

Контактная информация

Клюнин Алексей Олегович

aoklyunin@gmail.com

+7(921) 438-98-10

<https://github.com/aok-buran>

<https://habr.com/ru/users/aok-buran/>

Стек технологий:

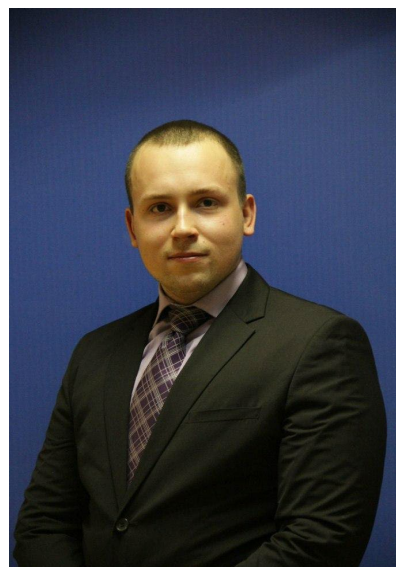
Java, C/C++, MATLAB, ROS, Android SDK, MySQL, Linux, Git, Python, Django, Docusaurus, Docker, Jenkins

Образование:

- В 2012 году закончил ФМЛ № 239 в Санкт-Петербурге,
- 2012-2016 - бакалавриат кафедры СУИИ университета ИТМО, тема диплома: «Силомоментное очувствление промышленного робота-манипулятора».
- 2016-2018 - магистратура кафедры СУИИ по направлению «15.04.68.06 Интеллектуальные технологии в робототехнике», диплом с отличием

Опыт работы

- 2013 – 2017 - инженер КИП и АСУ, ООО «НПО Поиск». Занимался проектированием, сборкой и наладкой автоматики. Также мною спроектирован и введен в эксплуатацию [станок по намотке баллонов](#). Суть работы станка в том, что необходимо синхронизировать движение рабочего инструмента с вращением баллона так, чтобы на баллоне получался заданный циклический рисунок. Это повышает надёжность и придает товарный вид изделию. Изделия задействованы в производственных цепочках ОПК РФ.
- 2015 – 2019 год инженер кафедры СУИИ университета ИТМО, сотрудник [Международного научного центра "Нелинейные и адаптивные системы управления"](#).
- 2018 – 2019 года инженер в компании TRA Robotics. Разработка программного обеспечения для управления робототехническими производственными линиями.
- 2019 – 2020 инженер-программист, ООО «НПО Поиск». Разработка аппаратно-программных комплексов неразрушающего контроля композитных изделий, модернизация и проектирование станков и измерительных систем.
- 2020 – 2023 проектирование и разработка фреймворка планирования движения для системы роботов-манипуляторов. [Обзорная статья на хабре](#). Данный фреймворк – моя личная разработка. С документацией фреймворка можно ознакомиться [здесь](#). Он написан на C++, собирается в deb-пакет с помощью Jenkins. Используемые технологии: C++, QT, OpenGL, Boost, Eigen
- С 2023 по настоящее время разработка с помощью OpenSource средства численного моделирования [Sparselizard](#) BLDC привода облегчённой конструкции (личный проект). Данная разработка мотивирована тем, что обычные роботы-манипуляторы с весом в несколько сотен килограмм могут поднимать только единицы. Основная причина такой низкой эффективности кроется именно в весе приводов.



Публикации

- [Быстрый поиск изоморфных подграфов \(habr.com\)](#)
- [Начало шестого технологического уклада \(habr.com\)](#)
- [Монотонная кубическая интерполяция \(habr.com\)](#)
- Интеграция робототехнических единиц в единую систему 2017 Сборник тезисов участников форума "Наука будущего-наука молодых" с. 309-310 А.О. Ключин, И.В. Петраневский
- Проблемы интеграции и взаимодействия роботов в производственную цепочку механической обработки деталей сложной геометрической формы 2017/2/21 А. О. Ключин, И. В. Петраневский, С. А. Колюбин, О. И. Борисов, В. С. Громов
- Идентификация сухого и вязкого трений в сочленениях робота-манипулятора И.В. Петраневский, А. О. Ключин, А.А. Пыркин - Навигация и управление движением, 2018
- Case study on human-free water heaters production for industry 4.0 Oleg I Borisov, Vladislav S Gromov, Sergey A Kolyubin, Anton A Pyrkin, Nikolay Y Dema, Vladimir I Salikhov, Igor V Petranevsky, Alexey O Klyunin, Sergey V Shavetov, Alexey A Bobtsov, 2018 IEEE Industrial Cyber-Physical Systems (ICPS) p. 369-374
- Силомоментное очувствление промышленного робота-манипулятора в задаче полировки 2017/2/21 И. В. Петраневский, А. О. Ключин, С. А. Колюбин, О. И. Борисов, В. С. Громов
- Идентификация статической модели робота-манипулятора А.О. Ключин, И.В. Петраневский, А.А. Пыркин - Навигация и управление движением, 2018
- Arc approximation algorithm of spatial movements for industrial robots Oleg I Borisov, Vladislav S Gromov, Anton A Pyrkin, Igor V Petranevsky, Alexey O Klyunin, Alexey A Bobtsov IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society 3429-3434
- Output robust control with anti-windup compensation for robotic boat Oleg I Borisov, Vladislav S Gromov, Anton A Pyrkin, Alexey A Bobtsov, Igor V Petranevsky, Alexey O Klyunin 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR) p. 13-18
- Human-free robotic automation of industrial operations Oleg I Borisov, Vladislav S Gromov, Sergey A Kolyubin, Anton A Pyrkin, Alexey A Bobtsov, Vladimir I Salikhov, Alexey O Klyunin, Igor V Petranevsky IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society p.6867-6872

Патенты

- Программа для управления робототехнической системой посредством силомоментного датчика RU 2016619315
- Gb PCT/RU2018/000737 SENSOR-FREE FORCE/TORQUE SENSING IN AN ARTICULATED ELECTROMECHANICAL ACTUATOR-DRIVEN ROBOT
- EP3880412 (A1) - SENSOR-FREE FORCE/TORQUE SENSING IN AN ARTICULATED ELECTROMECHANICAL ACTUATOR-DRIVEN ROBOT
- US2022009097 SENSOR-FREE FORCE/TORQUE SENSING IN AN ARTICULATED ELECTROMECHANICAL ACTUATOR-DRIVEN ROBOT

Участие в конференциях и конкурсах

- XIX конференция молодых ученых «Навигация и управление движением» 2017 год
- Награды за лучший научно-исследовательский доклад Всероссийский конгресс молодых учёных в 2014 и 2016 годах
- Форум «Наука молодых, наука будущего» в 2016 и 2017 годах, второе место в конкурсе научных докладов в 2017 году.

О себе

- Занимал призовые места на региональных олимпиадах по математике, физике, химии и информатике
- Интересуюсь философией и нейрофизиологией

Портфолио

1. НПО Поиск

Мною реализован ряд проектов по автоматизации для [НПО Поиск](#). Эта организация занимается опытным и мелкосерийным производством композитных баллонов и агрегатов высокого давления. Самые значительные из моих проектов описаны ниже.

1.1. Намотка баллонов



Чтобы снизить вес баллонов, поверх металлической оправки наматывается нить из стеклотекстолита, пропитанная эпоксидной смолой. За счёт такого упрочнения удаётся снизить толщину металлической оправки, а значит, и вес баллона в целом при равных предельных давлениях.

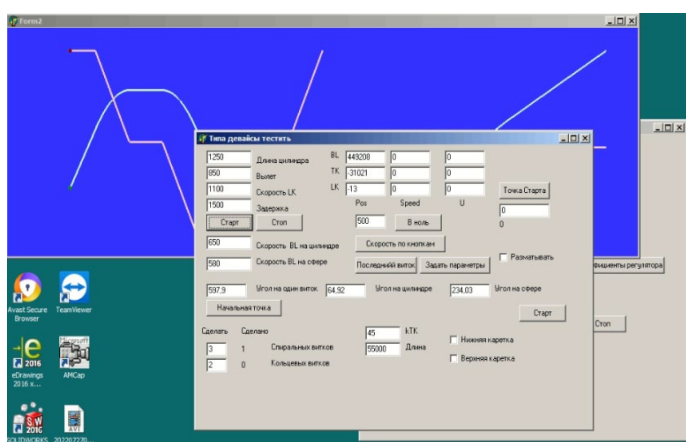
Видео работы намоточного станка после моей автоматизации можно посмотреть [здесь](#). Процесс намотки осуществляется следующим образом: на вращающийся баллон наматываются две нити, пропущенные, через подвижные каретки.

Эти каретки движутся по определённому профилю скорости, смещая тем самым точку укладки нити. При помощи различных режимов движения можно добиться различных рисунков намотки.

Изначально станок работал по концевым датчикам. Когда металлические пластины, закреплённые на каретке, пересекали тот или иной датчик, реле меняло направление скорости каретки на противоположную по направлению. Модуль скорости задавался с помощью потенциометра.

Программное автоматическое управление позволило синхронизировать движение кареток и баллона таким образом, чтобы каждая из координат станка отрабатывала заданную функцию положения от времени с приемлемой для технического процесса ошибкой. Подробнее о первой версии системы можно прочитать [здесь](#).

Программное обеспечение было написано на Delphi, т. к. не было серьёзных требований к скорости работы ПО, а эта среда позволяет очень быстро и удобно добавлять формы, к тому же в ней были встроенные инструменты для работы с COM-портом и сокетами. Приложение состояло из двух частей: сервер и клиент. Сервер отрабатывал задания от клиента по положению и скорости в текущий момент времени, а клиент формировал эти задания в цикле по той или иной параметризованной траектории (некоторый аналог G-кодов).



Во второй версии программы управление приводами было переведено на цифровые каналы управления. Два слабomощных частотных регулятора управлялись с помощью протокола Modbus (RS-232), а третий, движущий нижнюю быстроходную каретку, - с помощью внутреннего недокументированного протокола SEW. Т. к. программное обеспечение поставщика работало поверх RS-232, то с помощью UART-сниффера я распознал структуру пакетов для простых команд движения влево и вправо с заданной скоростью, а потом подобрал алгоритм контрольной суммы.

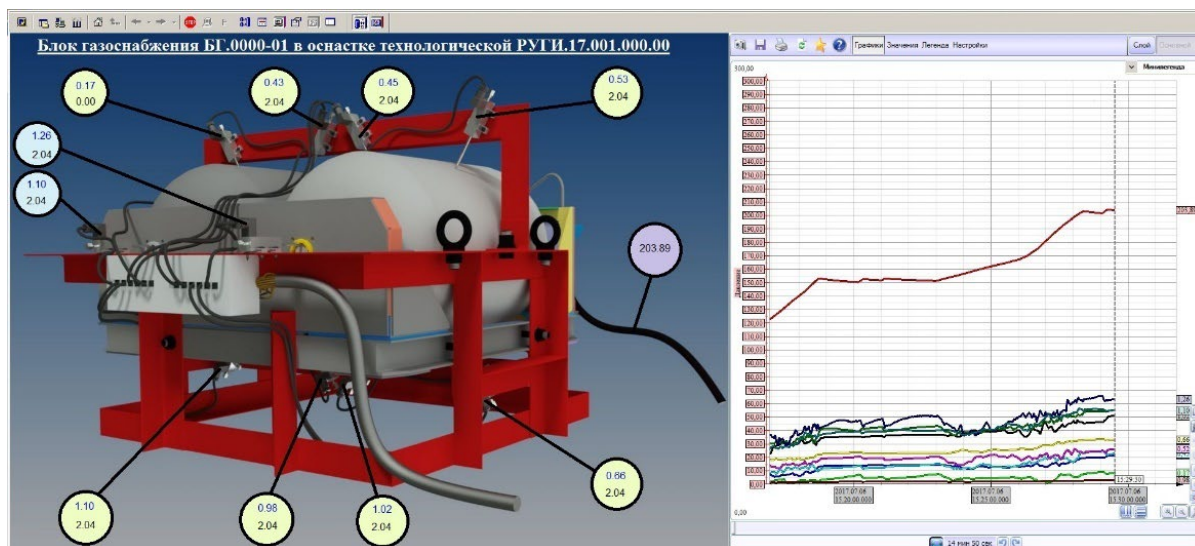
1.2. Измерение деформаций

У каждого баллона есть предельно допустимые деформации, зависящие от нагрузки.

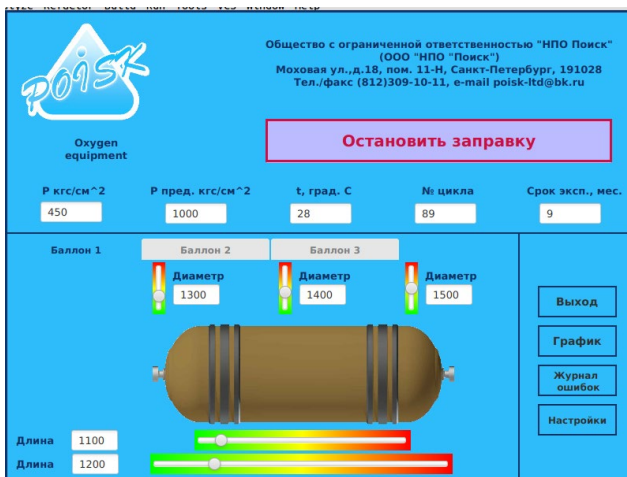
Сравнение реальных и теоретических смещений точек измерения на поверхности позволяет установить, пригоден баллон для дальнейшей эксплуатации или нет. Можно оснастить установку датчиками перемещения и с их помощью следить за состоянием баллона без демонтажа и транспортировки.

Для измерения перемещений поверхности баллонов мною были использованы резистивные датчики перемещения и промышленный контроллер. Также этот контроллер обрабатывал показания датчика давления. Из-за работы с высоким давлением испытательная зона находилась далеко от рабочего места оператора.

Испытания проводятся посредством нагнетания высокого давления в баллонах. В качестве рабочего вещества используется жидкость из-за её низкой сжимаемости. Такой режим работы не поддерживается обычными промышленными контроллерами. В случае разрыва баллона контроллер может быть разрушен. Поэтому было решено вывести контроллер из испытательной зоны, а все датчики подключить с помощью ethernet-кабеля и patch-панелей.



Такое подключение добавляет паразитное сопротивление в резистивных датчиках. Чтобы его скомпенсировать, была проведена серия экспериментов с эталонными расстояниями в форме пластин для каждого датчика. После этого с помощью аппроксимации полиномом пятой степени в Matlab была установлена зависимость между измеренным значением на датчике и реальным. Компьютерная система диагностики была построена на основе MasterScada.



Далее была разработана система диагностики более габаритных баллонов. Проект руководства по эксплуатации (собран с помощью tex) можно скачать [здесь](#).

Также была разработана экспериментальная программа для связи с контроллером напрямую по COM-порту (протокол Modbus). В ней производился общий расчёт изменения геометрии баллонов. Это важно, т. к. во время нагрузки баллон может перемещаться и изгибаться.

1.3. Система газификации

Инженеры НПО "Поиск" разработали криогенную систему обеспечения газом КСОГ-1. Она предназначалась для перевода азота и метана из жидкой фазы в газообразную исключительно за счёт тепла, получаемого из окружающей среды. Электроэнергия была необходима только для питания датчиков.

Для исследования этой системы был проведён ряд экспериментов с газификацией безопасного азота, а после - метана. В ходе экспериментов измерялись температура в разных точках системы, давление внутри ёмкости, а также изменения веса при стравливании.

Эти данные фиксировались с помощью системы датчиков, промышленного контроллера и MasterScada. Для записи изменений веса отдельно была написана программа, работающая с цифровыми весами по COM-порту. После эксперимента показания веса синхронизировались с показаниями остальных датчиков по характерным точкам.



По результатам экспериментов был составлен pdf-отчёт, который можно скачать [здесь](#). Также я смонтировал отчётное видео, с которым можно ознакомиться [здесь](#).

Для создания расширяющихся графиков из видео мной была написана отдельная программа. Исходные данные экспериментов сохранялись в формате csv. На их основе я порождал последовательность кадров, каждый из которых отображал всё большую и большую часть измерений.

2. Робототехника

Около пяти лет я работал младшим научным сотрудником на факультете СУиР университета ИТМО. Мною с коллегами по кафедре опубликован ряд статей как в российских, так и в зарубежных научных журналах. Их списки приведены на последних страницах портфолио.

В этом разделе будут описаны мои основные научные наработки. В университете я в основном работал с промышленным роботом Kawasaki fs06n и мобильным роботом Kuka Youbot. На основе придуманного мною алгоритма было получено несколько патентов, в том числе зарубежных.

2.1. Силомоментное очувствление

Несмотря на то, что звенья промышленных роботов позиционируются достаточно точно относительно друг друга, при обработке заготовок очень сложно достичь точного расположения рабочего инструмента относительно них. Помимо этого, часть технологических процессов требует воздействия с заданной силой или моментом.

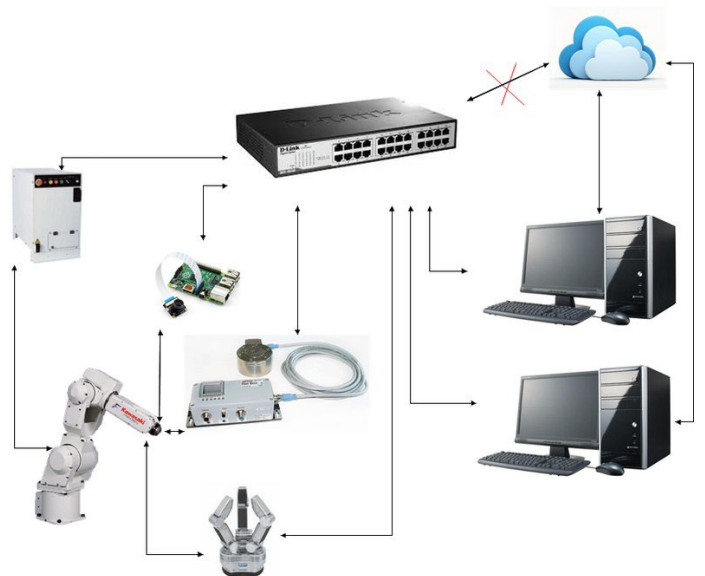
В таких случаях на энд-эффектор робота устанавливается силомоментный датчик, в моём случае - ATI F/T IP60 Delta, а уже на него рабочий инструмент. Тогда, зная теоретические и реальные силу и момент, создаваемые рабочим инструментом, можно получить избыточные значения. Эти избыточные значения будут вызваны реакцией заготовки на воздействие инструментом.

Для демонстрации силомоментного очувствления я разработал программу, которая по сети считывала показания силомоментного датчика и определяла теоретическое воздействие рабочего инструмента. После этого она пересчитывала избыточные показания в абсолютную систему координат и формировала задание, куда сместить рабочий инструмент.

Получилось два режима работы. В первом можно было давить рукой на рабочий инструмент робота, и он смещался в соответствующую сторону. Такой режим называется Force Feedback. Демо-видео можно посмотреть [здесь](#).

Во втором режиме можно было поворачивать рабочий инструмент за счёт приложенных внешних моментов. Режим управления по моментам называется Torque Feedback. Демо-видео можно посмотреть [здесь](#). Подробнее об этой работе можно прочитать в моей [бакалаврской диссертации](#).

Чтобы построить такую систему, мне пришлось разобраться с внутренним языком Kawasaki, он довольно плохо документирован. В итоге выяснилось, что самый оптимальный способ управления – это контроль позиции. На контроллере в бесконечном цикле обрабатываются входящие команды по ethernet, пока не будет получена команда завершения работы. При этом пока не будет отработано предыдущее задание, отработка нового не начнётся. Структуру пакета пришлось составлять самостоятельно.



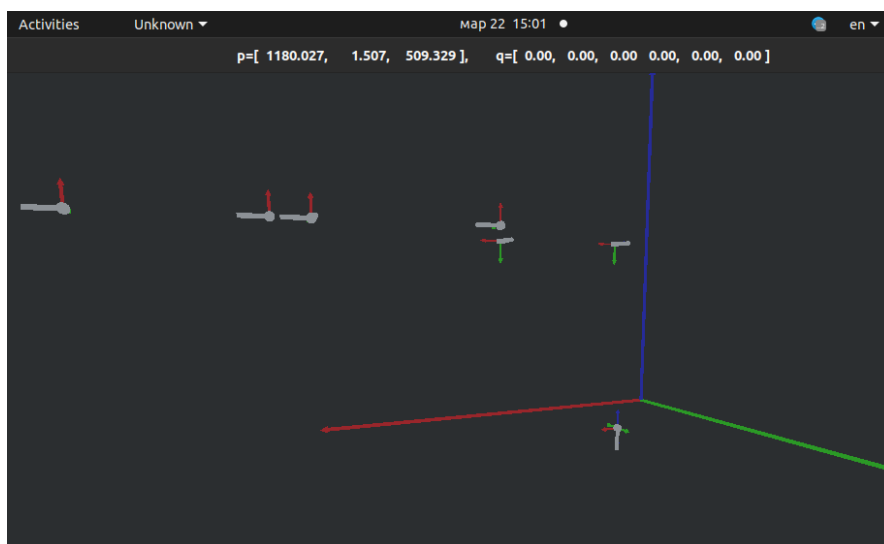
Строго говоря, такой робот вообще не был рассчитан под работу с обратной связью, поэтому быстрая работа системы происходила рывками, а плавный режим получался слишком медленным. Передвигать робота рукой, как на видео, без соответствующего алгоритма нельзя. Он поддерживает заданное положение и компенсирует все внешние воздействия.

Программа управления была написана на Java состояла из двух модулей: клиент и сервер. Сервер обобщал все показания системы и получал задание по перемещению от клиента. В качестве клиента я написал отдельно десктоп-программу для управления с обратной связью, отдельно Android-приложение, которое позволяло перемещать робота в заданное положение и поворачивать звенья в заданную конфигурацию. Также отображались показания силомоментного датчика. Для работы с датчиком АТІ в матлаб была написана библиотека. Демо-видео можно посмотреть [здесь](#).

2.2. Визуализация параметров Денавита-Хартенберга

Чтобы получить координаты рабочего инструмента робота-манипулятора в мировой системе координат, нужно составить матрицы перехода из каждого звена в следующее. Для этого каждое звено параметризуется четырьмя значениями: два отвечают за расстояния между звеньями, два – за повороты. Эти параметры называют параметрами Денавита-Хартенберга или ДХ-параметрами.

Для составления ДХ-параметров реального робота недостаточно составить кинематическую схему, требуется совместить нулевые точки (точки отсчёта углов) в звеньях робота. Точки отсчёта углов в каждом из сочленений робота не стандартизированы. Поэтому ДХ-параметры, составленные по кинематической схеме, могут давать положения рабочего инструмента, качественно отличные от реальных.



Для определения точек отсчёта и уточнения ДХ параметров была написана 3D программа визуализации параметров ДХ (C++ и OpenGL). Исходники можно скачать [здесь](#).

2.3. Обратная задача кинематики для мобильных роботов

У робота есть два способа определения пространства состояний: положение с ориентацией рабочего инструмента и углы поворотов звеньев. Первое называется операционным пространством, второе – конфигурационным.

Переход из конфигурационного пространства однозначен (прямая задача кинематики), а в обратную сторону (обратная задача кинематики) – нет. Из углов поворота звеньев робота мы однозначно можем получить положение и ориентацию рабочего инструмента. Решение обратной задачи достаточно трудоёмко.

Это вызвано сильной нелинейностью связи конфигурационного и операционного пространств. Обычно предлагается решать эту задачу геометрически либо через взятие частных производных.

В своей [магистерской диссертации](#) я предложил новый способ решения обратной задачи кинематики. Обычно ориентация рабочего инструмента робота параметризуется углами Эйлера. Однако они определяются неоднозначно и имеют разрывы, если рассматривать их зависимость от углов поворота звеньев.

Я ввёл новую параметризацию через кватернионы, выраженные параметрами Родриго-Гамильтона. Полученные зависимости оказались непрерывными относительно конфигурационного пространства.

Это позволило составить функцию ошибки между текущим положением рабочего инструмента и желаемым. При этом функция ошибки в конечном счёте являлась скалярной функцией, зависящей только от углов поворота звеньев. Тогда стандартными методами для каждого робота можно составить функцию такого рода, а потом классическими методами минимизировать значение, т. е. свести ошибку решения к нулю.

Для проверки разработанного алгоритма для робота Kuka Youbot было составлено несколько опорных точек, каждая из которых описывала произвольные достижимые положения рабочего инструмента и ориентацию. По каждой из точек с помощью минимизации были получены соответствующие углы поворота звеньев, после чего с помощью Gazebo и Ros был составлен python-скрипт, который последовательно отправлял виртуальному роботу задания по углам поворота звеньев и сравнивал полученные положение и ориентацию с заданными.



2.4. Бездатчиковое управление

Чтобы поддерживать заданную силу воздействия на рабочем инструменте робота, необязательно оснащать его силомоментным датчиком. Можно оценить динамическую модель робота, после чего сверить реальные моменты в сочленениях и теоретические. Этот избыток можно пересчитать в силу и момент на рабочем инструменте. Динамическая модель состоит из трёх компонент: первая зависит от ускорения - инерция, вторая от угла и скорости – различные трения, а третья – только от угла - гравитационная компонента.

Чтобы идентифицировать гравитационную компоненту, робот размещался в различных точках. Т.к. гравитационная компонента - это частная производная потенциальной энергии, то я выразил потенциальную энергию каждого из звеньев. Звенья были идеализированно представлены, как невесомые стержни с прикреплённым точечным весом. При движении звена соответствующий точечный вес движется вместе с ним. За счёт такого допущения я составил явные зависимости потенциальных энергий каждого из звеньев от углов поворота звеньев.

После этого я продифференцировал суммарную потенциальную энергию системы по каждой из обобщённых координат (углов поворота звеньев) и получил зависимость гравитационной компоненты от обобщённых координат.

Робот Kuka Youbot был демонтирован с колёсной базы, после чего был проведён ряд экспериментов с фиксацией манипулятора в произвольных положениях. В каждом из этих положений замерялась сила тока в каждом из сочленений. Она пропорционально связана с моментом в звене. Эксперименты выполнялись с помощью программ, написанных на Python+Ros.

Измеренные зависимости я разложил в виде матричного уравнения и применил метод наименьших квадратов. Сначала часть значений определялась с ошибкой, т. к. у составленной матричной функции число обусловленности приближалось к бесконечности. Это было связано с тем, что часть строк оказалась линейно зависимой (матрица неполного ранга), поэтому пришлось их найти и построить усечённую регрессионную модель.



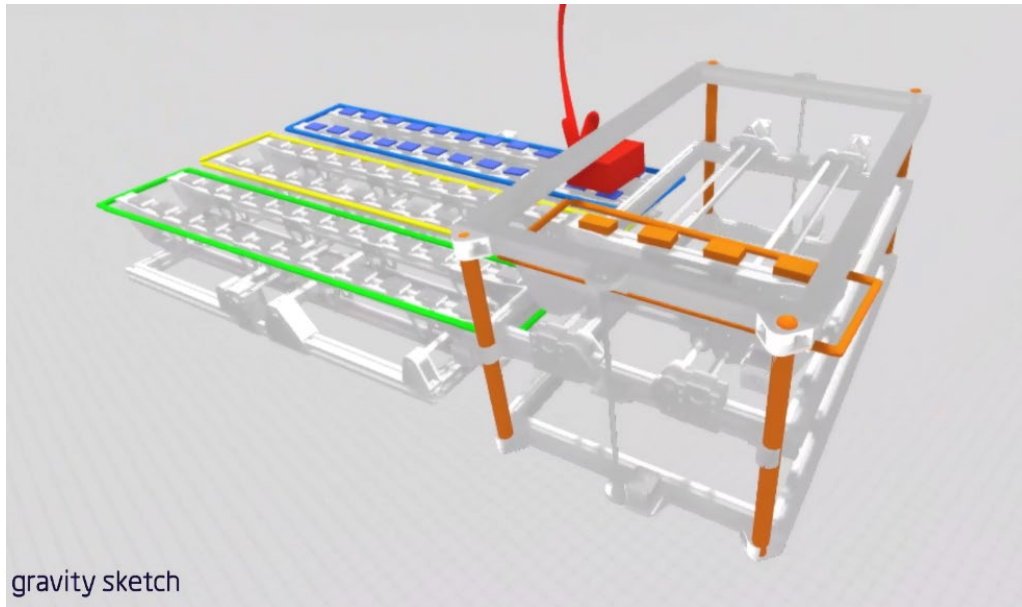
Для идентификации компоненты трения был проведён ряд экспериментов с каждым из звеньев, кроме первого. Суть экспериментов заключалась в симметричном относительно вертикального положения движении из крайнего положения в крайнее. Получившиеся два участка различались только на гравитационную компоненту, причём взятую с разным знаком. Поэтому я с помощью Matlab достроил недостающие симметричные значения с каждой из сторон и потом попарно сложил их. С помощью аппроксимации были установлены оценки зависимости компоненты трения для каждого задействованного в эксперименте звена.

По составленной оценке гравитационной компоненты я разработал аналог программы управления по моментам роботом Kawasaki, но без использования силомоментного датчика. Реальные токи на каждом звене сверялись с оцененными, после чего их разница преобразовывалась в оценку внешнего воздействия. Демо-видео находится [здесь](#).

2.5. TRA Robotics

Данная компания занималась проектированием полностью автоматизированной фабрики. Работа в ней, весь код я писал на C++.

Там я разработал концепт нового типа складов. Он экономил место хранения в замкнутом объёме. Идея такого склада основана на принципе ханойской башни. Система хранения состоит из направляющих механизмов (зелёный, жёлтый и синий) и лифта(оранжевый).



Каждый предмет хранения помещался на специальный поддон с магнитами, а на направляющих были расположены движущие ленты, тоже с магнитами. В месте перехода с направляющих к лифту были помещены специальные отсекатели, которые не позволяли грузу уходить вниз. За счёт них груз заталкивался на лифт во время вращения лент.

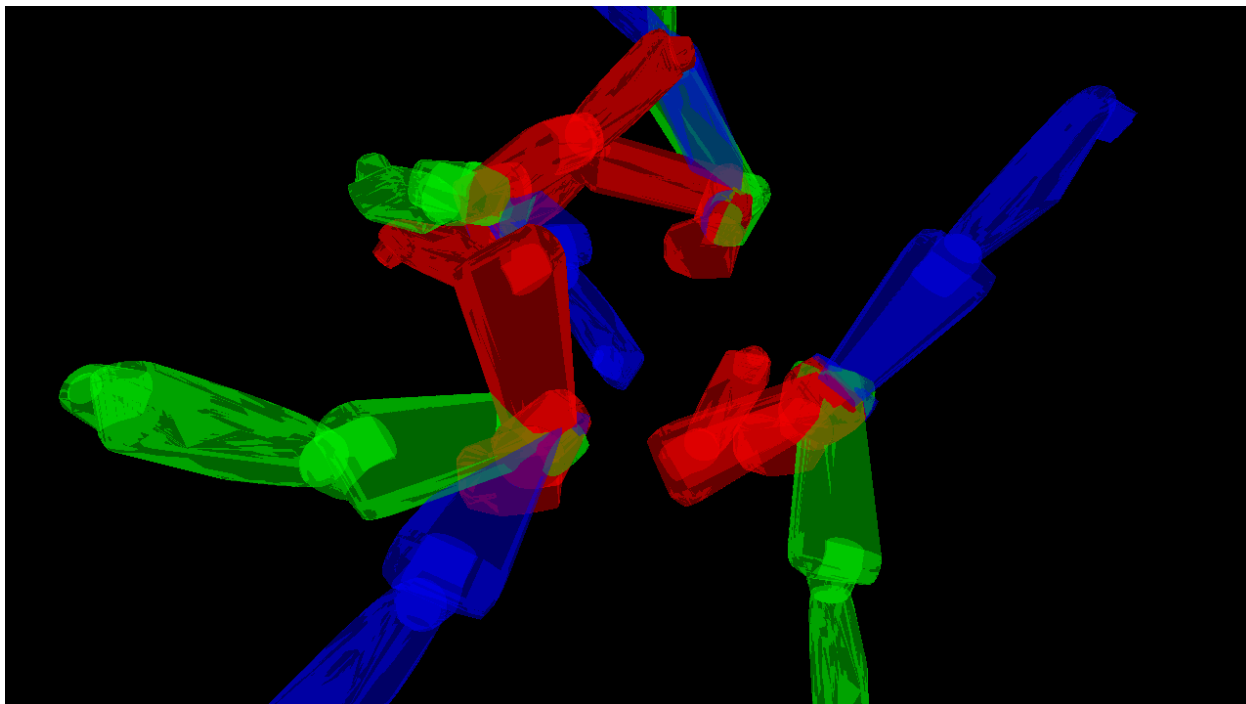
В TRA я разработал класс C++ для снятия показания с силомоментного датчика DynPick и прописал для этого датчика интеграцию в корпоративный BlackBoard. Мой код поднимал топик и раз в заданное время писал в него считанные значения сил и моментов.

Для разработки цифровых двойников робототехнических ячеек я написал модуль UnrealEngine, который по заданным конфигурациям формировал с помощью бисплайнов непрерывные траектории.

Для проекта автономной тележки я разрабатывал кинематическую модель, проводил моделирование системы и тестировал простейшие алгоритмы управления.

3. Buran Motion Planning Framework

Данный фреймворк – моя личная разработка, написан на C++, собирается в deb-пакет с помощью Jenkins. С обзорной статьёй можно ознакомиться [на хабре](#).



На текущий момент подавляющее большинство средств планирования движения работает по одному и тому же принципу: вся сцена описывается как один робот, после чего выполняется планирование на сетке.

У такого подхода есть две основных проблемы:

- 1) планирование на сетке гарантирует допустимость только состояний в её узлах, промежуточные никак не оцениваются и не проверяются.
- 2) для сцены из нескольких роботов размерность пространства планирования получается слишком большой (алгоритмическая сложность планирования растёт как показательная функция).

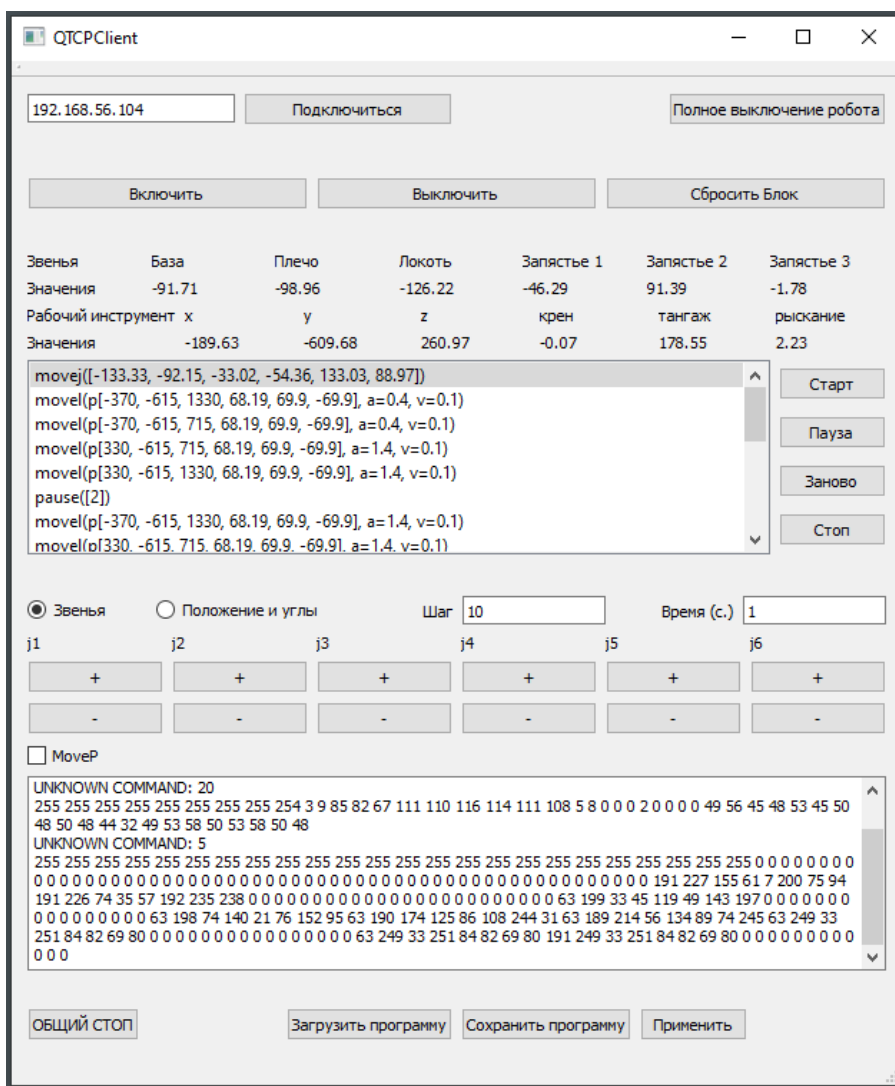
Данный фреймворк решает обе озвученные проблемы. С документацией фреймворка можно ознакомиться [здесь](#).

4. Завод «Контакт»

Данное [предприятие](#) специализируется на металлообработке.

4.1. Робототехника

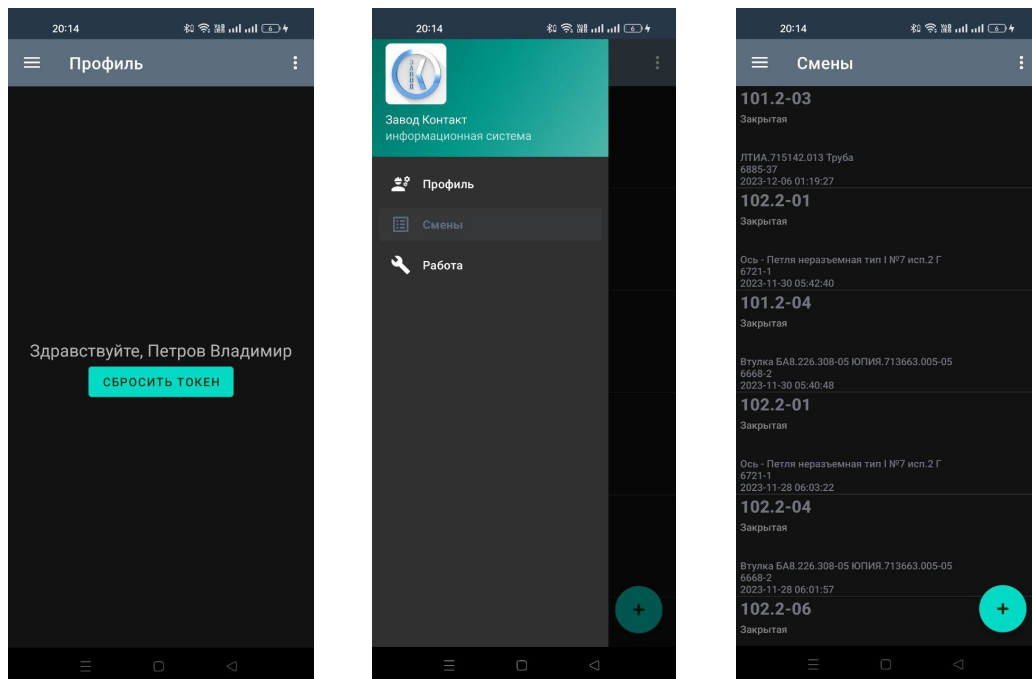
Для управления роботом UR-10 мною написана программа для ОС Windows (QT, C++). С инструкцией Вы можете ознакомиться по [ссылке](#).



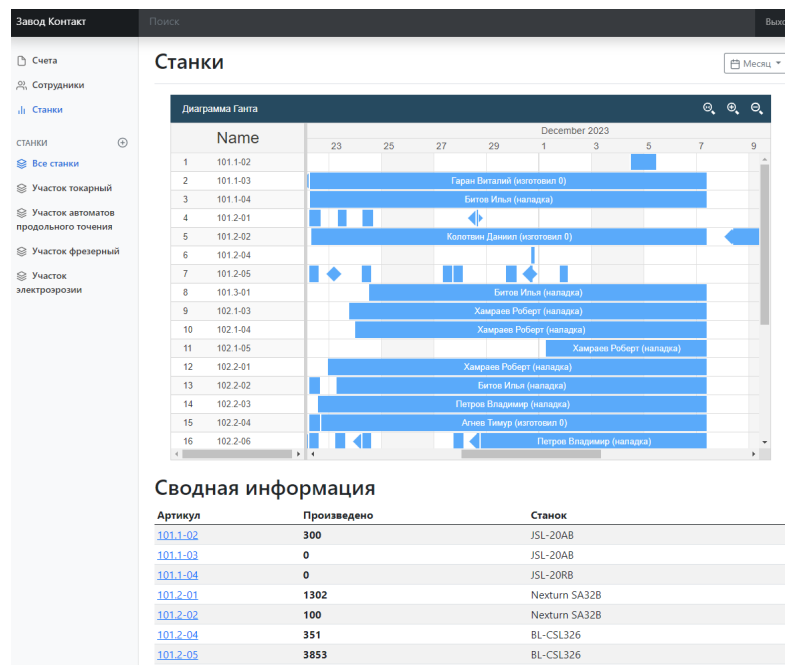
Она позволяет программировать движение робота с помощью аналога G-кодов без использования пульта управления и встроенного языка.

4.2. Диспетчеризация

Для повышения эффективности производства на заводе «Контакт» я разработал систему диспетчеризации токарного цеха. Для операторов и наладчиков токарного цеха я написал android-приложение. Менеджеры использовали веб-сайт.

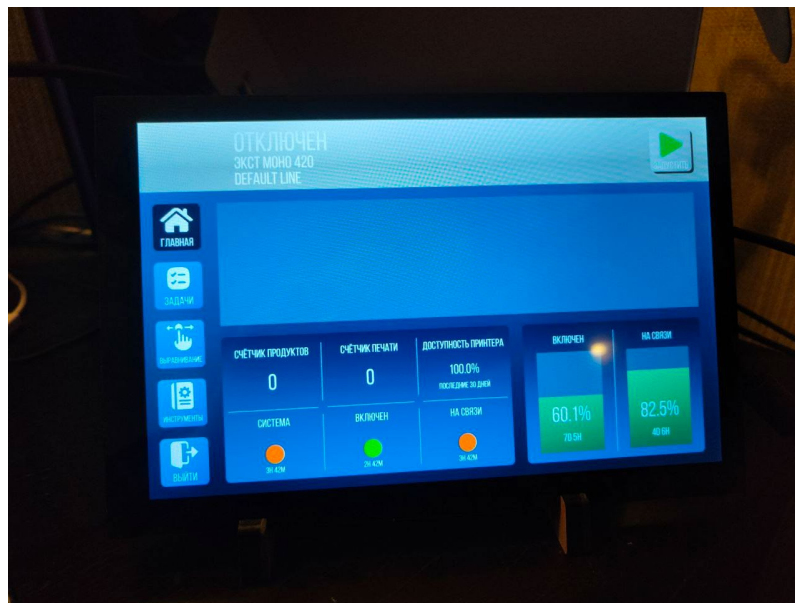


Вся информация хранится на веб-сервере на Django+Python. Для визуализации использую ibm-gantt (диаграммы Ганта) chart.js (графики) и bootstrap



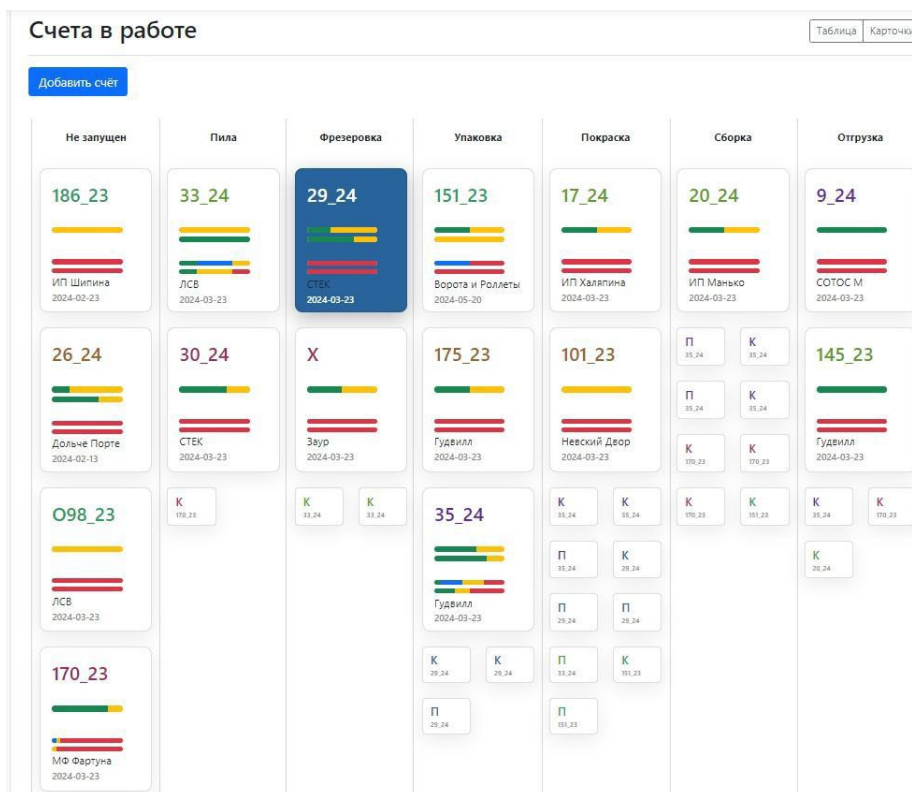
5. Институт ЭКСТ

Для [института электро-капеструйных технологий](#) я разработал программу управления принтером с графическим интерфейсом под плату HelperBox T507 с помощью Qt Embedded (C++).



6. Компания «Форма»

Для данной компании я разрабатывал систему диспетчеризации. Через единую систему исполнители видели карточки заданий своего цеха, а менеджер – всех цехов.

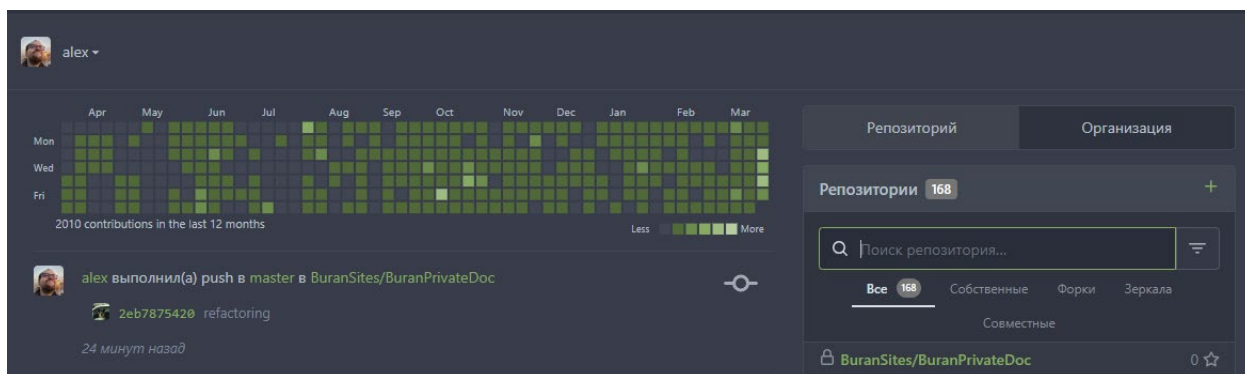


7. Дополнительная информация

7.1. Сетевое администрирование

У меня дома стоит сервер с гипервизором. На одной из виртуальных машин поднят обратный прокси nginx, он обрабатывает запросы к домену buran.center через внешний DNS-сервер.

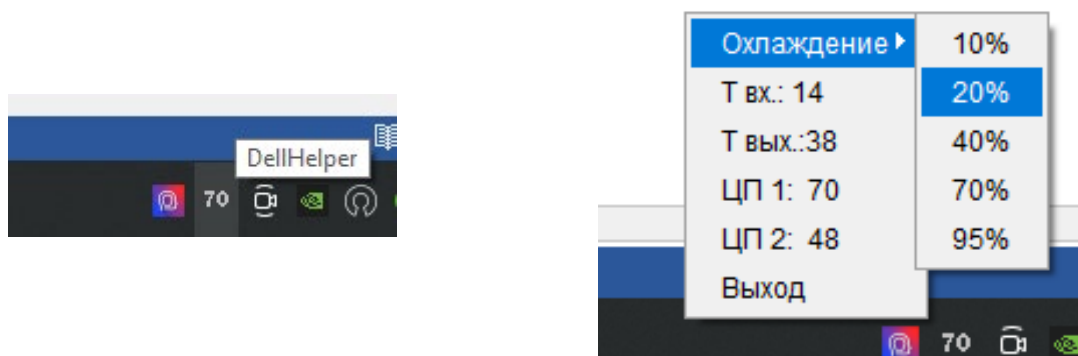
Свои сайты я развернул на нескольких виртуальных машинах. Все редактируемые мной сайты оборачиваются в докер-контейнер с помощью сервера Jenkins. Он, в свою очередь, в автоматическом режиме следит за изменениями в Gitea (git сервер с открытым исходным кодом).



Чтобы распределять запросы к этим сайтам на этой же виртуальной машине поднят Traefic, обёрнутый в докер-контейнер.

Также я развернул свой [maven сервер](#). Зависимости для него публикуются автоматически с помощью Jenkins. На отдельных машинах я развернул базу данных Postgerss и сервер облачного хранения Nextcloud.

Т.к. при вычислительных экспериментах сервер сильно греется, то для управления охлаждением сервера я написал небольшую программу в трей. Для управления использовался внутренний консольный инструмент DELL ipmitool. Программа в трее по таймеру запускает консольную команду запроса температуры и сохраняет ответ во внутренних переменных.



Также данная программа позволяет задавать мощность работы системы охлаждения в процентах.

