Drawing robot (plotter)

Цель проекта

Разработать работающую модель робота-плоттера для рисования и преобразования печатаемого текста в рукописный шрифт на лист А4

Значимость проекта

В современном мире человек всё меньше пишет текст от руки, ведь набор его на компьютере выглядит проще, хотя и в напечатанном виде он выглядит несколько однообразно. В связи с этим может возникнуть необходимость "ручного" написания, однако в таком случае невозможно что-либо исправить аккуратно. И в таком случае подобный проект - лучшее решение, позволяющее создать идеальный текст в электронном виде и затем перевести его в красивый рукописный, не затрачивая при этом собственных сил, так как для этого нужно всего лишь загрузить данные на робота.

Задачи

1. Придумать и собрать корпус и движущуюся часть робота на трёх осях.Выполняется в виде плоттера с помощью EV3 и его комлпектующих.
2. Реализовать рисование.
3. Внедрить программу для перевода печатаемого текста в рукописный

Описание работы

Наш робот должен принимать с компьютера входной параметр для будущего рисунка или текста в виде набора параметров передвижения по двум горизонтальным осям (*X* и *Y*) и на их основе выполнять движение. Человек, в свою очередь, пишет обычный текст в консоль, а на выходе получает рукописный текст.

Этапы работы

1. Начальный этап - создание внешнего вида робота и проверка его работоспособности. Сборка плоттера с пятью моторами, два из которых отвечают за движение по горизонтали (*X*), два за движение по вертикали (*Y*) и один за подъем/опускаие канцелярии.
2. Написание кода на Python.
3. Моделирование в *Gazebo*.
4. Внедрение ROS, настройка управления роботом.
5. Подключение функции преобразования текста.

Структура робота

Весь проект выполнен с использованием комплектующих *Lego EV3*

1. Корпус
    - рама для крепления на ней движимой части расчитаная на массу нашего робота
2. Ось X
    - два мотора
    - две рейки для крепления моторов оси Y
3. Ось Y
    - два мотора
    - рейки для крепления оси Z
4. Ось Z
    - мотор
    - крепление для опускания/подъёма карандаша
5. Блок *Lego EV3*
6. Держательный элемент

Что было сделано и чего удалось достичь на начальном этапе

После множества часов, проведённых за совершенствованием конструкции робота и борьбы с цепляющимися друг за друга элементами, мы наконец смогли достичь баланса. В результате работы удалось сделать робота с наиболее плавным и устойчивым движением двигателей по осям. Для более качественной работы блока *EV3* образ системы был установлен с сайта ev3dev на полностью "чистую" SD-карту. По итогам нескольких тестов удалось найти максимально правильные необходимые напряжения для подачи на двигатели.

Пример работы робота

Подключение ROS

Использованный софт

1. *ROS Noetic* на виртуальной машине Ubuntu версии 20.04
2. Образ *H4R Yochto Linux*
3. Контейнеры *Docker* версии 20.10.7
    - Используется для автоматизации развёртывания и управления
    - Также помогает ПО работать везде одинаково
4. Универсальная клавиатура *Teleop* для ROS (для управления роботом)
5. Контейнеры *rosev3*

Коротко о главном

Основные (и единственные найденные) возможности *EV3* взаимодействовать с *ROS* - использование пакета *ros-comm* и использование образа *H4R Yochto Linux*. В первом способе необходимо использование инструмента *brickstrap* для создания образа. Данный инструмент разрабатывался для *Ubuntu* и хорошо работал на виртуальной машине. Сейчас не поддерживается.

Для второго способа собственная SD-карта создаётся путём скачивания и дальнейшей распаковки файлов на неё. SD-карта форматируется и делится на две части. Одна из них имеет файловую систему *FAT16* на 48 Мб, другая же остаётся просто USB-накопителем (*FAT32* или *NTFS*) на 7,35 Гб.

С этим связаны основные проблемы - большая часть файлов не читается операционной системой, что было проверено как на *Windows*, так и на *Linux*. Очевидно, что они сделаны для *OC Linux*, однако в ней были обнаружены ошибки при попытке разархивировать файлы как "вручную", так и через консоль. Из-за этого *EV3* не смог работать в полной мере.

Управление движением робота с компьютера также позволяет использовать его в формате простейшего графического планшета, наиболее подходящего для простых фигур.

Конвертация текста

Решений для преобразования печатного текста в рукописный существует также не очень много.

В данном случае использовался следующий алгоритм, который позволяет преобразовать печатный текст в письменный и с его помощью можно получить желаемый текст своим родным почерком.

Для реализации необходим пример почерка и текст, который нужно написать. В базу данных добавляется новый почерк и затем проводится его анализ. Сначала он помещается ровно, чтобы быть точно на линии, происходит центровка. Затем происходит сегментация и выделение строк. Если существуют какие-то пробелы (например, часть буквы плохо прописана ручкой), то они заполняются пользователем вручную. Далее используется векторизованное линейное представление. Текст преобразуется, разделяются буквы. Если в тексте присутствуют какие-то ошибки, пользователь их исправляет, после чего элементы запоминаются.

После этого символы сопоставляются и получается рукописный текст. Для проверки точности используется функция стоимости (*cost function*). Текст выглядит не совсем естественно, поскольку некоторые буквы плохо читаемы и между ними есть большие пробелы в связи с использованием динамического программирования. Для вертикального выравнивания используется распределение по вертикальным смещениям. Эта информация интегрируется с помощью фильтра Калмана. Затем добавляются связи между буквами с помощью текстурирования.

Идея данного алгоритма хороша и полезна, однако он не идеален и в нём много неточностей. Он не полностью автоматизирован, имеет некорректности в работе. И, к соалению, внедрить его полностью не удалось.

Более подробно про алгоритм