rs
WHERE
  rs."id" = res.road_seg_id
    
```

Рис. 7. SQL запрос на выбор данных по неровностям по всем сегментам

Таблица 2. Эксперимент с выборкой данных по таблицам

СУБД	Время исполнения, сек	Разница по времени, %
MySQL	33.309	0%
PostgreSQL	0.504	-6508.9%

в котором происходит агрегирование к сегменту дороги, хранимых значений, которые после формируют оценку состояния дорожного покрытия методов среднего арифметического. В табл. 2 приведены результаты тестирования выборки данных неровностей со всех сегментов дорог. Проход по сегментам дорог был с использованием геометрических типов данных.

Проведя подробный анализ получившихся данных, можно с точностью утверждать что использование СУБД PostgreSQL выгоднее для построения приложений с выборкой большого количества данных за короткий промежуток времени, что сделало возможным информационную системы мониторинга, возможным в реальном времени.

4. РАБОТА АЛГОРИТМА ОПРЕДЕЛЕНИЯ НЕРОВНОСТЕЙ

Для получения информации о дорожном покрытии используются акселерометр и гироскоп.

Акселерометр – прибор, измеряющий проекцию кажущегося ускорения. Под кажущимся ускорением подразумевается разность между истинным ускорением объекта и гравитационным ускорением. Как правило, акселерометр представляет собой чувствительную массу, закрепленную на подвесе. Отклонение массы от её первоначального положения при наличии кажущегося ускорения несет информацию о ве-

личине этого ускорения.

Гироскоп – прибор, способный реагировать на изменение углов ориентации тела в пространстве относительно инерциальной системы отсчета.

Оба датчика реализованы в виде MEMS-микросхем в плате MPU-6050, используемой для получения данных. Информация с них поступает на контроллер, в котором производится классификация полученного сигнала. На рис. 8 показано расположение осей датчика MPU-6050 при его установке внутри автомобиля.

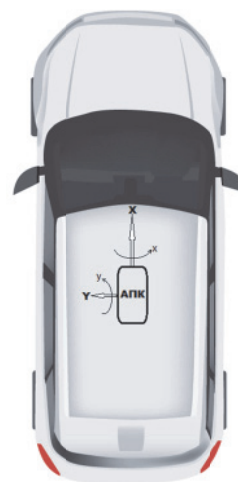


Рис. 8. Расположение осей датчиков АПК в автомобиле

Алгоритм классификации выглядит следующим образом:

- Данные поступают с датчиков на контроллер
- Происходит преобразование данных с помощью фильтра Калмана для получения углов наклона автомобиля в текущий момент времени
- На основании полученной информации за определенный период времени производится классификация неровности на дороге

Фильтр Калмана представляет собой эффективный рекурсивный фильтр, оценивающий вектор состояния динамической системы, используя ряд неполных и зашумленных измерений. В данном случае подобный фильтр помогает получить углы наклона устройства, зафиксированного в автомобиле, используя данные как с акселерометра, так и с гироскопа.

Для классификации сигнала планируется использовать рекуррентные нейронные сети, хорошо зарекомендовавшие себя в работе с временными рядами. В качестве возможных архитектур НС предлагаются LSTM (Long short-term memory) [7] и GRU (Gated recurrent unit). Обучение планируется производить на собранных и размеченных данных за определенный период времени. На рисунке 9 показаны собранные данные об углах наклона. После обучения НС будет использоваться для определения ускорения и замедления автомобиля, поворотов и неровностей на дороге.

5. ПЛАНЫ РАЗВИТИЯ

На данный момент идет обновление и улучшение архитектуры системы и тестирование платформы, а также доработка алгоритма идентификации неровностей на дорогах. Планируется использование технологий контейнеризации для улучшения масштабируемости информационной системы, в случае увеличения создаваемого трафика. А также необходим поиск новых эффективных способов устранения шумов на полученных данных, как и улучшение работы классификатора за счет усовершенствования его структуры.

В дальнейшем необходимо перестроить хранение статических данных о дорогах и добавить разбиение по регионам, что даст возможность разделять данные по кластерам в разные таблицы, в результате которой, повысится удобство пользования системой, разработки ее и не снизится скорость исполнения запросов к ней.

ЗАКЛЮЧЕНИЕ

Разработан прототип платформы для мониторинга состояния дорожного покрытия по Новосибирской области. Проведены начальные этапы тестирования, которые показали работоспособность системы. Обозначены цели и планы по развитию платформы.

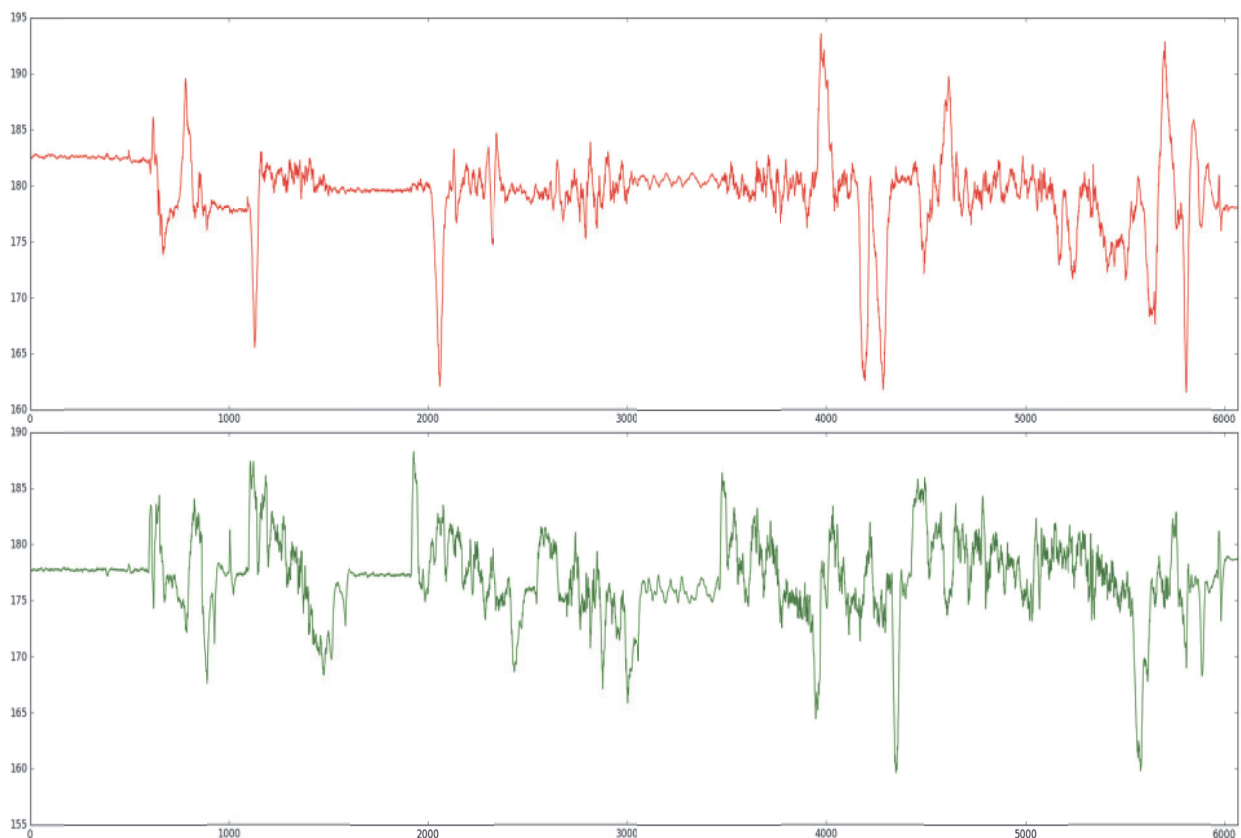


Рис. 9. Графики данных об углах наклона: красный (вверху) – наклон по оси X; зеленый (внизу) – наклон по оси Y

СПИСОК ЛИТЕРАТУРЫ

1. Google Trends JSON, XML [Electronic resource] // Google, 2017. URL: <https://trends.google.ru/trends/explore?q=JSON,XML> (дата обращения 11.10.2017)
2. GeoJSON [Electronic resource] // Wikipedia, 2017. URL: <https://ru.wikipedia.org/wiki/GeoJSON> (дата обращения 11.10.2017)
3. OpenStreetMap [Electronic resource] // Wikipedia, 2017. URL: <https://ru.wikipedia.org/wiki/OpenStreetMap> (дата обращения 11.10.2017)
4. Официальный сайт города Новосибирска [Electronic resource] // Новосибирск, 2017. URL: http://novo-sibirsk.ru/major/news_1153/71024/ (дата обращения 11.10.2017)
5. Roy T. Fielding, Richard N. Taylor, Justin Erenkrantz, Michael M. Gorlick, E. James Whitehead, Rohit Khare, Peyman Oreizy "Reflections on the REST Architectural Style and "Principled Design of the Modern Web Architecture" – Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017), pp. 4-11, 2017.
6. PostGIS - Documentation [Electronic resource] // PostGIS, 2017. URL: <http://postgis.net/documentation/> (дата обращения 11.10.2017)
7. Hasim Sak, Andrew Senior, Françoise Beaufays "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling" – Google USA, 2012

DEVELOPMENT OF AN INFORMATION SYSTEM FOR MONITORING ROAD CONDITIONS

© 2017 A.A. Yakimenko^{1,2}, A.A. Malyavko¹, D.A. Bogomolov¹, A.E. Morozov¹

¹ Novosibirsk State Technical University

² Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk

In the article, the process of developing a tool for monitoring the condition of the road surface, in particular, solving problems of storage and processing of data. The article confirms the relevance of the development. A prototype of an experimental system working on the basis of this platform was developed. The work was supported by the RFBR grant 16-37-00240 Mol_a.

Keywords: monitoring, load, PostgreSQL, PostGIS, platform, web service, REST, API.

Aleksander Yakimenko, Candidate of Technics, Associate Professor at the Computing Machinery.

E-mail: yakimenko@corp.nstu.ru

Aleksander Malyavko, Candidate of Technics, Associate Professor at the Computing Machinery Department.

E-mail: a.malyavko@corp.nstu.ru

Dmitriy Bogomolov, Master Student.

E-mail: Dimones951@yandex.ru

Anton Morozov, Master Student.

E-mail: tony-morozov@mail.ru